

/THEORY/IN/PRACTICE

The Art of Community

Building the New Age of Participation

O'REILLY®

Jono Bacon
Foreword by Leo Laporte

The Art of Community

"In the age of participation, there is no better tour guide than Jono Bacon. The Art of Community teaches leaders how to increase collaboration and authenticity. This isn't just for technology leaders—anyone who wants to harness community for their cause should read this book."

—Amanda McPherson, The Linux Foundation

Online communities offer a wide range of opportunities today, whether they're supporting a cause, marketing a product or service, or developing open source software. *The Art of Community* will help you develop the broad range of talents you need to recruit members to your community, motivate and manage them, and help them become active participants.

Author Jono Bacon offers a collection of experiences and observations from his decade-long involvement in building and managing communities, including his current position as community manager for Ubuntu, arguably the largest community in open source software. You'll discover how a vibrant community can provide you with a reliable support network, a valuable source of new ideas, and a powerful marketing force.

- 
- Develop a strategy, with specific objectives and goals for building your community
 - Build simple, nonbureaucratic processes to help your community perform tasks, work together, and share successes
 - Provide tools and infrastructure to help contributors work quickly
 - Create buzz around your community to get more people involved
 - Track the community's work so it can be optimized and simplified
 - Explore a capable, representative governance strategy for your community
 - Identify and manage conflict, including dealing with divisive personalities

Jono Bacon is an award-winning community manager, consultant, and author of four books. He also maintains one of the most popular blogs about open source software.

Leo Laporte is a broadcaster and founder of the TWiT Network.

US \$39.99

CAN \$49.99

ISBN: 978-0-596-15671-8



Safari[®]
Books Online

Free online edition
for 45 days with purchase of
this book. Details on last page.

O'REILLY[®] oreilly.com

Praise for *The Art of Community*

“The Internet provides the potential to separate us into a cacophony of discordant voices or to congregate us as purpose-driven communities. Jono Bacon, in his insightful *The Art of Community*, teaches the latter path, detailing the principles of successful community-building in a way that will appeal to both neophyte and expert alike. Given the increasingly critical role of community managers in the technology industry and beyond, *The Art of Community* should find a place on any businessperson’s bookshelf, not to mention that of the PTA president, book club organizer, or union activist. Yes, it’s that good.”

—MATT ASAY, ALFRESCO AND C|NET

“Jono Bacon truly understands communities, and, more importantly, how to build communities that thrive. This is the definitive guidebook to building successful communities—definitive because it is based on Jono’s extensive experience as community manager for Ubuntu, a product that inspires an Apple-esque devotion in very large part because of its vast and dedicated community. For developers and entrepreneurs who want to learn how to tap into the power of community, as Ubuntu has done so masterfully, this book is a must-read.”

—IAN MURDOCK, FOUNDER OF DEBIAN AND VICE PRESIDENT OF EMERGING PLATFORMS AT SUN

“One thing that’s impressed me about Jono Bacon—something one can notice back when he and others were building a community around their pioneering Linux podcast—is that he simply gets the concept of community. It comes out in most everything he says and most every decision he makes. This is the kind of a person you want writing a book on the topic. Open source community building cannot be boiled down to a formula. It’s a constant effort, a soft science, an art, and Bacon is an ideal art teacher.”

—DAN GOLDSTEIN, PROFESSOR OF MARKETING, LONDON BUSINESS SCHOOL,
AND PRINCIPAL RESEARCH SCIENTIST, YAHOO! RESEARCH

“The success of the open source software movement demonstrates that no obstacle is insurmountable when people come together around a shared vision. In *The Art of Community*, Ubuntu Community Manager Jono Bacon gives readers a profound glimpse into his hands-on experience as the orchestrator of one of the movement’s most powerful communities. His book offers valuable lessons on effective leadership and community building. Its compelling combination of useful theory, real-world best practices, and instructive personal anecdotes make it a richly comprehensive guide for both aspiring and experienced community leaders.”

—RYAN PAUL, ARS TECHNICA

“Communities are very complex ecosystems of human beings. Cultivating, growing, shaping, and guiding the community to make it productive is definitely as much (or even more) art as science. In *The Art of Community*, Bacon does an excellent job of explaining in detail the considerations for managing and cultivating a healthy open source community. He provides a blueprint for developing and maintaining an open source community in a programmatic way, and his attention to detail and understanding of the dynamics of communities make this book an invaluable resource for anyone looking to build and maintain a community. Drawing from his own extensive experience, Bacon does a great job of explaining how to help foster a community, and provides great advice, ranging from choosing infrastructure, measuring growth, and even hiring a community manager. All in all a must-read for any community manager.”

—MARK R. HINKLE, VICE PRESIDENT OF COMMUNITY, ZENOSS, INC.

“Jono Bacon has long been an insightful voice for the open source community. Now his artful stories distilling the ethos of organizing people and activities on the Net, at conferences, and in our daily routines provide a framework for successful, community-building strategies.”

—PETE KRONOWITT, LINUX AND OPEN SOURCE STRATEGIST, INTEL

“In *The Art of Community*, Jono Bacon once again shows that his nom de guerre is apropos. He breaks down the soft science of community management in a way few others could. With his trademark British humor, he deftly explores the intricacies and subtleties of his trade. The result is both informative and entertaining, and is a must-read for those looking to better understand the soft science that is community management.”

—JEREMY GARCIA, FOUNDER OF LINUXQUESTIONS.ORG

“To a soundtrack of heavy metal, free-software geekstar Jono Bacon recounts the story of how he learned to gently yet productively manhandle groups of unruly Internet folks gathered around a common topic or cause. His process and methods are set out in his book, *The Art of Community*, where Jono’s non-ego-driven account of community building will aid all manner of bosses, since almost every subject matter these days has a community with hundreds, thousands, tens of thousands, and even (as in the case of *World of Warcraft*) millions of people clamoring around it. (Even David Hasselhoff!) Be forewarned, capitalist! There is no chapter called ‘How to Turn Communities into Dollars,’ but following Jono’s suggestions may yield you what every leader (even a capitalist) wants: a loyal and passionate community willing to collaborate to achieve a common goal.”

—IRINA SLUTSKY, GEEKENTERTAINMENT.TV

“If you listen to open source fans, you might get the idea that the community is elves who come out of the woodwork to fix your broken software while you sleep. In *The Art of Community*, Jono Bacon explains how reality is a little more complicated, and what the community needs in return. This book will help you get started with the diverse skills required to keep a collaborative community on track, including copywriting, social software selection, conflict resolution, and measuring if it’s all working.”

—DON MARTI, CONFERENCE CHAIR, OPENSOURCE WORLD, AND ORGANIZER,
WINDOWS REFUND DAY, BURN ALL GIFS DAY, FREE DMITRY, AND
FREEDOMHEC

“Who would have known, when I first met a scruffy student from Wolverhampton Uni at a LUG meeting all those years ago, that he would end up being the name on the Internet synonymous with the word ‘community.’ The fact that the Internet’s Jono Bacon is now one of the foremost authorities on building and nurturing a community shows that in a volunteer project no one cares about your questionable dress sense, dodgy taste in music, or strange choices in facial hair—all that matters are your contributions, and your ability to get on with, and inspire, others.

“In this book, Jono draws upon a wealth of experience from projects small to big (and when you consider the worldwide phenomenon that was LugRadio, and the worldwide phenomenon that is Ubuntu, you’re talking pretty big) to lay out a blueprint for creating and sustaining communities, as well as using real-world examples from prime ministers to celebrity chefs to ground the topics in a wider context. There is a nice balance in that many of the examples are based on success stories, but Jono is brave enough to also illustrate his points with some of his (relatively few) mistakes.

“This book will be useful for anyone looking to build a volunteer community around any kind of project or cause, whether it involves software, open source, raccoons, or none of the above.”

—PAUL COOPER, MOBLIN UI & APPS ENGINEERING MANAGER, INTEL

“As a rock-solid book, *The Art of Community* is not only about communities, but also management, organization, and even marketing—it is the bible for community leadership. This book should have been out a long time ago, and reading through the chapters made me reflect on almost every important situation I had to face with teams, from conflicts all the way to handling buzz. It would have helped solve some of the issues I was stuck in much faster than I did (although all the issues solved in the end were exactly how Jono described it). I am eager to apply more of this wisdom on the current projects I am involved in.”

—SEIF LOTFY, GNOME FOUNDATION, ZEITGEIST COFOUNDER AND TEAM LEADER

“Few people, in my experience, understand how to create, build, and support community better than Jono Bacon. With *The Art of Community*, Jono’s taken his experience, his intelligence, as well as his great humor, and has effectively distilled it into an indispensable book for anyone who wants to start a community (whether around software or any other shared interest or endeavor, really) or participate in one in a positive and productive way. Jono understands that communication and authenticity are at the core of effective participation, and goes beyond the theoretical to provide practical guidance on things like governance, process, conflict resolution, and avoiding burnout that is right on the mark. *The Art of Community* is an excellent book!”

—DAVID SCHLESINGER, DIRECTOR, OPEN SOURCE TECHNOLOGIES,
ACCESS CO., LTD.; GNOME FOUNDATION ADVISORY BOARD MEMBER

“Jono Bacon, in *The Art of Community*, takes you on a personal journey to the heart of what it takes to have and become part of a productive and well-oiled community.”

—AMBER GRANER, UBUNTU COMMUNITY MEMBER

“Jono Bacon’s *The Art of Community* is a wonderful meditation on building communities using modern infrastructure tools and practices gleaned from the Free and Open Source Software movement. Jono’s examples, taken from his work on Ubuntu, give a good picture of a working community and how it functions. The fact that the book is backed by a conference (http://www.communityleadershipsummit.com/wiki/index.php/Session_Notes) and an online community (<http://artofcommunityonline.org/>) means this fine effort will potentially continue to grow into *the* watering hole for community gardeners, leaders, and managers.”

—DANESE COOPER, OPEN SOURCE DIVA AND OSI DIRECTOR

The Art of Community

The Art of Community



Jono Bacon

O'REILLY®

Beijing • Cambridge • Farnham • Köln • Sebastopol • Taipei • Tokyo

The Art of Community
by Jono Bacon

Copyright © 2009 Jono Bacon. All rights reserved.
Printed in the United States of America.

Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.

O'Reilly books may be purchased for educational, business, or sales promotional use. Online editions are also available for most titles (<http://my.safaribooksonline.com>). For more information, contact our corporate/institutional sales department: (800) 998-9938 or corporate@oreilly.com.

Editors: Andy Oram and Simon St.Laurent
Production Editor: Sumita Mukherji
Copyeditor: Genevieve d'Entremont
Proofreader: Sada Preisch

Indexer: John Bickelhaupt
Cover Designer: Mark Paglietti
Interior Designer: David Futato
Illustrator: Robert Romano

Printing History:
August 2009: First Edition.

Nutshell Handbook, the Nutshell Handbook logo, and the O'Reilly logo are registered trademarks of O'Reilly Media, Inc. *The Art of Community*, the image on the cover, and related trade dress are trademarks of O'Reilly Media, Inc.

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and O'Reilly Media, Inc., was aware of a trademark claim, the designations have been printed in caps or initial caps.

While every precaution has been taken in the preparation of this book, the publisher and author assume no responsibility for errors or omissions, or for damages resulting from the use of the information contained herein.

Printed on Rolland Enviro100 Book, which contains 100% post-consumer fibers and is manufactured in Canada by Cascades using biogas energy.

ISBN: 978-0-596-15671-8

[V]

1248982100

*For my loving wife, Erica, and all the ways
she makes me smile*

CONTENTS

	FOREWORD	xvii
	PREFACE	xix
1	THE ART OF COMMUNITY	1
	<i>Collaboration-Driven Ethos</i>	2
	<i>The Essence of Community</i>	3
	<i>The Basis of Communication</i>	8
	<i>Unwrapping Opportunity</i>	10
	<i>A Community Manager: Becoming the Community</i>	13
	<i>Moving Forward</i>	19
2	PLANNING YOUR COMMUNITY	21
	<i>Planning for Success</i>	23
	<i>Teams: The Building Blocks of Belonging</i>	31
	<i>Designing Your Community</i>	43
	<i>Filling Out the Plan</i>	52
	<i>Pulling Together the Threads</i>	55
	<i>Documenting Your Strategy</i>	61
	<i>Wrapping Up</i>	63
3	COMMUNICATING CLEARLY	65
	<i>He Said, She Said</i>	66
	<i>Building Your Communication Channels</i>	67
	<i>Leading by Example</i>	75
	<i>Summary</i>	89
4	PROCESSES: SIMPLE IS SUSTAINABLE	91
	<i>Eyes on the Prize</i>	92
	<i>Building Great Processes</i>	94
	<i>Assessing Needs</i>	101
	<i>Getting Buy-In for Your Processes</i>	112
	<i>Process Reassessment</i>	115
	<i>Moving On</i>	117
5	SUPPORTING WORKFLOW WITH TOOLS	119
	<i>Understanding Your Workflow</i>	120
	<i>Building Great Infrastructure</i>	129
	<i>Avoiding Resource Fetishism</i>	133
	<i>Technical Considerations</i>	133
	<i>Building and Maintaining Transparency</i>	137
	<i>Regular Workflow Assessment</i>	141

	<i>Moving On</i>	143
6	BUILDING BUZZ	145
	<i>Mindshare</i>	146
	<i>The Building Blocks of Buzz</i>	148
	<i>Setting Up Your Base</i>	154
	<i>The Buzz Cycle</i>	163
	<i>Buzz Targets</i>	170
	<i>Building Alliances</i>	175
	<i>Summary</i>	186
7	MEASURING COMMUNITY	187
	<i>Community Self-Reflection</i>	188
	<i>The Foundations of Feedback</i>	189
	<i>Hooks 'n' Data</i>	190
	<i>Anonymity and Privacy</i>	206
	<i>Moving On</i>	209
8	GOVERNANCE	211
	<i>Accountability</i>	213
	<i>Governance Does Not Suck</i>	213
	<i>Governance and Community</i>	214
	<i>The Case for Governance</i>	215
	<i>Learning from the Leaders</i>	221
	<i>Setting Up a Community Council</i>	226
	<i>Ubuntu Governance Example</i>	245
	<i>Expanding Governance</i>	260
	<i>Summary</i>	265
9	HANDLING CONFLICT	267
	<i>The Nature of the Beast</i>	268
	<i>The Calm Before the Storm</i>	270
	<i>The Conflict Resolution Process</i>	280
	<i>Dealing with Burnout</i>	294
	<i>Summary</i>	300
10	CREATING AND RUNNING EVENTS	301
	<i>Building Family Values</i>	302
	<i>Events</i>	303
	<i>Getting Organized</i>	305
	<i>Organizing Physical Events</i>	308
	<i>Getting Sponsorship</i>	324
	<i>Organizing Online Events</i>	329
	<i>Summary</i>	339
11	HIRING A COMMUNITY MANAGER	341
	<i>Why Community Building Has Become a Big Business</i>	343
	<i>The Role of a Community Manager in the Corporation</i>	344
	<i>Managing Your Community Manager</i>	351
	<i>Summary</i>	356

End of Part One

357

INDEX

359

FOREWORD

FROM ANTS TO ANTEATERS, bees to beekeepers, community is a fundamental part of our life on the planet. We thrive when we are immersed in it, suffer when deprived of it, and wherever humans go we create it. We define ourselves by our communities: tribe, family, work, clubs, schools, churches and temples, these are who we are. We are born into community, and if we're lucky we'll end our days surrounded by it.

It's no surprise that as soon as humans began to go online, communities formed, but as easy and natural as group formation is for us in real life, we can find it frustrating online. Many of the cues that grease the wheels of human interaction in person are missing online. Gone is the grin that can soften a criticism, the pat on the back that can heal a rift. How can you "hug it out" when your antagonist is a continent away and you know no more about him than his handle and a few lines of signature? Online groups can breed the most vicious of rivalries. The Hatfields and McCoys have nothing on *alt.tv.doctorwho*.

Communities are tough enough to maintain when you're all in the same room; how much harder is it to build, maintain, and nurture a community online? That's why this book is such a boon to those who run communities and the rest of us who participate in them. Jono Bacon has firsthand experience with managing a group of the most bloody-minded and independent people on the planet: open source programmers. The information in this book has been forged in the white-hot crucible of free software. You don't get tougher than that.

My experience with online forums began 25 years ago when I started a bulletin board for Macintosh users called MacQueue. It's not easy to start a flame war with dual 14.4 kbps modems and 20 MB of storage, but the MacQueuers managed. A few years later I joined The Well, a legendary online community based in Sausalito, California, and imbued with the peace and love ethos of the San Francisco hippies. That didn't last long. The Well went through an arc I came to know intimately, one that most online communities seem to follow.

When any affinity group forms online it's a joyous occasion. The founders and early members are wreathed in the cooperative enthusiasm that accompanies most new beginnings. Conversations are civil, helpful, and kind. Posts twinkle with good spirits and bonhomie. All's right with the Web. Then the rot begins to set in. Tempers flair, resentments build, rivalries form. It's a lot like marriage.

Unlike most marriages, however, online members have looser ties to the group and a reduced stake in its success. When trolls become annoying, the flame wars too fiery, members move on, and pretty soon that happy online forum turns into a ghost town, or worse.

But it doesn't have to be that way. With his usual wit and good humor, Jono has written a guide with everything you need to keep your online groups healthy and productive. With proper planning, a modicum of guidance, and the occasional banishment, your community can avoid that seemingly inevitable descent into fear and loathing. We need good community managers because we need healthy communities online. I've started my share of communities online, and killed a few with neglect, too. I'm so grateful to Jono for giving me the tools to do it right from now on. I know we all are.

—Leo Laporte
Broadcaster and Founder of the TWiT Network
Petaluma, California
June 30, 2009

P R E F A C E

COMMUNITY IS A FUNNY OL' WORD. In recent years our humble nine-letter friend has gone on to mean many things to many people. No longer merely the domain of charity groups and overtly friendly neighbors, community has gone on to be the talk of technologists, business-people, politicians, students, welfare groups, and just about anyone who has connected to the Internet. Throughout this explosive community love-fest, a minor detail has been omitted in all the excitement: how on earth do we built an inspiring, engaging, and enjoyable community in our own walk of life?

Toward the end of Summer 2008 I received a phone call from Andy Oram, a well-respected author and editor at O'Reilly. Although at the start of the call Andy was soliciting advice for building community in the educational world, the call ended by sowing the seeds for *The Art of Community*.

Andy's interest in putting together this book was intriguing, but it could not have come at a more complicated time. My days were hectic as the Ubuntu community manager, leading my team to grow, refine, and optimize the global Ubuntu community; I was in the midst of recording a solo metal album as part of a new Creative Commons project called Severed Fifth; I was coorganizing LugRadio Live 2008, recording and producing LugRadio shows every two weeks; and I was making plans to relocate to California. I had written three books before and I was intimately aware of just how incredibly time-consuming they are. Writing a book is like having a baby: it requires care and attention, and typically results in late nights, lack of sleep,

and heartburn. Consequently, my best friend (who is also an author) and I had struck a no-more-books pact.

Despite all of this, I was intrigued. Community and the skills involved in motivating, building, and inspiring it were rampantly undocumented, and much of my own skills had been developed through trial and error, exposing myself to different communities and observing how they worked. I was fortunate enough to have cut my teeth in community in some compelling environments, and I had always wanted to write a book on the topic.

Fortunately none of these aforementioned challenges made any difference when I talked it through with my best friend, Stuart. He and I have been discussing, debating, and at times arguing about community since 1999, and he knows my views, perspectives, drive, and ambitions about community better than anyone. What's more, he had been wittering on about me writing something down about community, despite our no-more-books pact. Ten minutes with that ginger ball of fury and my mind was made up: it was time to buy some anti-heartburn pills and get some coffee in....

Documenting the Undocumented

Part of my initial hesitation in writing a book on community was that I knew it was going to be a tough one to write. In my talks at conferences I often referred to my role as “herding cats.” Much of the art of community is subtle, undocumented, and unwritten, and much of my own approach was largely the product of feeling my way around in the dark and learning from what I found. I knew that to write this book I would need to think carefully about not only how to articulate these topics, but also how to handle the more complex challenge of structuring this stream of consciousness into a consistent read that, y’know, actually makes sense.

What you hold here is the result of that challenge, and I am proud of the results. This book brings together many of the primary elements involved in building a productive, collaborative community. To do this I distilled my own experiences and insight along with wisdom from others and illustrated these topics using a wealth of examples, stories, and anecdotes. This book begins by taking a high-level view of how communities work at a social science level, and then we delve straight into topics such as strategic planning, communicating well, building effective and non-bureaucratic processes and infrastructure, creating buzz and excitement, handling conflict and burnout, measuring community, creating and managing governance, organizing events, and even how to hire a community manager.

While this first edition provides a solid map for the road ahead, I am a firm believer that the road map will continue to expand and take on color and texture through further editions. Community leadership is still very much a young science, and this book is the beginning of what I hope to see as a series of further editions that expand and refine this focus. Where much of this insight will continue to grow is on the book’s website at <http://www.artofcommunityonline.org> and at the annual community leadership event that I organize, the [Community Leadership Summit](#). I would like to invite all of you good people to first enjoy *The*

Art of Community and to then provide your own feedback, stories, and experiences to guide future editions.

Who Is This Book For?

This book has been written to be open and applicable to a wide range of communities. While O'Reilly is traditionally a computer book publisher, *The Art of Community* is not specifically focused on computing communities, and the vast majority of its content is useful for anything from political groups to digital rights to knitting and beyond.

Within this wide range of possible communities, this book will be useful for a range of readers:

Professional community managers

If you work in the area of community management professionally

Volunteers and community leaders

If you want to build a strong and vibrant community for your volunteer project

Commercial organizations

If you want to work with, interact with, or build a community around your product or service

Open source developers

If you want to build a successful project, manage contributors, and build buzz

Marketeers

If you want to learn about viral marketing and building a following around a product or service

Activists

If you want to get people excited about your cause

Every chapter in this book is applicable to each of these roles. While technology communities provide many examples throughout the book, the purpose of these examples requires little technical knowledge.

The Road Ahead

Throughout this book we are going to delve into the wide range of topics that face those of us who want to build and inspire great communities. Page after page we are going to weave an intricate web of the concepts, skills, and approaches involved in energizing a vibrant community and helping the members of that community to energize themselves.

This book is broken into 11 chapters, with each building on what went before. Let's take a quick glance at the road ahead:

Chapter 1, *The Art of Community*

We begin the book with a bird's-eye view of how communities function at a social science level. We cover the underlying nuts and bolts of how people form communities, what keeps them involved, and the basis and opportunities behind these interactions.

Chapter 2, *Planning Your Community*

Next we carve out and document a blueprint and strategy for your community and its future growth. Part of this strategy includes the target objectives and goals and how the community can be structured to achieve them.

Chapter 3, *Communicating Clearly*

At the heart of community is communication, and great communicators can have a tremendously positive impact. Here we lay down the communications backbone and the best practices associated with using it.

Chapter 4, *Processes: Simple Is Sustainable*

We then move on to focus on putting the facilities in place for your community to do great things. In this chapter we build simple, effective, and nonbureaucratic processes that enable your community to conduct tasks, work together, and share their successes.

Chapter 5, *Supporting Workflow with Tools*

We continue our discussion of community facilities to build workflows that are driven by accessible, sensible, and rock-solid tools that enable your contributors to do great work quickly and easily.

Chapter 6, *Building Buzz*

With a solid foundation in place, we move on to build excitement and buzz around your community and encourage and enthrill every man and his dog to get involved and participate.

Chapter 7, *Measuring Community*

Although many consider community hand-wavy and unmeasurable, this chapter confronts the myth and guides you in tracking, monitoring, and otherwise measuring the work going on the community so it can be optimized and simplified.

Chapter 8, *Governance*

Our next stop is the wide-ranging and seemingly complex topic of governance. We explore what options are available for a low-friction, capable, and representative governance strategy for your community.

Chapter 9, *Handling Conflict*

One of the most sensitive topics in community leadership is handling conflict. In this chapter we explore how to identify, handle, and prevent irksome conflict; handle divisive personalities; and unblock problems.

Chapter 10, *Creating and Running Events*

Events offer an excellent opportunity for your community to bond, be productive, and have fun, and this is where we cast our beady eye in our penultimate chapter.

Chapter 11, *Hiring a Community Manager*

Finally, we close *The Art of Community* with some advice and guidance for organizations who want to hire a community manager to conduct and implement the wide range of topics that we have discussed throughout the book.

Each of these broad topics is a piece in the jigsaw puzzle, a note in the song, and a letter in the book. Step by step we will discuss these topics using a liberal supply of stories, anecdotes, and examples to illuminate the path ahead. As we continue throughout the book, more and more of the road will become clear, and you will begin to develop your own approaches, patterns, and methods of engaging your own community.

If You Like (or Don't Like) This Book

If you like—or don't like—this book, by all means, please let people know. Amazon reviews are one popular way to share your happiness (or lack of happiness), or you can leave reviews at the site for the book:

<http://oreilly.com/catalog/9780596156718/>

There's also a link to errata there. Errata gives readers a way to let us know about typos, errors, and other problems with the book. That errata will be visible on the page immediately, and we'll confirm it after checking it out. O'Reilly can also fix errata in future printings of the book and on Safari, making for a better reader experience pretty quickly.

License

This book is licensed under a Creative Commons Attribution Noncommercial Share Alike license.

Join Our Community

Since the beginning of *The Art of Community* project, this book has developed its own community, which is primarily composed of those passionate about building strong and compelling communities.

The hub of this activity is at <http://www.artofcommunityonline.org>. The website has a range of features available at the time of writing, and likely will have many more when you get there:

Download the book

You can download the full version of the book, available under the Creative Commons Attribution Noncommercial Share Alike license.

News

Get updates on the book, and share and read about success stories of communities who are using the book.

Discuss

The website is filled with resources in which you can chat and talk about great community building with other readers.

Articles

Stories, case studies, and other content are regularly published to the site, to help build your knowledge.

Feedback

The website is a great place to leave feedback about the book for future editions.

In addition to the main website, you can also keep up-to-date with news on the book and other community-building stories on Twitter at <http://www.twitter.com/jonobacon>.

Typographical Conventions Used in This Book

The following typographical conventions are used in this book:

Italic

Indicates new terms, URLs, email addresses, filenames, file extensions, pathnames, and directories.

Constant width

Indicates code, text output from executing scripts, XML tags, HTML tags, and the contents of files.

How to Contact O'Reilly

We have tested and verified the information in this book to the best of our ability, but you may find that features have changed (or even that we have made a few mistakes!). Please let us know about any errors you find, as well as your suggestions for future editions, by writing to:

O'Reilly Media, Inc.
1005 Gravenstein Highway North
Sebastopol, CA 95472
800-998-9938 (in the U.S. or Canada)
707-829-0515 (international/local)
707-829-0104 (fax)

We have a web page for this book, where we list errata, examples, and any additional information. You can access this page at:

<http://oreilly.com/catalog/9780596156718>

To comment or ask technical questions about this book, send email to:

bookquestions@oreilly.com

For more information about our books, conferences, Resource Centers, and the O'Reilly Network, see our website at:

<http://oreilly.com>

Safari® Books Online

 When you see a Safari® Books Online icon on the cover of your favorite technology book, that means the book is available online through the O'Reilly Network Safari Bookshelf.

Safari offers a solution that's better than e-books. It's a virtual library that lets you easily search thousands of top tech books, cut and paste code samples, download chapters, and find quick answers when you need the most accurate, current information. Try it for free at <http://my.safaribooksonline.com/>.

Acknowledgments

This book was a long time coming, and I want to thank a number of people who directly and indirectly helped me to make this book reality. First and foremost I want to thank my incredible family: my wonderful and hugely encouraging wife, Erica; my Mum and Dad; Martin and Simon; Grandad; Joe; Sue; Adam; and Lindsay. Thanks also to my best pal Aq for bullying me into writing this book; to the horsemen Daniel Holbach, Jorge Castro, and David Planella on my team at Canonical; Matt Zimmerman, Mark Shuttleworth, and the LugRadio Team (Adam, Chris, Ade, and Matt).

I also want to express huge thanks to the many people who contributed stories and interview content that is featured in this book. Your contributions have added flesh to the bones and helped to illustrate the book so well. Thanks also to the many people who are featured in the book's stories and examples; those experiences have taught me so much and allowed me to share this with my readers.

I also want to share my utmost thanks and gratitude to Andy Oram from O'Reilly for making this book happen, and to Simon St.Laurent and Isabel Kunkle in lending their editing prowess. Also thanks to our fantastic team of review editors: Stephen Walli, Stuart Langridge, Amber Graner, and Erica Bacon.

Finally, huge thanks to everyone who has supported my work and the book on <http://www.artofcommunityonline.org/> and <http://www.jonobacon.org/>, and to the hundreds of people who have spread the word throughout their blogs, podcasts, Twitter/identi.ca feeds, Facebook, and elsewhere. I appreciate every ounce of your support!

All right, 'nuff chatting. Let's get started....

The Art of Community

“Great things are not done by impulse, but by a series of small things brought together.”

—*Vincent Van Gogh*

AS MY WATCH TICKED OVER TO 6 P.M., I KNEW I WAS IN TROUBLE. First of all, I was late, and not fashionably late, either. In fact, at the time, I was about as unfashionable as you could get for someone staring 18 down the barrel. Long hair, Iron Maiden t-shirt, baggy camouflage trousers, and a thumping-great leather jacket. I left my parents’ house and got into my small van, adorned with oversized speakers and a tree-shaped air freshener. It was time to roll.

“Rolling” was optimistic. Instead, I sat bumper-to-bumper in traffic with half of Southern England, all joined in curiosity about whether or not that film with Michael Douglas could become a reality on this cold English day.

This wasn’t helping my nerves. As a fairly outgoing, angsty teen, nerves were not usually my bag, but tonight, I was dining on them.

You see, tonight was different. Tonight I was doing something unusual, something that had seemed like a great idea...when I wasn’t running 30 minutes late, hammering my way down the motorway, with my *Number of the Beast* cassette ritualistically sacrificed to the gods of hi-fi just for good measure.

Thankfully, the world’s longest mechanical conga line decided to crank it up a notch. Before I knew it, I found myself on a street I had never been to, in a city I had never been to, about to head into a room full of people I had never met before, all united by one simple symbol....

A penguin.

An hour before, that penguin had seemed so inviting and friendly. It was a symbol that encompassed everything about the movement it represented, a movement that came together in spirit and mind to build a system that drove a new generation of technology and freedom...a movement that celebrated this drive by forming user groups in unknown streets, in unknown cities, and with unknown people.

But as I stood there, doorbell already pressed, none of that was even close to my conscious thoughts. Instead, the brain of one Jonathan E J Bacon was battening down the hatches, preparing for ultimate, unparalleled discomfort as I walked into a place I both did and didn't want to be at the same time.

Then, the door opened and a rather nice chap called Neil welcomed me into his home.

Community is a funny beast. Most people—the kind who watch talent shows on television and occasionally dip bread in oil in an expensive restaurant—don't understand people like Neil. Why on earth would this guy decide to open his home, free of charge, to a collection of strangers who met on the Internet? Why would he want to spend an evening drinking tea and making jokes about something called “Emacs”? And why would he fund online resources like fliers, a mailing list, and a website from his own pocket; start a book-lending service for the group—and even shell out for tea and biscuits?

One person who really didn't seem to understand was Neil's wife. Somewhat bemused, and referring to us as his “Internet friends,” Neil's significant other decided tonight was the night for visiting a long-lost (or possibly ignored) relative, rather than sticking around and faking interest.

Collaboration-Driven Ethos

But Neil is not unusual. At least, not in the Open Source, Free Software, Libre, and Free Culture world. There are many Neils all over the globe, organizing groups, setting up mailing lists, scheduling meetings, and coming together to share an *ethos*: the combined set of beliefs, customs, and sentiment that flows between like-minded people.

In the last 10 to 15 years, we have seen Free Culture in technology, art, and media explode into our consciousnesses. The entire machine is driven by people like Neil: people who volunteer themselves to the concepts of community and togetherness wrapped around such an ethos.

There are Neils outside the Free Culture world, too. They're in church groups, helping the poor and unfortunate; in Neighborhood Watch and Meals on Wheels campaigns, reaching out to those around them; and in public art installations, political groups, and craft fairs. They volunteer, perform, and share their opinions and creativity on anything from aerobics to knitting to yoga.

What intrigued me when I first walked into Neil's living room was the concept of a *collaboration-driven ethos*, although at the time I had no idea what those words meant. What that experience taught, and what that evening inspired in me, was an excitement about what is possible when you get a group of people together who share a common ethos and a commitment to furthering it.

In my world, that ethos has thus far been Free Culture, Free Software, digital rights, and breaking down the digital divide, but it can be as critical as creating world peace or as fanciful as sharing photos of kittens playing guitars on the Internet. The importance of community is not in the crusade, but in how you unify people to march forward together, side by side.

At its heart, *The Art of Community* is a distilled set of approaches and thoughts about how to build community. The book is a collection of experiences, observations, and thoughts from my career and elsewhere. My aim is to bring this grab bag of concepts and curiosities together into one consistent text.

However, it is important that we keep the book in perspective in the wider scheme of your growth as a community leader and organizer. You should mentally frame the content here as a foundation for your own ideas, but remember that practical experience is the real magic that we want to create, with theory merely the glittery jacket and spinning bow tie.

Community is fundamentally a *soft science*. Compare it with, for example, programming. If you want to write a computer software application, you write it in a programming language. These synthetic languages are vessels of logic. They live and breathe in a world where the answer to a question is either *yes* or *no*; there is no *maybe*. In a world where *maybe* does not exist, you can plan ahead for an answer. With community, the importance and diversity of the question is equally essential.

MAPPING OUT THE JOURNEY

In this chapter we are going to be exploring the big-picture attributes that are present in every community. As such, this chapter is filled with a lot of high-level theory that is important to our journey.

It may be tempting to steam ahead and dig into the hands-on content in later chapters, but it is recommended that you read and understand all of the concepts in this chapter first.

This chapter was designed for tea and snacks. Go and grab some, curl up in a chair, and get ready to explore the social schematics of your community.

The Essence of Community

On February 26, 2004, three friends and I released the first episode of a new audio show called LugRadio (<http://www.lugradio.org/>). Although LugRadio will be featured extensively in this

book as a source of stories, all you really need to know about it right now is that (a) it was a loose and fun audio show (a podcast) about open source and free culture, (b) on that day it was entirely new, and (c) we had absolutely no idea what on earth we were doing. Radio personalities across the world were not exactly shaking in their boots.

Recorded in a very small room that I called a studio, but was actually a bedroom filled with secondhand recording equipment, LugRadio involved my three compadres and me opining into four precariously balanced microphones that fed into a computer. Episode 1 was around half an hour long, composed of bad jokes and a book review, and totally unpolished. At the time, it was just new and different. (Little did we know that four years later we would wrap up the show having achieved over two million downloads.) Anyway, enough of the self-congratulatory back-patting and back to the story....

With the show out, we did what many of us in the open source world do—we set up forums, wikis, and channels, and tried to get people together around our new project. The forums went online first (<http://forums.lugradio.org/>), and people started joining.

The 22nd member was a guy called Ben Thorp, known as mrben on the forums. An Englishman living in Scotland, mrben was an open source enthusiast who stumbled onto the forums, listened to the show, and liked what he heard. For the four years that LugRadio lasted, mrben was there every single day: in total contributing over 3,000 posts; involving himself in the chat channel, the wiki, and the organization of the live events; running an episode download mirror; and much more. mrben was there every step of the way, loving every second of it.

The first question is—why? Why does a 30-something Engli-Scot decide to immerse himself so deeply in a group of people he has never met before? What is it that makes him want to spend time away from his friends and family to contribute to a radio show performed by four strangers in a different country? Why would he want to contribute to something with seemingly no financial, career, or other conventional benefit to him?

A cynic could argue that mrben is some kind of socially challenged nerd who can only communicate with other similarly socially inept nerds. Conventional wisdom sometimes argues that anyone who contributes their time freely to something that could not benefit them financially is weird. This was clearly not the case with Ben. He had a job, a wife, and a child. He went to church regularly. When I had the pleasure of socializing with him, I found him a fun, smart, and entertaining part of the group. In fact, at two of the live events, he was a guest in my home. Social deviation was clearly not the answer, or if it was, he hid it well.

The reason why Ben was so involved in LugRadio, why Neil ran the Linux User Group meeting, and why thousands of other community members around the world get together, comes down to one simple word: *belonging*.

By definition, a community is a collection of people (or animals) who interact together in the same environment. Community exists everywhere in nature. From people to penguins, from monkeys to meerkats, the vast majority of organisms exhibit some form of collective grouping. Grouping, however, is a touch simplistic as a means to describe community. It is not merely

the group that generates community, but the interactions within it. These interactions, and the feeling of belonging that they produce, are generated from a distinctive kind of economy: a *social economy*.

Building Belonging into the Social Economy

At this point in our journey, it is clear that *belonging* is our goal. It is that nine-letter word that you should write out in large letters and stick on your office wall. It is that word that should be at the forefront of your inspiration behind building strong community. If there is no belonging, there is no community.

From the outset, though, belonging is an abstract concept. We all seemingly understand it, but many of us struggle to describe it in words. I identify belonging pragmatically: as the positive outcome of a positive social economy. In the same way that we judge a strong financial economy by prosperity, wealth, and a quality standard of living, belonging is the reward of a strong social economy.

An economy is a set of shared concepts and processes that grow and change in an effort to generate a form of capital. In a financial economy, participants put goods and services on the market to generate financial capital. The processes and techniques they use include measuring sales, strategic marketing, enabling ease of access, and so forth. A social economy is the same thing—but *we* are the product, and the capital is respect and trust. The processes and techniques here are different—open communications mediums, easy access to tools, etc.—but the basic principles are the same.

OPEN SOURCE IN THE ECONOMY

Stephen Walli, a prominent commentator on open source in business and review editor for *The Art of Community*, drew some interesting connections between the underlying concepts in a financial economy and how they apply to the open source social economy. He presented these thoughts in his piece entitled “Free and Open Source Software Developers Working for Free (Economics 101)” (http://stephesblog.blogspot.com/my_weblog/2007/09/free-and-open-1.html):

People value their skill sets differently in different contexts, but value them they do. I use writers as an example to explain this to nondevelopers: a technical or marcomm writer may spend 8 hours a day at their paid job, then spend their evenings and weekends teaching ESL classes at the local college, working on a newsletter for their local church/synagogue/neighborhood organization, helping a child with a school project, and writing a sonnet to their significant other (or the next great novel or screenplay). In every case they're using their writing skills; they're just valuing them differently in different contexts.

There's another way to look at it. Not every market involves exchanging money for goods and services. A gem of an economics book (*Reinventing the Bazaar* by John McMillan, 2002, p. 135) points out that well-designed markets, regardless of market type, have a number of things in common:

- Information flows smoothly.
- People can be trusted to live up to their promises.
- Competition is fostered.
- Property rights are protected, but not overprotected.
- Damaging side effects on third parties are curtailed.

Let's look at well-run free and open source project communities in terms of such market dynamics:

- Information flows smoothly—transparency of community, process, code, policy, bugs, discussions.
- People can be trusted to live up to their promises—the project's license is a social contract. Its governance culture is well understood and supported.
- Competition is fostered—what fixes and features are accepted, and which ones don't make it.
- Property rights are protected, but not overprotected—code copyright management and licensing is handled properly in well-run projects.
- Damaging side effects on third parties are curtailed—the point here from the book is that WHEN real damage might be done to third parties, there are ways governments can involve themselves in the market to curtail such effects, whether by defining/enforcing property rights, taxes/incentives, or policy/regulation. The community's license comes to mind.

Individual projects behave as markets from one perspective, and code is currency, the medium of exchange. Just like all economic exchanges, the contributor offers something they value less (a fragment of code solving a particular need) for something they value more (the functioning software package in its entirety). Nobody is working for free in an economic sense.

Social capital is known by us all, but we know it by many different words: *kudos*, *respect*, *goodwill*, *trust*, *celebrity*, *influence*, *supremacy*, *greatness*, and *leverage*, to name a few.

The first known use of the term “social capital” (referred to in Robert Putnam’s *Bowling Alone: The Collapse and Revival of American Community* [Simon & Schuster]) was by L. J. Hanifan, a school supervisor in rural Virginia. Hanifan described social capital as “those tangible substances [that] count for most in the daily lives of people: namely goodwill, fellowship, sympathy, and social intercourse among the individuals and families who make up a social unit....”

Social capital is the collective family of positive interactions between two or more people. When you affect someone positively, it builds your social capital. This could include being generous, helping someone, sympathizing over a problem, or something else. Hanifan identifies the opportunity behind social capital:

The individual is helpless socially, if left to himself.... If he comes into contact with his neighbor, and they with other neighbors, there will be an accumulation of social capital, which may immediately satisfy his social needs and which may bear a social potentiality sufficient to the

substantial improvement of living conditions in the whole community. The community as a whole will benefit by the cooperation of all its parts, while the individual will find in his associations the advantages of the help, the sympathy, and the fellowship of his neighbors.

The meat in Hanifan's description is the opportunity for social capital to "bear a social potentiality sufficient to the substantial improvement of living conditions in the whole community." In essence, if a member of your community has a positive approach to another member, her social capital grows, which has a positive impact on that person and the community as a whole. It all sounds a lot like karma, and it is.

Of course, capital, whether monetary or social, is not the end game. People don't make money for the purposes of just having money: they make money because it allows them to do other things.

This is an important aspect of understanding where an economy starts and ends. Most folks riding the financial economy are not purely greedy numbers freaks who just want a big pot of money; most people who work with social capital are not merely air-kissing, hand-wavey, superficial animals who simply want to name-drop and be name-dropped in the interests of social acceptance. Of course, the greedy and the socially obsessed do exist, but it is important not to use them as a basis for judgment. The economy is not flawed; those people are flawed.

A final point: for an economy to work, every participant needs to *believe in the economy*. Belief is a critical component in how any group of people or animals functions. This can be belief in God, belief in values, or belief in a new future. Whatever the core belief is, the economy and the community can be successful only if everyone has faith in it.

So let's have a quick recap:

- A sense of belonging is what keeps people in communities. This belonging is the goal of community building. The hallmark of a strong community is when its members feel that they belong.
- Belonging is the measure of a strong social economy. This economy's currency is not the money that you find in your wallet or down the back of your couch, but is *social capital*.
- For an economy and community to be successful, the participants need to believe in it. If no one believes in the community that brings them together, it fails.
- Like any other economy, a social economy is a collection of processes that describe how something works and is shared between those who participate.
- These processes, and the generation of social capital, which in turn generates belonging, needs to be effectively communicated.

So far, we have talked extensively about our goals (belonging), the medium of exchange (social capital), and what is at the heart of an economy (processes). We now need to focus on the final component that binds each of these concepts together: communication.

In many ways, an economy is like a flowing river: it never stops, and the flow is critical to its success. Economies never stand still. Every day they change, adjusting to stimuli in the world that affects them. At the heart of how this movement works is *communication*.

The Basis of Communication

Peter Bloch, a consultant on learning, makes an important foundational observation about communication in a social economy: “community is fundamentally an interdependent human system given form by the conversation it holds with itself.” When I first heard that quote, I realized that the mechanism behind communication in a community is *stories*.

Stories are a medium in which we keep the river flowing. They are the vessels in which we not only express ideas (“I was taking the subway to work one day, and I saw this lady on there reading the paper, and it made me think xyz”), but also how we learn from past experiences (“There was one time when I saw David do xyz and I knew I had to adjust how I myself handle those situations in the future”). Furthermore, when the characters in the stories are people in a community, the stories are self-referencing and give the community a sense of reporting. Communities really feel like communities when there is a news wire, be it formalized or through the grapevine.

Not all stories are cut from the same cloth, though. Communities tend to exchange two very different kinds of story: *tales* and *fables*.

Tales are told for entertainment value and to share experiences. They are individual units of experience that are shared between people, and their primary value is in communicating a given person’s experience and adding to the listener’s repertoire of stories and experiences.

Fables are different. Fables are stories designed to illustrate an underlying message. The vast majority of us are exposed to fables as children, and these stories are passed down from generation to generation, each one extolling a moral message to the youth of the day.

Let us now take a step back to our earlier story about mrben joining the LugRadio community. This story was itself a tale that shared an experience that encased many of the concepts we have explored.

When mrben joined the LugRadio community, he identified with the ethos of the show. Then he began to engage with stories: first hearing them on the show itself, then getting them from the community, and finally sharing them himself. As mrben contributed more and more, his social capital started to rise—the community had a lot of respect for him and his opinions. He, in turn, had belief in the community and his own abilities. This objectivity in his storytelling and his general demeanor all contributed to his social capital. As he continued to be a part of the community, his sense of belonging developed. At this point, mrben was living and breathing LugRadio, its community, and its ethos.

The result of this process is a community member with a strong sense of loyalty. Some of the greatest examples of belonging and commitment to an ethos occur when the community is threatened. An interesting example of this was when we released Season 5, Episode 3 of the show and received a rather angry statement from a listener who was clearly agitated at the level of expertise on the show and the generally positive attitude toward Ubuntu (which all of the presenters expressed):

Nowadays I mostly stick to Dave Yates at [lottalinuxlinks](#) who is a genuine linux obsessive, Chess Griffin at [linuxreality](#) who maybe does stuff for noobs but is genuinely knowledgeable about Linux, and the guys at the [linuxlinktechshow](#) because they work with Linux and know what the fuck they're talking about.

mrben, who had spent a few years in the community by then, responded to the criticism using stories to make his point:

I think you'll find that all of the presenters on LugRadio work with Linux on a daily basis. Whether or not they know wtf they're talking about is, of course, a matter of opinion. But the addition of Chris and Adam to the team, both of whom (IIRC) are professional Linux sysadmins, is an influx of knowledge on that side of things. Jono has a long history of working with Open Source and Linux within the community (bingo!) even if his technical knowledge is not at the same level. Aq is a Free software zealot, but is also experienced in web development and usability. I still think it's a good mix, personally.

The Ubuntu thing is an issue, admittedly. But then, LugRadio still reflects my experience of LUGs—the majority of people are talking about Ubuntu. It has become the mainstream desktop distro, and the benchmark that most people mark other distros against. But, IMHO, the recent shows haven't shown an overly Ubuntu slant. Look at this show—you've got an interview with Quim Gil, which is about Maemo, not Ubuntu, the finger of God, which is plain silliness, the software vendors and security issue, which applies across the board, and packaging, which was fairly Ubuntu specific, but could easily relate across to other Debian-based distros, and, as Chris said, he would've talked about RPM if it had been possible.

The "Ubuntu slant" is more about personal usage and experience, rather than a change in the show's direction (which was unashamedly Debian slanted before Ubuntu came out....)

In his three-paragraph response, mrben referred to 12 distinctive points and facts, citing many from existing online material. His response not only sought to convince the original poster of his error, but to demonstrate to the community that the poster was wrong, thus providing a sense of security. By using objective facts, he also spoke with the voice of the community, not just his own opinion. mrben's response was driven by belief in the community, formed by familiarity with stories, and legitimized by a wealth of social capital. Subtle, yet inspiring.

Although the underlying social economy infrastructure in community is compelling, it is important to remember that it is merely a structure designed to deliver a far more exhilarating prospect—*opportunity*. And with that, let's spin back in time....

Unwrapping Opportunity

When I first learned about Linux, I was running a small bookshop in Milton Keynes, in Southern England, and living at home, having taken a year off before starting university. When Simon, the eldest of my two siblings, stayed in our house for a few weeks on his return from the U.S., we frequently spent the evenings talking about computers and stand-up comedy.

On one of those evenings, while I was hurling abuse at my computer, Simon expressed surprise that I used a “Mickey Mouse Operating System.” I was surprised myself. As far as I knew, Windows—Windows 98, at that—was all that existed. Simon told me about something called “Linux,” which I could get for free, from the back of a book.

Armed with my 10% discount, I eagerly snagged a copy of *Slackware Linux Unleashed*, and Simon set to installing Slackware 96 on my desktop computer. Two weeks later, having used guile, cunning, and a soldering iron (literally), and maintaining the alignment of the planets, I actually got the thing to boot. As I gazed eagerly at the screen, ready to experience the next generation of operating system technology, I was confronted with:

```
darkstar login:
```

It was not exactly *Minority Report*.

Simon, being the kind and sharing brother he was, wrote the username and password down on a piece of paper, stuck it to my screen, and promptly sodded off. The following day, he moved out. I was left with a login prompt, some nerves, and absolutely no idea of what to do. So I cracked open the book, threw on a Testament album, and started reading.

It was then that I read about the Free Software community: a worldwide collection of enthusiasts all connected by the Internet, sharing an ethos that software should be free while building a replacement to the Microsoft behemoth that frustrated so many. Piece by piece, this global army provided software alternatives, many of which improved on their commercial counterparts. Back then, Linux was in the dark ages of computing. It was all command-line-driven, devices rarely worked, and to do anything you needed to compile code. Still, this concept of a worldwide community sharing code absolutely fascinated me. I first smelled the sweet aroma of *opportunity*.

Although the reality of open source in 1998 was primitive, the potential within the community is what inspired me to stick with it. To be honest, I was pretty perturbed by the sheer complexity of it all. In those days it was insanely complicated to get a system up and running, and the innards of the operating system were on display for all to see. (These days, as Uncyclopedia [<http://uncyclopedia.wikia.com/>] so eloquently puts it, “Linux distros are so idiot-proof that you can put their install CDs into the floppy drive upside-down and it will still work” [slightly edited for a family audience].) Back then, we all knew that life with Linux was a lot harder than it needed to be, but the strong sense of underlying opportunity helped spark the imagination to put up with that complexity for the potential of a better future.

There is an important connection here in which *imagination* and *opportunity* are close friends. Imagination offers the mind a vision of how things could be. If there is a viable path toward this future, we build a sense of opportunity. If there is no viable path, we enter the world of fantasy.

Linux, and the possibility of it becoming a prominent operating system, was by no means a fantasy. The rails were on the ground. The community just needed freely available tools and communication channels to gather the materials, build the train, and put it on the track. In the case of Linux, this manifested in three primary areas:

Open communication

With an open community and publicly visible and accessible communication channels, anyone can join the community and meet hundreds of thousands of other community members just like them.

Licensing of work

Every contribution to the Linux community is licensed in such a way that it benefits the entire community. The fair licensing of all contributions adds a strong sense of confidence to the security of the community.

Open tools

Anyone with an Internet connection and a computer can contribute. All of the development tools and documentation are entirely free and open to access. This provides a low barrier to entry, and lets new users play with the technology.

Although these elements were essential at the birth of Linux, it is not open communication, licensing, and tools that generate *opportunity*. These elements merely made it possible to build a world-class Free Software operating system. Opportunity is born in a sense of *belief*.

Belief is a critically important human function. Whether your belief is in an all-creating god, in a family member's ability to achieve something for herself, in a better future in your neighborhood, or in the reliability of a restaurant guide, belief is what gives us hope for the world around us. Belief can also make human beings surprisingly resilient in intensely difficult and uncomfortable situations.

One example of this is an incident that occurred a few years back. Every year, as part of LugRadio we host a face-to-face get-together called LugRadio Live (<http://www.lugradio.org/live/>), which is a very different style of conference. We have worked hard to deliberately make the conference fundamentally a community event. Equality between commercial vendors and the community is a key attribute, and we deliberately set a low cover charge to keep it accessible. In addition to this, we have worked to produce a very informal and inclusive atmosphere, inspired largely by music events. (Many referred to LugRadio Live as a "rock conference.")

LugRadio Live has carved out something of a reputation for being different, and each of its participants has been very keen about advocating it and its formula. There was a strong sense

of belief in the event—an event that was distinctively community-oriented and -driven, open to participation, and available to all.

With LugRadio Live scheduled for July 22–23, 2006, everything was going to plan. The speakers and exhibitors were sourced, the schedule was in place, the social events were arranged, and the crew and community were ready. Everything was great until the evening of July 18, when I received news of an impending rail strike. The strike was planned for the full weekend of the event, with every rail link going down. The country would be completely inaccessible by train.

I have never experienced such anger and frustration. For about an hour, I transformed into an ultra-conservative right-wing anti-union crazy, and I stomped around the house, venting in the direction of my computer screen. We had spent six months of feverish planning and hard work, and this union decided that their problems were more important than anyone else's, and it was entirely reasonable to take the country down. I, for one, was not a happy bunny.

But, as my fellow organizers and I seethed on the phone, the community was already doing its thing. Forum threads appeared instantly to keep people up-to-date on the strike, blog entries were drafted, a nationwide car-sharing scheme kicked into play, and speakers and exhibitors were notified. While all of this was going on, I was on the phone tearing a strip out of both the union and the rail organization for their decision. Fortunately, the strike was called off a few days later.

What stunned me was just how mobilized the LugRadio community was. The community saw a threat to something they felt invested in, and reacted as a team to cover all the bases and try to limit the damage. Without any prodding from us, they made things happen. In a time of such panic and frustration, that community wrapped around each of the organizers like a comfort blanket. It was one of the most inspiring examples I have ever seen of a community coming together, driven by a belief in something we all shared.

Where belief gets exciting is when it is combined with that friend of ours from a few pages back: opportunity. Belief in a shared crusade—and a sense that the tools and opportunities are available to achieve that goal—is an intensely liberating feeling. People get a sense that they have control over their own destiny.

An example of this was the election of Barack Obama as president of the United States. Building up to his victory, the U.S. was facing difficult times. Led by a president who many lacked faith in and faced with a global economic crisis and a complex set of foreign affairs, the U.S. had a lot to deal with, including a growing sense of cynicism among its people. Many Americans had lost faith in politics and pride in their country. As Barack Obama stepped up as a candidate for the presidential election, he instilled a sense of belief and opportunity that inspired his followers.

When people feel that they can achieve a dream, it builds an incredible sense of liberation and a willingness to step up to the plate. People become very committed, very quickly. We saw this

in droves throughout the presidential election. Thousands of people across the country took to the streets to tell the world about Obama.

Around that time I had kissed chilly England goodbye and relocated to sunny California. The Bay Area was a particularly fascinating place to be—people setting up tables, selling stickers, knocking on doors, and making phone calls. It seemed that one in three people on the street was wearing an Obama t-shirt.

Whether Obama was the right man for the job is the topic of a thousand other books, but he had the ability to define belief, opportunity, and liberation in a language that a nation could understand. His inspiration—and his army of passionate Obamaniacs—sealed his place in the Oval Office. This in itself was an incredible exercise in building, energizing, and inspiring community, and regardless of your political inclination, it was a stunning feat.

A Community Manager: Becoming the Community

So far, we have explored some of the high-level concepts and architecture behind community and how it is structured. It is these concepts, such as social economy, belonging, belief, social capital, and communication, that help to draw the outline of the picture. We will use the rest of the book to color in the details.

Before we move on, though, we should spend some time profiling the artist holding this palette of colors. What are the skills required to draw the picture? What attributes will help us put the right colors in the right places? What do you need to build really great community?

Metaphor aside, community building is a genuine art form. Like any art, there are attributes and characteristics that define someone as an artist, but every artist has his own “special sauce” that makes him unique and different. Every one of you lucky enough to have your nose buried in this book will have yours. Although I talk about some common characteristics all community managers should have, always strive to find your own approach.

Cracking Open the Personality

When exploring the mind of a community manager, we can break the attributes you need into two broad areas: *personality* and *strategic traits*. The latter, strategic traits, are skills, perspectives, and capabilities that help you to organize complex problems into logical boxes, develop action points and a plan, and implement those intentions in a controlled way. Strategy is a large and complex subject, and we will discuss it extensively in the next chapter. As such, I will defer exploration of this part of the community manager’s brain until later. For now, let’s talk about personality.

My wife and I often joke about how I respond when someone asks me what I do for a living. I used to describe my job as “managing a worldwide community of volunteers,” which sounded rather accurate and complete to me. My wife was less impressed. She rightly nudged me and

suggested that made it sound like I manage a large online forum of very weird Japanese animation fans with a worrying obsession with tentacles. From that discussion onward, I have tried to summarize what we community managers do in one sentence. The best I have come up with is:

I help to enable a worldwide collection of volunteers to work together to do things that make a difference to them.

I know, it needs work. Send better suggestions to the usual address....

Twenty of those twenty-one words are really just filler around the word that I *really* think describes what we do: *enable*.

Our function as community leaders is to enable people to be the best they can in the community that they have chosen to be a part of. Our job is to help our community members achieve their greatest ambitions, and to help them work with other community members to realize not only their own personal goals, but the goals of the community itself.

Trust Is Everything

At the heart of this enablement is *trust*. As we have already discussed, community is fundamentally a social economy, and its participants build up social capital via their contributions. With social capital being, by its very nature, a product of social interaction, trust is critical. If people in a community don't trust you, you will be met with caution and you will struggle to build your social capital.

For community leaders and managers, trust is a critical component in gaining the support and confidence of your community members. Earlier, we explored the example of Barack Obama stepping forward to enthuse a nation in turbulent times. Part of the reason why those times were turbulent was a significant lack of trust in President George W. Bush. When trust vanishes, words and promises lose their meaning. When trust is present, words and promises flourish in a world where they have purpose and potential.

Trust, though, is not something you can learn. You are either trusted or you are not. As my father-in-law said to my family one evening over dinner, "Live your life honestly—if you don't, you always have to remember to not be yourself." His words teach an important lesson: when trust is implicit in every step you take, you can always be confident in the transparency and openness of your actions. This is the most important aspect of community leadership, and of life itself.

Part of the reason why trust is so critical is that, as a community leader, you want to be emotionally close to everyone in your community. You want everyone in that community to think of you as an accessible, approachable, sensitive person, and trust is required for any of these roles. People will approach you for advice, for guidance, to discuss personal issues, to handle conflict, and more. Many of these situations will be complex, and will require a significant level of sensitivity and confidence.

The Value of Listening

Part of achieving that sense of trust and confidence is having a firm foundation of understanding and patience. You should be aware right now that some people are going to frustrate you. Some people will be too quick to act and opine on a subject, and some will be too timid and reluctant to put their hands up. Some people will obsess about the wrong things and regularly produce what appears to be a tempest in a teacup.

But then again, some people will inspire you with their sense of responsibility, their ability to react to situations with grace and elegance, and their willingness to care for the community. As a community leader you will experience all sides of human nature, from strength and innovation to weakness and uncertainty. Whatever you hear from your community, you should endeavor to be the best *listener* that you can.

When you can demonstrate trust and the capability to listen, your community will develop respect for you. They will be there to listen to you, work with you, to stand side-by-side with you in your battles and become a large extended family that you can rely on.

This respect has an important function in reinforcing belief in your community. When community members have responsive positive interactions with community leaders, it makes the community feel more inclusive, which generates belief and, importantly, belonging.

Respect is a wonderful gift, and you should cherish it and protect it at all costs. Getting that respect back after you lose it is a near-impossible task.

Avoid Ego, or Others Will Avoid You

Just as the right kind of inspiration can cause lasting effects, wrong decisions and approaches can cause lasting damage.

The biggest risk that can face any community leader is *excessive ego*. Unfortunately, ego is something that plagues a lot of people who assume a form of leadership.

One rather fun side effect of an influential position is *fame*; people know who you are, they read your words, and they listen to your opinions. The growth of the Internet and online communities has made it easier than ever to bolster your ego. Google searches, social networking websites, news feeds, alerts, statistics tracking, and more make it easier than ever to find out how many people love (or loathe) you. While this may seem incredibly flattering at times, strive to get your audience in perspective. The world is a big place, and you have some incredible contributions to add to it, but always aim to strike a balance between valuing your contributions and disappearing up your own arse.

Unfortunately, I have seen ego claim too many victims. I have seen community leaders who have commanded incredible respect and adoration from their legions of fans, but have washed it away when they assume a sense of *entitlement*. In any community, entitlement is an enemy: it values the person over the contribution, creating unrealistic expectations about how people

should be treated. People with this sense of entitlement typically identify the wider contributions of the community, but see their fame as a red carpet that others don't have access to. You can avoid the wrath of ego by always remembering that in your role as a community leader you are responsible *to* the team, not *for* the team.

One such example of this was a guy (who shall remain nameless) who got very involved in a large open source project. Although not a developer, he became a very public face in the project. He talked very loudly and very enthusiastically about the project and its work, and he actively participated on the mailing lists, discussion channels, and in blog conversations. His contributions were certainly valid: he acted as a champion and enthusiastic voice in the project, encouraging others and getting people pumped up.

As time went on, his name started to be mentioned more than the project. He started performing speaking engagements around the world, and his blog became filled with in-jokes shared with his well-respected, cool friends. He was very much part of the perceived "cool club," if there is such a thing in ultra-hardcore technology circles.

As his fame grew, so did his inbox. He clearly felt that other people, with "less important" tasks, should respond to the email, as the demands on his time were substantial. But he was also riddled with insecurity, and didn't want to loosen his grip on the project. Things stopped getting done. People started getting frustrated, but these frustrations were largely stifled by the community. No one really felt like they could speak out or criticize: if you spoke out, you sounded like the only voice who did not respect and appreciate the contributions of a "rock star." What a pickle.

Of course, this situation could not go on forever. People started to share their frustrations at conferences, in chat channels, and in other arenas. Before long, annoyance and disrespect had become public, and ego had claimed another victim.

The most frustrating part of this tale is that the guy in question was and still is a good guy. At no point did malice or ill will drive any of his decisions or actions. He simply got a little too big for his boots and lost track of how to manage his responsibilities.

I would like to pretend that these cases are rare, but unfortunately they are not. They happen every day, in a range of communities around the world, and you should always have this risk at the forefront of your mind: don't be *that* guy.

Theory Versus Action: Action Wins

A subtler side effect of ego is one that doesn't threaten reputation so much as how you prioritize what is important. The threat is based on a sense that your opinion, approach, and perspective are the only ones with merit. While arrogance is one outcome of these elements, a much more subtle risk that can bubble to the surface is becoming *too focused on theory*.

Theory has an important place in community leadership. Heck, I have just spent the last several sections talking about social capital, belonging, economy, belief, and other theoretical

dispositions. However, you will note that the hardcore theoretical content is confined to this chapter. The emphasis of our work should be on getting on the front lines and trying out ideas instead of burying our heads in a book. Sure, read and learn, but use reading and theory to help you decide where to focus your practical efforts. The most critical lesson here is that you should never replace practical experience with theory.

Part of the reason why I have filled this book with so many examples is that I believe that the most appropriate and effective form of teaching theory is sharing stories and experiences. I would much rather tell you a story about something that I experienced or heard than fill your mind with social science definitions and terms that would be enormously helpful in a game of Buzzword Bingo but of limited use on the ground.

Unfortunately, some folks have something of a love affair with theory. Many of these people write extensive blog entries, give very generic (although well-meaning) presentations, and often seem to think that their primary role is to impart knowledge to others and sound as wildly academic as possible. But there is no secret ingredient in growing community. What makes a great community leader is experience: trying new ideas and concepts and learning from the successes and mistakes.

Of course, theory does have its place: it can help us to see, analyze, and deconstruct the things that are in front of us. I have absolute respect for anyone who teaches others their art, but the teaching really needs to be secondary to getting out and being there with your community and helping to lead and inspire them. Use theory as a means to see the shapes in the chaos, but always ensure you focus your primary efforts on making the chaos less chaotic.

Becoming Yourself

Before I wrap up this part of our journey, I want to go back to our discussion earlier about finding your “special sauce.” In the last few pages we have talked about many of the admirable traits in a good community leader (trust, patience, respect, ability to listen, etc.) and some of the problematic traits (ego, control freak, theory obsession, etc.), and you may be a little lost in how you can get the balance right. Fortunately, the solution is simple.

Be yourself.

Your “secret sauce” is you. Your personality is the greatest asset that you have. Earlier we talked about how trust is the most critical component in being a great community leader. If you try to become someone who you are not, you will sacrifice that most important of traits. Be yourself. Identify your own traits, celebrate the good, and learn to improve the bad, but always be yourself; it will put you in good stead.

I learned this lesson when we started doing LugRadio. At the time I was a journalist: I wrote for around 12 magazines and also wrote a number of columns for various publications. As part of my work I would write my articles and features in a text editor, and then have plenty of time to edit, choose my words carefully, refine my language, and perfect my tone. I could

perform plenty of research, ensure my citations were accurate, and pull from a worldwide library of quotes and stories that would illustrate my points effectively. In a nutshell, being a journalist allowed me to take my voice and ensure it was as refined as possible.

We then started doing LugRadio. From day one, LugRadio was set to be a shoot-from-the-hip, loose, opinionated social exploration of Free Software and Free Culture topics. There was no editing. There was no censorship of opinions. Every word that came out of our mouths was committed to history. It was ballsy, it was controversial, but importantly, it was *us*. It was the honesty of LugRadio that we were all so proud of.

Although my writing and journalistic work was honest and I never painted an inaccurate picture of my voice or viewpoints, the editing phase of writing added a little more sheen and more opportunity to remove any controversial aspects to my work. With LugRadio, every word that would flow out of my mouth into people's ears was much less formalized: it was me in my birthday suit, no holds barred.

For about two days I really worried about this. Was LugRadio going to affect my career as a journalist? Would people think less of me because of my participation in this hugely opinionated tech audio show? Was I closing off potential opportunities if people heard me for who I really am?

One evening, while at my karate training, sweating and performing second kata, I decided I didn't care. I had always promised myself that I would be myself in my professional and personal life. I would never cover up my perspectives, my views, my interests, or my ambitions. I was always keen to be professional and respectful of the situations I was in, be they more formalized occasions or low-key social functions, but fundamentally the heart of my approach would be my own. This was the first test of that approach, and since then I have never looked back.

I believe that this commitment to being who you are is critical to being a great community leader. My father-in-law's statement—"Live your life honestly—if you don't, you always have to remember to not be yourself"—actually goes much deeper. Not only should we aspire to lead a good and principled life, but if we live it in a way that is honest to who we really are, we never need to worry about maintaining the illusion. My father-in-law has taken that approach, and so has my own father, and I have learned from them and have incredible respect for their methods.

If you take this approach in your work and manage to do a good job engaging and working with your community, you will never need to worry about trust, transparency, or respect; people will know that the words that come out of your mouth are your own, that your opinions are your own, and that your advice is your own.

This is particularly important when you are a professional community manager. When I started working for Canonical as the Ubuntu community manager, I expected people to worry that Canonical may be pulling the strings that make me dance. It has been very important to me that I demonstrate to any community I am involved in that my commitment to it is genuine,

and I am as willing to stand up against the commercial sponsor as I am willing to stand up against the community. Trust and responsibility is a two-way street.

Moving Forward

In this chapter we have grabbed a piece of paper and sketched out the primary outlines of community. We have discussed some of the high-level mechanisms of how a community works, explored why people build community, and analyzed what kind of attributes exist in great community leaders. The aim of this chapter was to build out an overview that we will delve into in more detail throughout the rest of this book.

The next step in our journey is to build a plan for our community. Although a community is a fairly organic collection of people and processes, we can architect a surprising amount of structure around it. This gives us an opportunity to more easily map out the road ahead.

I don't know about you, but I am ready to roll. Let's get to it....

Planning Your Community

“We must all hang together, or assuredly, we shall all hang separately.”

—Benjamin Franklin

MY BEST FRIEND IS A GUY NAMED STUART LANGRIDGE, WHOM I CALL “Aq.” (He was nicknamed “Aquarius” in an online user group devoted to a fantasy author, for reasons that make my eyes glaze over when he tries to explain them.) I first met Aq in Wolverhampton in Central England, where I’d moved to go to university. We became fast friends.

With my curiosity initially piqued by Neil’s Linux User Group, I was eager to form my own: the cunningly named Wolverhampton Linux User Group. Six months later, Aq wandered into a meeting, complete with now-trademark bombastic personality.

Over the years, Aq and I shared many a pint and a curry, debating and discussing every imaginable topic about Free Software. No subject was out of reach, and we relished in each other’s passion for the subject. We also relished in the opportunity to prove each other wrong. These debates inspired many projects: one of them was LugRadio.

Throughout the life of LugRadio, Aq and I debated how we—or more specifically, I—recorded the show. As the resident musician in the fabulous foursome, with a room full of recording equipment, I handled recording and editing, using Mac OS X and the Cubase audio production system.

Yes, folks, you read that right: LugRadio was a show all about Free Software but recorded on a proprietary system, with a proprietary application. Fortunately, the community took good

stead to remind me of my alleged “freedom hating” pretty much every day. Lucky me. Unfortunately, I didn’t want to spend my life engaged in the rocket science that was Linux audio engineering. I love to play music, not spend my days thinking about which sample size I should set my software to.

The debate raged on, and I was getting increasingly sick of the discussion. Something had to change.

One evening at Aq’s house, we were drinking tea and debating open source like normal. One additional area in which my cantankerous friend and I shared a deep interest was interaction design: how to make products and interfaces easier. Thus the topic of Linux audio recording arose.

Our debate was more akin to resounding agreement. We both cited example after example of poor interface decisions: methods of interaction that relied on redundant questions, complex assumed knowledge, and other travesties. My solution was to start from scratch. So we did.

We thought it would be fun to totally rethink audio recording. We sat down with paper and pens, and more cups of tea, discussing and debating until 4 a.m. When I got home and dragged my drained body into bed, my laptop bag contained three pieces of paper outlining an entirely new approach to audio recording.

Despite our brainstorming efforts, I just didn’t have the time or knowledge to write an audio editor. I could have used my meager audio programming and development skills to produce a rather cruffy attempt, but it would have been of little use, and I was already intensely busy. Despite this lack of time and skill, I didn’t want our designs to languish in obscurity, so I drafted some mock-ups and wrote a lengthy blog entry explaining how they worked. I informed the LugRadio community and expected silence: the world moving on, our designs unnoticed.

A few weeks later I wandered onto the LugRadio forums and noticed that some code had been committed to a repository. I downloaded it and it looked like an incredibly simple first cut of the interface that existed in my mock-ups.

I was stunned.

So was Aq.

The author was a rather nice chap called Jason Field who had a passion for coding and Linux. I immediately emailed him to make contact. His simple contribution had inspired me to consider the project further and to see whether the designs were really possible to build. He said yes.

The LugRadio community members were equally intrigued by the story about this new audio editor: they nicknamed it *JonoEdit*. I was flattered, if a little embarrassed.

It was time to get the machine rolling. We set up a code repository, a website, mailing list, and a bug tracker, and scheduled regular meetings. We organized hack days, bug squashing parties,

and online discussions to plan and decide on major architectural decision. New people joined the team, including Laszlo Pandy, who became the subsequent leader of the project.

A little later in the development, the project got an important name change; I wasn't keen on *JonoEdit*. After asking for suggestions, Steve Parkes, one of the original LugRadio presenters, suggested *Jokosher*. The name was formed from "Jo" and "Kosher," which he claimed as "No Bacon," thus constructing my name. Again, I was flattered. It felt weird, but the team liked the new name, so we stuck with it.

Everyone worked hard. We spent long evenings writing code, debugging, fixing bugs, and writing documentation. Piece by piece we built not only an application but also a community. We developed a sense of unity, and we started to become a team.

Eventually, after months of work, we made a first release. From a few ideas, expressed with my amateur-grade design skills, we built something that people could touch. Today, although I have stepped back to work on other things, *Jokosher* is a thriving project.

Most Free Software projects form from one person scratching an itch. They write code and release it; if it scratches other people's itches, collaboration begins. *Jokosher* was different. It existed entirely on paper before it did in software. The application was rooted in a new approach to interaction design, so having a documented design was essential. The design and accompanied specification of the interface acted as a reference from which to build the software.

What this experience taught me, entirely by chance, was that the speed and success of a community has a direct correlation to strategy, structure, and planning: even a simple set of mock-ups can help drive progress in the right direction. Communities that appear more by accident than by intention tend to be slow to develop and mature. Organized communities thrive because structure provides a sense of worth, conviction, and oversight. A strategy will make things happen for your community.

Planning for Success

As you will discover throughout *The Art of Community*, I like to introduce concepts in a very specific way. First, I present a high-level discussion of the top priorities and get the basics down. Next, I focus on the details and flesh out the subject. This approach makes introducing the subject more akin to lowering yourself into a warm pool on a sunny day, as opposed to hurling yourself into an icy river in Finland.

Our goal in this chapter is to explore how to build a strategy for our communities. We first explore four foundations within community. Each of these houses the underlying details that we explore throughout the rest of the book. Inside these foundations are teams, the vessels of community, which we later crack open to see how they work. Next we define our mission, objectives, and goals and build them into a final strategic plan. We are going to cover a lot of

ground in this chapter, so grab a cup of tea (I would also recommend a HobNob to dip in it) and get comfortable.

With many areas of discussion, we need to pick out the key concepts and have them close at hand. It is easy to forget or skim over some subtle yet important aspects in community growth. To make this simple, we will add them to a TODO list as we uncover them throughout the chapter.

-
- Community TODO List
-
- Example item.
 - Example item.
-

Later in the book we will revisit this list and use it to form the basis of our strategic plan.

Community: The Bird's-Eye View

Building a strong community is an exhilarating and rewarding prospect, but getting there can be complex. You only need to look at this book's table of contents to see that the subject is hugely diverse and detailed.

We first need to step back and understand our broad aims. When we wake up and decide we want to build a community, what do we want to achieve? Aside from the goals of the community itself, be it building a software project, changing a political system, or whatever else, how do we inspire a collection of people to march forward as one? How exactly do we unite a people? Well, we can explain this using...dots.

Yes, dots.

I know what you are thinking: you are mad, Bacon. Dots seem a little simple. Sure they are, but they're different in many ways. Different colors, different sizes, different shapes. You may not know this, but dots have oodles of personality. Take these three, as shown in [Figure 2-1](#).



FIGURE 2-1. The Dot family. Don't they look cute?

Readers, I would like to introduce you to the Dot family: John, Pauline, and Ken. Although similar looking, each has a very distinctive personality and skills. What brings them together is the same passion: creating a completely new and original social networking website, built for dots, by dots, called...DotBook.

John, Pauline, and Ken contribute in very different ways. John (left) is a programmer, Pauline (middle) loves to write documentation, and Ken (right) is rather fond of drawing. I know, he looks like the arty type.

John, Pauline, and Ken are not alone. There are lots of dots like them out there on the Internet....

The problem, as you can see in [Figure 2-2](#), is that all of the Johns, Paulines, and Kens, and their respective skills, are scattered all over the Internet. None of them really know each other. They are all united by the same goal, but they're not working together.

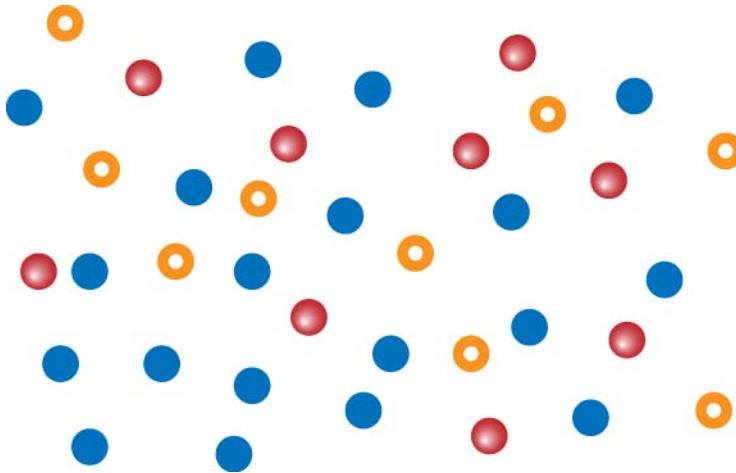


FIGURE 2-2. The Dot family is not alone

A community is just like [Figure 2-2](#): a scattered collection of dots. One of our first goals is to bring these connected areas of interest together into well-formed teams. It turns out that when Johns are able to talk to other Johns, some interesting things can happen. These teams are important containers of expertise and interest inside our wider community (see [Figure 2-3](#)).

If we think of a community as an interspersed group of dots huddled together over a shared interest (e.g., protesting a ludicrous law, discussing a topic, building an operating system), teams are the smaller subgroups typically based around a primary interest or skill set (e.g., advocacy and documentation) that helps forward that shared interest.

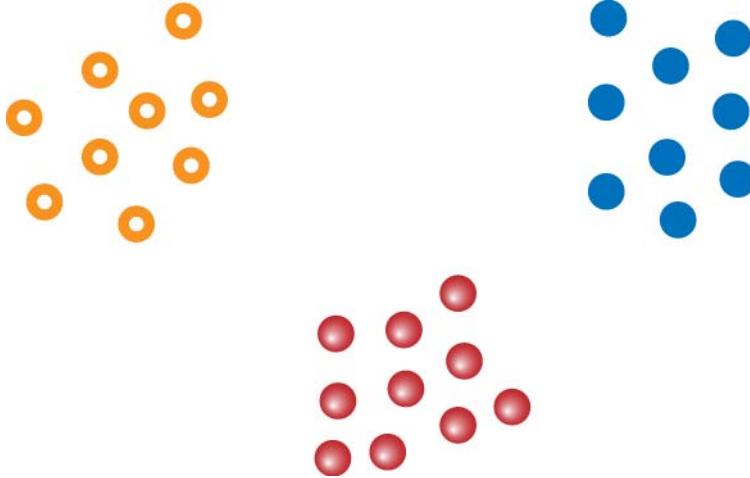


FIGURE 2-3. Uniting people by their interests and passions is an important first step in growing a strong community

As an example, the Ubuntu community has a shared interest in building a Free Software operating system. To do this there are many smaller groups who perform translations, produce packages, write documentation, test software, advocate Ubuntu, and much more. These are the teams that we seek to build: to break our wider community into smaller, more manageable chunks. Each of these teams is united by solving a part of the grander aim of the community.

Teams are an essential construct in community building: they are not only the containers in which your community grows, but also convenient units of capability that help you to strategically understand and structure your community and find out what it's capable of. When John Dot meets other dots who share his interests and get excited at the same opportunities, teams also become containers of *belonging*.

Although teams have a primary focus (typically a skill, such as art or documentation), you should not be too rigid in that focus. Every team will have members who are interested in this primary function but who will also bring other expertise and insight to the team. As an example, in a software community, it is hugely valuable for the art team to also have members with capabilities in programming: they can often expedite getting art contributions implemented in the technical development of the application. As such, encourage and optimize for membership based on the primary focus of the team, but also celebrate and make use of the other skills of your members.

Teams offer a wealth of opportunities and benefits in building a strong community, and we discuss them extensively later in this chapter. Let's first add this goal to our TODO list.

Community TODO List

- Identify how we can divide our community into teams.
-

Although constructing teams is valuable, it is not enough. As a unit of capability, a team is still part of a wider community that strives for a common goal. We need to ensure that our teams fit together like a completed jigsaw puzzle. Communication, ideas, and stories must flow freely between your teams, as in [Figure 2-4](#).

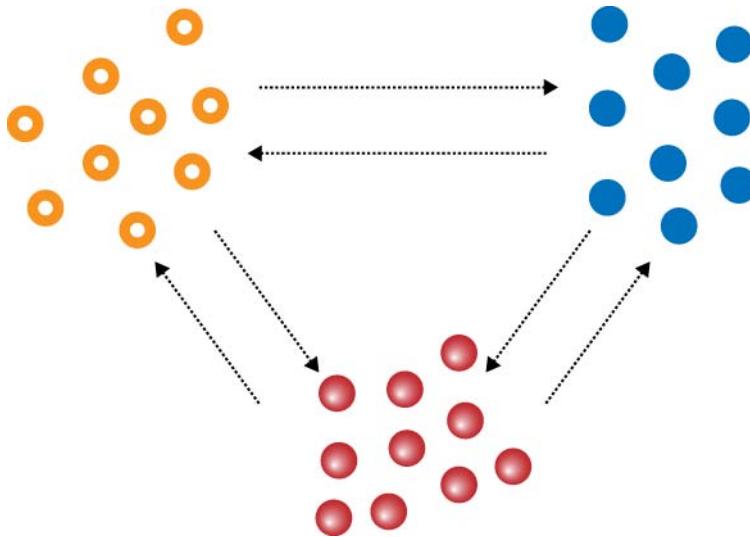


FIGURE 2-4. Communication between teams is essential

The flow of communication between teams is a lot more complex than you would first imagine. How can you ensure an easy flow of ideas and progress between two teams who focus on different parts of the community? How does your art team communicate well with your development team? This opens up a huge array of questions. Even with the three teams in [Figure 2-4](#), how do they communicate? What mediums do they use? How do they deal with geographic and time zone issues? How do they report their interactions to the wider community? How do they track progress? How do we understand how different teams work together? Not an easy problem to solve.

These questions are not merely about how two or three teams communicate. They get to the heart of the ethos of the community as a whole: the standard for how teams are structured, how they behave, and how they communicate.

As I mentioned earlier, although your teams have a primary focus (such as translations), there will be many other skills present in your teams, and many people will be in multiple teams. We need to not only foster effective communication between teams (such as regular meetings, progress checks, and shared communication mediums) but also to make use of people who have their feet in multiple teams. They can be the glue that sticks teams together. These people should absolutely be on your Christmas card list.

This topic is part of *governance*, and is so large and critical to community success that I have dedicated [Chapter 8](#) to it later in this book. Let's ensure that we make a note of team communication on our TODO list, even though we're not looking at how to build this until later, when we discuss governance.

Community TODO List

- Identify how we can divide our community into teams.
 - Ensure that teams can communicate clearly and effectively.
-

The next area to focus on is contributor growth. We like dots like John, Pauline, and Ken, and we want to encourage more of them into our community, as we can see in [Figure 2-5](#).

When it comes to new contributors, we essentially seek to satisfy two primary desires: *capacity* and *diversity*.

With *capacity*, our goal is to provide more hands on deck. More (coordinated) hands generally mean that more things get done. Most communities have somewhat audacious goals (which we will discuss later), and these goals almost always outstrip the resources available to implement them. This bottleneck can cause burnout (a topic we discuss extensively in [Chapter 9](#)), but more immediately it generates a need to find more resources.

Attracting members to your community is one task, but attracting a diverse range of contributors is an entirely different ball game. Although not critical in a community, diversity has huge value: different skills, cultures, perspectives, attitudes, and experiences make for a richer community experience.

Later in this book, in [Chapter 6](#), we explore how to attract members to your community. This requires not only making your community attractive to prospective members but also offering them an effective workflow so they see their contributions put to use without too much hassle. What challenges do your members face in contributing? How can we make the barriers as low

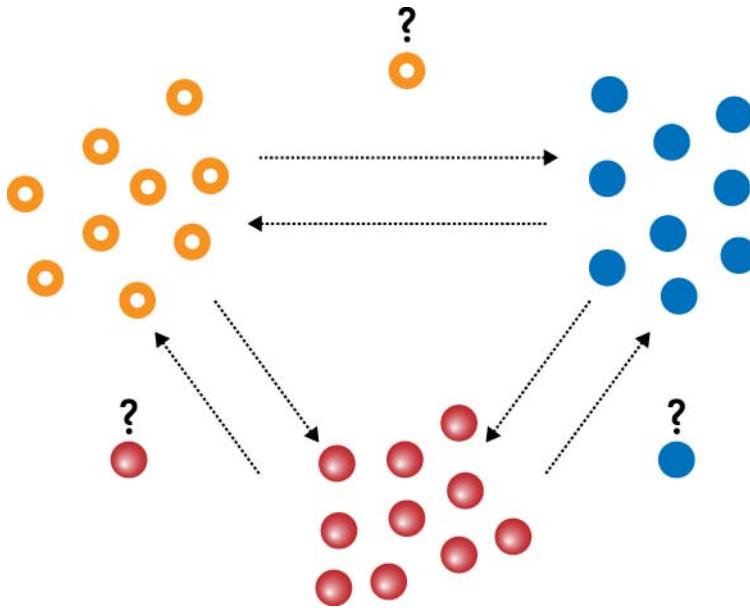


FIGURE 2-5. Bringing in new contributors is an essential task

as possible to new blood while still bringing in the right skills? What are the right skills? Let's add this important goal to the TODO list.

Community TODO List

- Identify how we can divide our community into teams.
 - Ensure that teams can communicate clearly and effectively.
 - Attract a diverse range of contributors to our community to get involved and contribute to our goals.
-

The final major step in building a strong community is in building a positive environment, as shown in [Figure 2-6](#) (we will focus on building a strong environment booking in [Chapter 4](#)). Your community should feel inspired, engaged, and thrilled by the opportunity to achieve the goals they dream of.

Environment plays a huge role in everything that we do. Every element of our environment affects our perspectives, emotions, and expectations.

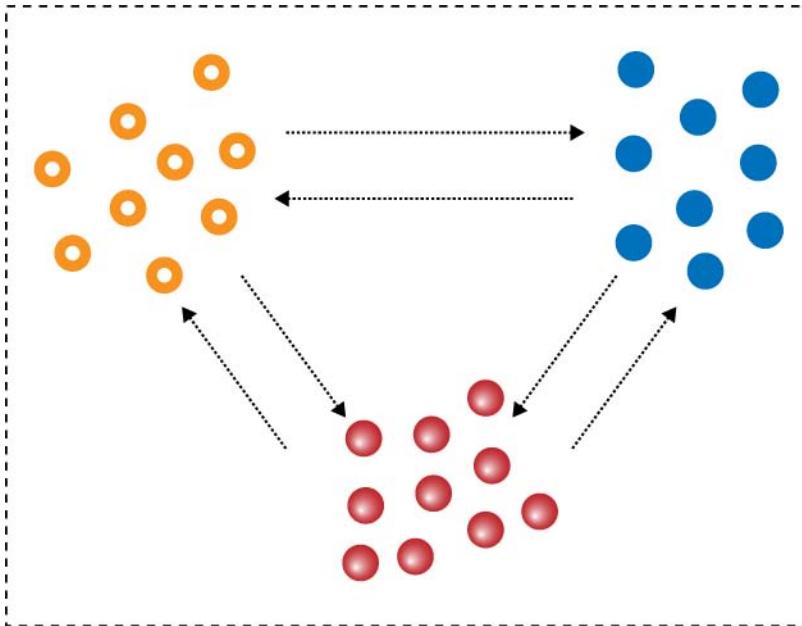


FIGURE 2-6. Build a strong environment, and you will have a strong team

Consider a conventional environment such as a neighborhood. Many environmental attributes can affect perception, with one such example being the feeling of safety. Examples that affect this feeling include the size and style of the houses; the types of cars on the street; the residents, what they wear, and their body language; the lighting of the street at night; the background noise; the amount of traffic; and more. Compare and contrast a seemingly friendly, social neighborhood with kids walking around and people talking in the street with an unfriendly, closed, and restricted place with few people talking and sharing their lives together. Money is not the decider here: super-expensive gated communities are often as socially sparse as very poor inner-city tower blocks. Fundamentally, both have the same core pieces—houses, roads, cars, streetlights, residents, etc.—but the environment hugely affects perception. The mental perception of a small country town adorned with its small streets, tea shops, and abundance of elderly people playing chess, and an inner-city block with its square buildings, large gates, and commuters rushing to work are very different.

Environment not only affects perception but opportunity, too. In a neighborhood that is perceived as unsafe, residents are less likely to interact with each other due to fear. In a community that feels safe and welcoming, local groups, ad hoc conversations, and other interactions thrive. These considerations don't just apply to geographical communities; your online environment is just as important.

Let's add environment to the list and take a quick recap of our key issues thus far.

Community TODO List

- Identify how we can divide our community into teams.
 - Ensure that teams can communicate clearly and effectively.
 - Attract a diverse range of contributors to our community to get involved and contribute to our goals.
 - Build an environment conducive to our wider goals.
-

Teams: The Building Blocks of Belonging

Every goal on our TODO list is key to strong community, and although you may not realize it, every item on that list also has a commonality: *teams*. Let's explore this in more detail.

Teams are the building blocks of community structure: they are the Lego bricks that build an army of volunteers united by a mission. But as building blocks, they fit together in many different ways, with countless variations and possibilities in how they are constructed.

For us to understand community, we need to understand what makes teams tick. We are now going to spend some time focusing on the essential ingredients that should be present in all teams. These ingredients are the best-practice elements that make teams strong, welcoming, and productive. We will explore these foundational ingredients now, and then later in the chapter we will determine which specific teams will comprise our community.

Finding Your Place

Everyone has dreams. We all have ambitions and experiences that we lust after, and we are all guilty of using the opportunity of a rainy day to stare out the window and let our minds take us toward these grand visions, be they flying into space, making millions of dollars, or playing bass guitar in front of thousands of people.

Adam Sweet's dreams were different. He *had* played bass in front of thousands of people. The young, spiky-haired, cigarette-smoking bundle of energy got a band together with his friends and managed to score a record deal at the tender age of 18. Known as Passion Star, the band went on to play the Radio One Roadshow: a UK-wide tour composed of established artists and new talent.

Though it sounded fun, Adam had had enough. He wanted normality. He wanted a house, a girlfriend, and a regular life, not life in a small bus with four hygienically challenged musos.

Adam still wanted a passion, though (without the star). He needed an interest to wrap his natural sense of curiosity and exploration around. So he bought a computer.

Even though it was the year 2000 and he was 24, it may surprise you that Adam had never owned a computer before. Immediately puzzled with his new friend, he bought a magazine to understand it better, and said magazine happened to feature the Linux operating system on the cover. Not knowing any better, he jumped in and installed it. Of course, it was an epic disaster, as Linux generally was back then. While trying to find help, Adam stumbled across his first community, which doubled as a team: my Linux user group.

When Adam walked in, he faced the same fears, uncertainties, and curiosities that I shared at the start of the book. However, he was as equally inspired and intrigued by the concept of community. Although he felt weird, he also felt compelled to stick with it.

When Adam first joined the group, he asked a barrage of questions, regularly apologizing when he felt they were “stupid,” despite reassurances that there were no stupid questions.

As he built his confidence, he started experimenting. The group had become an excellent place for people to give away old equipment. Adam would take this endless stream of clapped-out gadgets, try new software, and share his experiences with the rest of the group. As new members joined and asked questions, he would answer them and share his experiences. He was fast becoming something of a Linux expert.

He continued to play, continued to grow, studied at university, and, before long, landed a job as a system administrator. Shortly after he met his girlfriend, Jenny, and before long they moved into their own house. Adam had achieved what he wanted: normality, a sense of security, and a passion.

What I find so endearing about Adam’s story is that a Linux user group (a team within the general Linux community) was so instrumental in furthering his ambitions. If Adam had not joined a team, whether my LUG or any other group, I am not sure that the story would have had the same ending.

Teams fill an important role in a strong community: they are small ecosystems with attributes that can be hugely valuable to success on the wider scale of your community. Once we’ve taken a high-level look at these attributes, we can use this knowledge to build our practical implementation strategy later in this chapter.

Units of Belonging

Teams are units of belonging. Members join, are energized by the team’s spirit, and develop a sense of belonging that encourages them to contribute back to the team. This “Circle of Life” philosophy provides the team with a consistent exchange of experiences and value.

If we slice a team open, we can see a number of generations, like rings in a tree trunk. Each generation is a source of stories (already established in the previous chapter as an important

source of communication) and also a source of mentorship. Each generation passes down stories, experiences, and life lessons to the new generation.

This is exactly what happened with Adam. When he joined the group, he needed assistance and help. When he received this help, the team provided a sense of value to him. This value inspired Adam to treat the team with respect and to prioritize its importance in his life. As the team opened doors to knowledge, social connections opened and he felt a sense of belonging. This belonging was sealed each time he imparted his own experience to new and existing members of the group. He would duly receive the appreciation and thanks that he delivered to others himself when he was new.

Part of the reason why Adam had such a positive experience is that he understood the full breadth of the team. In any natural human grouping, *scope* and familiarity of environment plays an important role.

Scope defines the breadth and range of something. It is the length of a book, the size of a festival venue, and the comprehensiveness of a project. Scope describes the extent of the mental fishbowl that you are in, be it a book, venue, or initiative. A typical example of scope that is familiar to all of us is the place in which we live.

The world is a pretty big place. Its magnitude is reinforced by the television, websites, books, stories, and photographs that we see every day. For thousands of years, humans and animals have developed a natural inclination to divide the world into more manageable chunks. Society takes the same approach: countries are divided into states and counties; states and counties are divided into cities, towns, and villages; and cities, towns, and villages are divided into streets and, ultimately, houses.

Geographical communities are analogous to digital communities: with both, our perception affects our confidence. Take the example of someone who relocates to an entirely new area. For most people who have just moved, the environment is initially hugely overwhelming. Once this person has explored most parts of her city, town, or village and knows where the grocery store, hospitals, and other key resources are, she starts to develop a sense of confidence. Understanding the scope of the area gives us this confidence.

The reason for this is that we are all raised to be cautious of the unknown. Unfortunately, this makes us afraid of it, too. When we know the full extent of the fishbowl that we live in, as well as the other fish in that bowl, it helps us to build a measured sense of safety. Your sense of scope makes you feel safe when you confirm that you live on a safe street and not in the middle of a gangland.

The scope of the natives also affects our perception. Contrast someone with few friends and seemingly alien neighbors with someone who has lots of friends and very social neighbors. When you start to make recurring personal connections with regular friends, acquaintances, and love interests, a sense of belonging begins to flow through your veins. This is no different in a small face-to-face or online community.

Teams are hugely valuable in providing a manageable scope for your new contributors. If your community has 1,000 people in it in a single team, your community can feel overwhelming. If your community is instead broken down into 10 teams, each with 100 people in it, and each team focuses on a distinctive area, this is far more approachable.

Let's add this important premise of team building to our TODO list.

Community TODO List

- Identify how we can divide our community into teams.
 - Ensure that teams can communicate clearly and effectively.
 - Attract a diverse range of contributors to our community to get involved and contribute to our goals.
 - Build an environment conducive to our wider goals.
 - Define the scope of each team, and help team members understand that scope.
-

Read Versus Write

Communities come in many forms. They surround books, movies, software products, political campaigns, civil rights efforts, hobbies, and more. In all their colorful and varied forms, all communities share one distinctive trait: the unity of people around a shared belief or interest. It is passion that binds together these people.

Read-mostly communities

A great example of this passion is *Star Trek* fans. While I am by no means a member of that community (and only have a basic knowledge of the show), it is clearly far-reaching and maintains an extensive membership. *Trekkies*, as they are known, gather around the world to discuss, debate, and consume *Star Trek* in all of its wild and rather weird glory. Oh, and they wear quite ridiculous and at times bizarre outfits, too.

We can define Trekkies as *fans*—they enjoy a common interest together, either online or at conventions around the world, and their primary role is to share the consumption of that interest with others. As fans, Trekkies rarely have any direct impact on the unifying interest: the average Trekkie could not change the storyline of the show, contribute to the costumes or the digital effects, or improve what already exists. Instead, collaboration in the Trekkie world produces content for other Trekkies: it won't take you long to find wonderfully detailed fan-made costumes, props, tattoos, fan fiction, and other creations.

Although there appears to be a very distinctive collaborative divide between the provider (the show) and the consumer (the fan), there is a middle ground. These communities are becoming increasingly important to providers and never before have artists, musicians, producers, and politicians been so aware of their followers. The Internet has enabled fans to connect more easily with their heroes, and this has developed a kind of collaboration.

There are many examples of this. Stephen Fry, a well-respected British actor, started using the Twitter microblogging service and within months he had 400,000 people following his updates. The Obama administration has made use of blogging and YouTube extensively to distribute content and invite feedback.

One producer who has engaged repeatedly with his fan community is Joss Whedon, the Academy Award-nominated and Hugo Award-winning American writer, director, and executive producer of *Buffy the Vampire Slayer*. Whedon has been known to use a range of online media to interact with fans, and has even referenced fan contributions in his work.

One such example was a reference in an episode to a “Polgara demon.” The name “Polgara” originally comes from books written by American author David Eddings. Whedon’s decision to use the name was inspired by a fan who posted regularly to the official *Buffy the Vampire Slayer* forum with the same name. Whedon and producer David Fury used the name as a tribute to her regular contributions in the community.

Another example of Whedon’s commitment to his community bubbled to the surface during a writers’ strike. He worked with other writers to write and produce *Dr. Horrible’s Sing-Along Blog*, a 45-minute short musical that was released on the Internet. Each episode of the musical went online for only a few days, and Whedon generated interest inside the community using Twitter to interact with the fans.

These are all examples of community building in action. Producers are uniting groups of interested people together, providing communications facilities, and encouraging and enabling those people to collaborate. Even if the extent of that collaboration is a group of fans providing feedback on a forum, collaboration still happens.

This is how most communities work. Collaboration is an unofficial by-product, and although it may not change or improve on the creation of the producer, it is still likely to offer real value. The primary focus in these communities is to ensure that the community (a) always has available access to the product, and (b) that they can communicate about it with others. The foundation of these kinds of communities is *access*.

While each community has the characteristic of people gathering around a shared belief or interest, the actual impact of the community on this shared belief or interest varies greatly.

Write-centered communities

For some communities, collaboration goes much further. It becomes much deeper, more intrinsic, and more accessible to all. Instead of merely *enjoying things together*, collaboration

goes so far as to help people *create things together*. In these environments, the community also assumes the role of producer of the content.

The typical example here is one of the many Free Culture communities, such as Linux, Wikipedia, OpenStreetMap, Creative Commons, etc. In these communities, community members have the opportunity to change the very content that brings them together.

The Ubuntu community is one such example. Ubuntu is an entirely free Linux operating system that is designed to provide a complete, free, stable, and secure system for desktops, servers, or mobile devices. Ubuntu is built using hundreds of pieces of preexisting Free Software tools that we refer to as *upstream* applications. Each of these upstream applications has its own community of volunteers and developers. Ubuntu itself has a community of volunteers and developers who take said upstream software and assemble it in an easy-to-install and easy-to-use system.

Ubuntu has spawned a huge community. Over 200 Ubuntu user groups, known as Ubuntu LoCo Teams, have formed around the world. Nearly every country has a LoCo team, with a range of users from Linux experts to entirely new users. These people are, in a word, *awesome*. (Maybe I am a little biased....)

Anyone in the Ubuntu community can affect what appears in Ubuntu itself. Every part of the system is open to contribution. This includes the software on the disk, the supporting documentation and resources, art and design, bug reports, and almost anything else. Everyone has the opportunity, through hard work and merit, to make a positive contribution to the Ubuntu system that the community gathers around.

Understanding how our communities collaborate is important to understanding what we want to achieve and what is possible. Let's add this to our TODO list.

Community TODO List

- Identify how we can divide our community into teams.
 - Ensure that teams can communicate clearly and effectively.
 - Attract a diverse range of contributors to our community to get involved and contribute to our goals.
 - Build an environment conducive to our wider goals.
 - Define the scope of each team, and help team members understand that scope.
 - Understand the extent and range of collaboration between our teams.
-

Meritocracy

Before we continue to build the blueprint for our community, I want to take a few minutes to digress and talk about an interesting social attribute that applies to many (but not all) communities: *meritocracy*.

A meritocracy is a system of governance in which its members are given responsibilities and recognition based upon achievements, merit, and talent. Those who are part of a meritocracy (such as in the Ubuntu and other open source communities) can make tremendous advances in respect and responsibility by simply doing good work. In these communities, money, class, and family connections have little or no impact on the ability to progress and build a reputation.

The magic of meritocracies is that the playing field is level for *everyone*. Those who work hard and show a recurring commitment to the community are rewarded. Those who think that driving a car with a blue neon light underneath it will impress us are going to be sadly disappointed.

Few would argue that a meritocracy is a bad thing. Its fundamental basis is in rewarding hard work. This concept largely maps to the general life lessons that we are all raised with: work hard and you reap the rewards of your efforts.

In these collaborative meritocracies, our primary goal is to ensure that the communication and contribution channels to collaboration are open, well defined, and enforced. These communities are complex: there are many different aspects that affect how simple it is to become involved and collaborate.

Although meritocracies represent the poster child for what many consider great communities, they are *not* a requirement. Some communities absolutely distinguish between members based upon who they are, where they come from, and other attributes. This has been particularly applicable for business-oriented communities that maintain a clear hierarchy and members who are by no means considered equal.

Your community needs to decide for itself whether it is a meritocracy. I would, however, tender one piece of advice: if you are involved in a volunteer community that is open to all, I would highly recommend you take a meritocratic approach. It will make your community look and feel more accessible and help to encourage a sense of belonging and equality as opposed to a community divided by classes. From the perspective of new members, the opportunities offered by meritocracies are inspiring. It is hugely attractive to members when anyone can join a community and further themselves and their reputation based upon great work and participation.

If your community is currently or has decided to be a meritocracy, you should communicate this extensively to the outside world. Don't use the word "meritocracy," though: most people have no idea what it means. Instead, talk of equality and provide examples of how your members have built their reputations based on their efforts.

We talk about many of these areas throughout the book, specifically when we talk about *processes* in [Chapter 4](#) and *infrastructure* in [Chapter 5](#). For now, though, we need to focus on the raw material that forms teams—*people*.

Working Together Is Success

Henry Ford was a pretty smart guy. In 1891 he went to work for Edison Illuminating Company, where he started experimenting with the concept of a gasoline engine. After refining his design into what he called the Ford Quadricycle, he resigned in 1899 and founded the Detroit Automobile Company, which would later become the Ford Motor Company.

Although Ford was a brilliant engineer, this is not why I am talking about him. Instead, it's because of this quote that slipped out of him:

Coming together is a beginning, keeping together is progress, working together is success.

Although many consider Henry Ford the founding father of the motor car, this quote points to his other incredible achievement: understanding and motivating people.

Ford was very firmly a businessman who had his beady eye on the dollar, but in addition to this he was a pioneer of *welfare capitalism*, intended to reduce staff turnover and improve efficiency. To achieve this, he used many methods: he paid staff a higher salary, improved working conditions, and automated large parts of the process. We can thank Henry Ford for inspiring the modern assembly lines that we see in mass production factories around the world. (Those of you who work on such lines may not be quite so enamored, mind.)

Ford built a business based not only on a core product, but on an understanding of people. He knew how to divide his workforce into teams and have them work together to produce a single, consistent product. He knew that people driven by a common goal and united in similar skill sets would be productive. And he was right.

One of Ford's most significant and, at the time, controversial initiatives was the introduction of a 40-hour week and a minimum wage. His seemingly generous increase took a daily wage from \$2.34 to \$5. Back in 1914, this was unheard of, and rocked other industrialists and Wall Street. Ford's reasoning was simple and elegant, though: he wanted his employees to be able to afford the very cars that they were building.

Ford knew that if his employees could afford to buy Ford cars, they would value their contributions to the company more. His employees would be able to see, feel, and enjoy the fruits of the labor.

Your teams also need to enjoy the fruits of their labor. Volunteer communities can sometimes feel like production lines. The work is not always enjoyable and not always pleasant. There are times in every community when repetition, housekeeping, and conflict play a role in an otherwise enjoyable merry-go-round. When the community begins to see more bureaucracy and repetition than useful and enjoyable contributions, something is wrong. Very wrong. It is

important, in these challenging times, to remind your community members of their crusade and the value of their work in achieving it.

Ford's generous wages did not come without a catch. The increased salary that his workers enjoyed was only available to employees who had been at the company for six months or more, and also demanded that employees lived their lives in a way that the company's "Social Department" approved of: no excessive drinking or gambling. Erk. Unsurprisingly, that scheme was soon abolished, and the company had to accept its staff for who they were. *Diversity* was becoming a hot topic, and continues to be in any community.

Diversity

The building blocks of community are its teams, and the material that makes these blocks are people. When we understand people, we can build humane environments that are energizing and inspiring. Central elements of a healthy environment are respect, diversity, and rewarding people for their efforts, irrespective of who they are or where they come from.

Typically, when we talk about diversity, we use familiar examples: gender, race, sexuality, and class. Although important, these poster children of diversity can sometimes focus attention away from more subtle and potentially potent forms of diversity that we can encourage, explore, and celebrate. Diversity descends much deeper than skin color and gender.

George B. Graen, author of *Dealing with Diversity* (Information Age Publishing), argues that not all differences are equally relevant or as important as you would think in all circumstances. He broadly divides diversity into *surface-level* diversity—readily observable characteristics such as race, gender, or age—and *deep-level diversity*, important but less readily transparent traits such as personality, values, and attitudes.

Building deep-level diversity can bring a wealth of goodwill and openness to your community. Often these deeper, hidden kinds of diversity teach us life's most valuable lessons. While all equality is important, we need to grow this sense of deep-level diversity.

At the start of the chapter we identified that we need to build an environment that is conducive to energizing and enabling our community. Deep-level diversity is the open door that welcomes this enablement. When we encourage this deep-level diversity of contributions (e.g., translations, documentation, development) as well as diversity of opinions, values, and experience, your members will feel unrestrained, unrestricted, and energized. This should be a constant consideration throughout your work.

Unfortunately, many communities merely focus on hammering home equality of surface-level diversity, but Graen suggests in *Dealing with Diversity* that it doesn't make a big difference in the effectiveness of the community:

In a study of 45 teams from electronics divisions of three major corporations, Pelled, Eisenhardt, and Xin (1999) found that the effects of surface-level diversity (age) on emotional conflict diminished as a function of team longevity. Similarly, Chatman and Flynn (2001) found that

demographic homogeneity (race and gender) was less predictive of team cooperation as team members interacted with each other.

This insight is interesting when combined with another research study that found that deep-level diversity provides the kinds of benefits we are looking for:

In a study of 144 student project teams, Harrison, Price, Gavin, and Florey (2002) found that surface-level diversity negatively affected early cohesion in the team. Over the course of a semester working together, surface-level diversity became less predictive, whereas actual deep-level diversity (measured by conscientiousness, task meaningfulness, and outcome importance) and perceptions of deep-level diversity became increasingly important to team social cohesion and performance.

Although this experiment may seem a little abstract, Graen summarizes that “as team members interact, attributions about underlying differences based on race, gender, and age are likely to be minimized; however, the underlying differences in terms of personality, values, and attitudes are likely to have an increasingly negative effect on team cohesion and performance.” We need to be cognizant of these underlying deep-level attributes in people and ensure we encourage and help them thrive in our communities.

Deep-level diversity is further underlined by the sheer extent of diversity in most communities, particularly those online. Diversity is everywhere. We have so many opinions (sometimes it can feel like too many at times), viewpoints, perspectives, recommendations, and other reactions to stimulus. At every step, we need to foster and encourage open and frank debate. Every communication channel that you construct needs to have a common theme of openness and respect that encourages this kind of diversity.

Although this is an intellectually responsible position, people are people, and people can be irresponsible. For diversity to thrive and prosper, it needs to be built on a foundation of *respect*. When members of a community are respectful and considerate of each other, it encourages an environment in which people feel comfortable bringing their own kind of diversity to the game. It is this respect that affirms Graen’s previous acknowledgment of the risks of diversity on team cohesion and performance.

In the Ubuntu community, there is an important document called the “Ubuntu Code of Conduct” that builds this foundation of respect in contributors. I have reproduced the core of the document here, as it not only outlines these core attributes of respect, but also could be useful in a range of communities:

Be considerate

Your work will be used by other people, and you in turn will depend on the work of others. Any decision you make will affect users and colleagues, and we expect you to take those consequences into account when making decisions. For example, when we are in a feature freeze, please don’t upload dramatically new versions of critical system software, as other people will be testing the frozen system and will not be expecting big changes.

Be respectful

The Ubuntu community and its members treat one another with respect. Everyone can make a valuable contribution to Ubuntu. We may not always agree, but disagreement is no excuse for poor behavior and poor manners. We might all experience some frustration now and then, but we cannot allow that frustration to turn into a personal attack. It's important to remember that a community where people feel uncomfortable or threatened is not a productive one. We expect members of the Ubuntu community to be respectful when dealing with other contributors as well as with people outside the Ubuntu project, and with users of Ubuntu.

Be collaborative

Ubuntu and Free Software are about collaboration and working together. Collaboration reduces redundancy of work done in the Free Software world, and improves the quality of the software produced. You should aim to collaborate with other Ubuntu maintainers, as well as with the upstream community that is interested in the work you do. Your work should be done transparently and patches from Ubuntu should be given back to the community when they are made, not just when the distribution releases. If you wish to work on new code for existing upstream projects, at least keep those projects informed of your ideas and progress. It may not be possible to get consensus from upstream or even from your colleagues about the correct implementation of an idea, so don't feel obliged to have that agreement before you begin, but at least keep the outside world informed of your work, and publish your work in a way that allows outsiders to test, discuss, and contribute to your efforts.

When you disagree, consult others

Disagreements, both political and technical, happen all the time, and the Ubuntu community is no exception. The important goal is not to avoid disagreements or differing views but to resolve them constructively. You should turn to the community and to the community process to seek advice and to resolve disagreements. We have the Technical Board and the Community Council, both of which will help to decide the right course for Ubuntu. There are also several Project Teams and Team Leaders, who may be able to help you figure out which direction will be most acceptable. If you really want to go a different way, then we encourage you to make a derivative distribution or alternative set of packages available using the Ubuntu Package Management framework, so that the community can try out your changes and ideas for itself and contribute to the discussion.

When you are unsure, ask for help

Nobody knows everything, and nobody is expected to be perfect in the Ubuntu community (except of course the SABDFL [Self-Appointed Benevolent Dictator for Life]). Asking questions avoids many problems down the road, and so questions are encouraged. Those who are asked should be responsive and helpful. However, when asking a question, care must be taken to do so in an appropriate forum. Off-topic questions, such as requests for help on a development mailing list, detract from productive discussion.

Step down considerately

Developers on every project come and go, and Ubuntu is no different. When you leave or disengage from the project, in whole or in part, we ask that you do so in a way that minimizes disruption to the project. This means you should tell people you are leaving and take the proper steps to ensure that others can pick up where you leave off.

NOTE

The “Ubuntu Code of Conduct” is available at <http://www.ubuntu.com/community/conduct> and licensed under a Creative Commons Attribution Share Alike license, which allows you to use it in your own communities. Although a Code of Conduct is not required for a thriving community, it is highly recommended.

Although the “Ubuntu Code of Conduct” draws attention to understanding and respecting a deep level of diversity, it is sometimes misinterpreted as simply “don’t be an idiot.” It means far more than that: it encourages us to not only take responsibility for our actions and our reactions, but also to use this diversity as an opportunity to learn and grow, turning differences into opportunities for personal development and learning.

Therefore, in addition to adding diversity as a goal to the TODO list, let us also note that we should put in place a Code of Conduct.

Community TODO List

- Identify how we can divide our community into teams.
 - Ensure that teams can communicate clearly and effectively.
 - Attract a diverse range of contributors to our community to get involved and contribute to our goals.
 - Build an environment conducive to our wider goals.
 - Define the scope of each team, and help team members understand that scope.
 - Understand the extent and range of collaboration between our teams.
 - Encourage diversity and opportunity in the community.
 - Produce a Code of Conduct.
-

Every community needs to cherish and respect deep-level diversity. Its importance is not something that can be enforced with actions, bullet points, success criteria, or other organizational devices. Leaders are responsible for modeling the right behavior and encouraging it in others, but ultimately all members are responsible for remembering why your community is doing what it is doing and standing shoulder to shoulder, connected by diversity to grow and take on future challenges. It is diversity and this sense of united openness that will keep your community strong and reactive to any obstacles in its way.

In the previous few pages we have explored some of the core aspects of building community. We have learned that we need to take a cast of dots, bring them together into teams, help them communicate, and create an environment that is conducive to building strong community.

These essential elements of community are the primary goals that we want to achieve in our social economy. There are no specific actions or tasks that can achieve each of these elements; instead, they need to live in the woodwork of your community. They are attributes that we should always strive for, both throughout the rest of this book and throughout our future endeavors.

Now it is time for us to change gears and delve into the specifics of exactly what you want to achieve in your community.

Designing Your Community

At the start of this chapter, I waxed lyrical about strategy. Unfortunately, many community leaders consider strategy as an afterthought: they think of it as a nod toward the bureaucrats, not as something that actually helps their community grow. This view is misplaced because leaders *can* maintain the flexibility they need while producing plans that can help structure and enable the community. Indulge me....

When your community kicks off, you'll be way ahead if you can get down on paper its primary purpose and core goals. To do this you need 1 tablespoon of aims, a mission statement, 1 cup of objectives and goals, bake for 45 minutes, and allow to cool. The result: a *strategic plan*.

NOTE

The strategic plan that we develop throughout this chapter is a hugely important tool if you are a company who is likely to hire a community manager. We discuss this in more detail in [Chapter 11](#).

To get started, you first need to answer some simple yet broad questions. Write down a single detailed sentence or a collection of single words that answers the following:

What is the mission?

We want to understand our primary mission—the bright, shiny prize for which we encourage and inspire our community. What is this eventual outcome that we lust for? Is it a software release, political change, to help a demographic of people, or to produce something?

What are the opportunities and areas of collaboration?

We want to explore how our community can work together to create and achieve things. What are these areas? How can we work together in different ways?

What are the skills required?

We want to identify what skills we need in our community so we can later establish teams to house these skills. What are these skills?

These simple questions are the foundation of a more detailed set of objectives that we will flesh out over the following pages. Let's look at a few example answers.

Earlier in this book, we talked about the Jokosher audio production software project that I kicked off from my mocked-up designs. The project is focused on producing an easy-to-use application to record and mix audio. Let's apply these questions to Jokosher:

MISSION: To produce an integrated audio production environment for the purposes of recording, mixing, and exporting audio built on the principles of ease of use and simplicity, utilizing open source technology.

OPPORTUNITIES: Easing audio production, rethinking assumed knowledge, and asking questions in an integrated environment, and open and free access to simple audio production technology.

AREAS OF COLLABORATION: Interface design, development, documentation, translations, testing.

SKILLS REQUIRED: Programming (audio, interface), documentation, web design, web content, testing, bug triage, translations.

Note how these answers have been structured. We have used a single, very detailed, and carefully considered sentence for the "MISSION" and "OPPORTUNITIES," and single words or short phrases for the other areas.

Now spend some time thinking carefully about how you distill your answer to the "MISSION." Let's delve into it in a little more detail:

To produce an integrated audio production environment for the purposes of recording, mixing, and exporting audio built on the principles of ease of use and simplicity, utilizing open source technology.

This single sentence communicates all the key aims of the project:

- The kind of tool we wish to produce: audio production environment.
- The primary functions of the tool: recording, mixing, and exporting audio.
- The principles of the project: ease of use and simplicity.
- The scope of its approach: using open source technology.

Distilling the broad goals of the project into a single sentence helps you to understand what you really want to achieve. It also gives you a great summary of the project that you can share within minutes: this is called the *elevator pitch* and is useful for attracting new contributors and spreading the word about your community. More on this later.

The “OPPORTUNITIES” answer is the area in which you should identify all the exciting opportunities that are possible if you achieve your goals with the community. Producing revolutionary software? Changing the quality of life for homeless people in your neighborhood? Furthering a particular skill? Helping kids to eat healthy foods? Whatever your dreams, these should be the most important and inspiring opportunities that you are seeking. Remember, we are looking for high-level, essential goals here.

The “AREAS OF COLLABORATION” answer is where you can note the areas in which the community can *work together* on a task. This could include writing documentation, organizing events, advocacy, writing software, etc. Some tasks in the project cannot or will not be collaborative (for example, administering the infrastructure facilities for the community); you obviously should not include these.

The “SKILLS REQUIRED” part is where you should note the skills that are required to make your mission a reality. What types of skills are needed to achieve those goals? Scribble down the full range of skills that you will need. This explicit statement will be handy later when attracting contributors.

Our Jokosher example earlier was a very technical project focused on open source software. Let’s explore another: a neighborhood watch community.

MISSION: To build a sense of safety within the FooVille area of BarTown by creating a sense of community, opening up communication, and building oversight into daily lives.

OPPORTUNITIES: A safer neighborhood, fewer worries for kids playing outside, more desirable living conditions, more attractive house-buying area, better social interaction between residents, improved relations with the police.

AREAS OF COLLABORATION: Organizing events, publicity, online discussion, signage.

SKILLS REQUIRED: Event organization, design, web design, advocacy, writing.

Here we focus on building a sense of safety in a specific region of a specific town.

Go ahead write the “MISSION,” “OPPORTUNITIES,” “AREAS OF COLLABORATION,” and “SKILLS REQUIRED” for your community.

With a defined, core set of objectives and their characteristics noted, it is now time to flesh them out into a mission statement and form the basis of our strategy.

Baking in Openness

Always remember that, when building your community, transparency and openness are critical considerations. Few open volunteer communities succeed without transparent methods and approaches. As such, you should be open and inclusive with your community as you develop your strategy.

NOTE

Throughout this book we talk about transparency and the best methods of growing it in your community. It is an important consideration in everything you do, and it should be a top-level priority in how you work.

Your role in building this strategy is in facilitating discussion and helping to bring conclusions from those discussions into a single strategic document. You are not here to make decisions on your community's behalf. You also are not here to dictate direction and decisions. You are here to gather feedback, opinions, and ideas publicly and develop a strategy that meets as many needs and expectations in your community as possible.

There are many approaches to building a strategic document in an open and inclusive way, but here is my recommended approach:

- Have a central place in which the strategic documentation is developed. I recommend a website or wiki. Ensure that everyone in your community has access to this location.
- As you work through the concepts in this chapter, engage in public discussions with your community to make decisions. This could happen in face-to-face meetings or in online mediums, such as mailing lists and forums. Gather this feedback and use it as a basis for additions to the plan.
- Regularly update the documentation and solicit feedback from the wider community on your changes.

This approach ensures that you develop your strategic plan in an open and transparent manner, and that the plan reflects the perspectives and desires of your community. The key is regularity of communication and feedback, and a central document that brings it all together. If you approach your strategy in this manner, your community will feel open and accessible.

One difficult aspect of this process is when people disagree on direction. You may get some people who believe that Approach A makes sense and some who feel that Approach B is more appropriate. Unfortunately, there is no easy solution to this problem.

An example of this problem occurred in a derivative distribution of Ubuntu called Xubuntu. The Xubuntu community took the underlying Ubuntu system and swapped out certain components to make it more lightweight and suitable for lower-powered computers. Some time ago, the community unfortunately hit something of a roadblock when deciding in which direction to move forward. They not only disagreed on the direction of the project, but also disagreed on who should coordinate the strategy.

To resolve this, I stepped in and scheduled a public online meeting and invited the entire Xubuntu community to attend. My first goal was to focus on what the community agreed on. Much of the meeting was gathering everyone's feedback and collating the recurring themes together into a broad, high-level direction. After 45 minutes or so of discussion, this was finalized as:

To produce an easy to use distribution, based on Ubuntu, using Xfce as the graphical desktop, with a focus on integration, usability and performance, with a particular focus on low memory footprint.

The integration in Xubuntu is at a configuration level, a toolkit level, and matching the underlying technology beneath the desktop in Ubuntu. Xubuntu will be built and developed as part of the wider Ubuntu community, based around the ideals and values of Ubuntu.

I managed to converge on this statement by repeatedly refining it based on feedback. This happened over and over again until we got something that virtually everyone agreed on. This agreement is composed of two parts. The first provides a broad mission statement, and the second specifies a broad set of high-level goals that can implement that statement. This dynamic duo could then be used to form the basis for an additional set of discussions to flesh out a full strategic plan.

For this process to happen effectively, it needed a coordinator, and my next job was to get agreement on who this coordinator should be. The community was united in recommending a competent and regular contributor in the form of one Cody Somerville. Cody worked with the community using the approach I recommended and produced a full strategic plan. The Xubuntu community went on to do some great work built around this strategy.

Building a Mission Statement

In the same way the Xubuntu community agreed on their high-level goals, a mission statement holds a lot of value for any kind of collaborative project, be it a commercial product or a volunteer community. These statements emphasize the promise, opportunity, and definition of what your community is seeking to achieve. Their purpose is to articulate ambition with a detailed, succinct, and elegant approach. Mission statements help get everyone in your community on the same page.

Mission statements are intended to be consistent and should rarely change, even if the tasks that achieve that mission change regularly. When building your mission statement, always have its longevity in mind. Remember, your mission statement is your slam-dunking, audacious goal. For many communities these missions can take decades or even longer to achieve. Their purpose is to not only describe the finish line, but to help the community stay on track.

Many of us are more than familiar with mission statements. They are one of the first documents we see when we join a new company. Unfortunately, staff members typically ignore them. These statements are usually written by senior management and often bear no day-to-day resemblance to the work done by the folks on the ground. Don't use these often-pointless examples as a reason to consider mission statements irrelevant: these companies are simply doing it wrong. The mission needs to be at the forefront of every member's mind, and should be a regular driving force that is the justification behind the day-to-day work. Your community

members need to be able to draw a connection between the daily work of the community and your mission.

Now sit down and write your mission statement. Take the “MISSION” sentence that you wrote down earlier, break it apart, and illustrate it using descriptive, evocative, and stimulating words. Your mission should define the purpose of your aims, the bigger picture of where they fit in, and the uniqueness of your approach. You should expand on your “MISSION” while also using your “OPPORTUNITIES,” “SKILLS,” and “AREAS OF COLLABORATION” for inspiration. Although there are no fixed guidelines for the size of a mission statement, keep it detailed yet concise. If you exceed 300 words, you may be babbling a little too much. This, my friends, is a babble-free zone.

When you have completed the statement, you should run it through three rigorous tests:

Test 1

Put yourself in the position of a potential community member. If you had no knowledge of the community or its goals, would the mission statement explain it all within a minute? With every sentence, you risk the reader getting bored and wandering off for a love affair with his PlayStation. If your mission doesn't deliver your aims quickly, efficiently, and compactly, go and improve it.

Test 2

Get someone else to read it. Ask her to tell you what she thinks and how she perceives the aims of your community. Typically this person may be a friend, but friends often skirt around criticism. Make sure your reader is encouraged to criticize where needed, and tell her you won't get in a huff if she says something sucks. If she says something is unclear, fix it.

Test 3

Is this mission statement going to inspire you and your members through the toughest times of the community? In *Organizational Vision, Values and Mission* (Crisp Learning), Cynthia D. Scott et al. describe this perfectly: “A mission evokes a personal response. Work on it until it gets to be so clear that reminding yourself of it will keep you, on a really bad day, from walking out and quitting.” Is your mission going to stop you from quitting when the world has climbed onto your shoulders? One day this is going to happen, and you need to be able to look at your mission statement and have “a moment.”

When you have successfully navigated through these three tests, you should have a rock-solid and rigorously tested mission. Our mission is our first guiding document for our community. We'll use it now for the next step: building our *strategic plan*.

Building a Strategic Plan

The purpose of a strategic plan is to document your goals and ambitions for a given period of time and to provide a central body of agreement in your organization or community. It should

clarify what your objectives are and which goals are part of those objectives; it should also state how progress is measured and who is leading the work. A detailed, realistic strategic plan is hugely valuable for your community. Having a strategy does not mean that you have bent to the gods of bureaucracy.

A strategic plan is broken into objectives that we seek to achieve before a given milestone. This milestone can be measured in either time or a specific achievement. Your choice is entirely dependent on the type of community and project you are involved in.

As an example, some software projects use fixed release cycles, such as Ubuntu with its six-month cycle. Other software projects set their milestones as a given release that will have an agreed amount of functionality. Nonsoftware communities often use other kinds of metrics, such as an amount of money to raise in donations or a date when a particular event or initiative is launched.

FIXED SOFTWARE CYCLES

Many of you reading the *The Art of Community* will be seeking to apply these principles to open source software development. With this in mind, here are a few more words on the merits of fixed release cycles versus the release-when-ready approach.

Fixed release cycles offer many benefits to your users and developers:

- They're predictable and reliable. Your users know when every release will happen.
- It's easier to break the cycle into parts. When the cycle is fixed in length each time, you can divide it into the requisite parts (development, translations, UI freeze, hard freeze, release candidate, etc.) more easily.
- Particularly for open source projects, fixed release cycles make it easier for distributions to factor your release into their road map. This happens with the GNOME and Ubuntu projects, for example.

There is, however, one distinctive disadvantage with a fixed cycle: it makes it harder to develop large new features and changes that take longer than the duration of the cycle to implement. This is something we discovered in the Ubuntu community. Planning longer-term features sometimes needs to be broken into chunks and spread across release cycles. Interestingly, breaking the problem into chunks is often a better approach, irrespective of release cycles.

Release cycles are fairly serious beasts. Fortunately, they lead to more fun and exciting prospects: our goals. When you hit these goals, you should let your hair down and celebrate, and when this occurs regularly, it reaffirms a strong sense of community and teamwork by acknowledging that you have all been working your socks off toward your goals. We discuss methods of celebrating releases in [Chapter 10](#).

You should think carefully about what kind of milestone you want to apply to your project. It is far better to choose a milestone that is six months away and achieve less than pick a milestone that is two years away and achieve more. Regular milestone achievements give your community a much-needed dose of excitement and satisfaction.

Choose your milestones now (e.g., using a fixed cycle, a given set of features, a given date, or some other indicator of completion).

Structuring the plan

With your milestones decided, you can focus on the structure of the strategic plan. We will familiarize ourselves with this structure and then begin discussing how we can identify the objectives and goals to fill it. Before we begin, you should remember to fulfill your role as a facilitator and work with your community in a transparent fashion to combine input, feedback, and opinion into a single consistent plan that everyone can follow.

There are many, many ways of structuring a strategic plan. Everyone has their own approach, and countless books have been written on the subject. The technique that I am going to use here is one that I have found particularly effective for the communities that I have been involved in. Unfortunately, many of the books written about strategic design and planning are written with businesses in mind, and some aspects of business strategic planning are unsuitable for community strategy.

This is because business strategic planning typically presumes an organizational structure that provides a more central decision-making function: it is always clear who has the authority and expectation to make decisions. Ironically, this perspective is becoming increasingly old-school in business circles, with many businesses focusing on a more community-oriented collaborative planning approach that helps foster deeper team commitment even when there's a central position of responsibility (not authority).

The challenge here is that there is no cut-and-dried path to community, and the same applies to your strategic planning. As you grow and build your community, you should feel free to experiment, explore, and refine the approach we use here. You should develop a strategic plan that works for you and your specific community.

Our approach here involves defining a number of *objectives*. These are the broad high-level things that you want to achieve. Each objective is divided into a number of *goals*. Each goal includes three pieces of information.

- The *success criteria* describe a set of measurable methods for evaluating the goal's success. You should be able to look at this statement and determine straightaway whether the goal was achieved (example: 20 new community members).
- The *implementation plan* describes what tasks need to happen to achieve the goal. These are a set of directions that clearly indicate the steps involved in achieving the goal.

- Finally, and this is optional with many communities, we specify the *owner*: the person responsible for the goal. Accountability is an important element in building successful community: when people feel responsible for their work, they ensure positive outcomes are generated.

Your plan should not live in isolation. Its final home really shouldn't be a piece of paper in your office drawer or a random document on your computer. It should be shared. It should be read. It should become a core document in your community.

To make the plan easy to read, I recommend that you structure each objective and its goals in a consistent manner. Let's look at an example of an objective and one of its goals (remember that most objectives will have many goals):

OBJECTIVE: Build a website for the project.

GOAL: Build a structural design for the website content.

SUCCESS CRITERIA:

All areas of the website structure documented in a specification.

Community feedback gathered on the proposal.

IMPLEMENTATION PLAN:

Identify the needs of the website by liaising with the community.

Document the structure of the website on the wiki.

Email the core community team with feedback and merge feedback in.

Organize an online meeting to propose any changes to the specification.

Build a prototype.

OWNER: Jono Bacon.

This example demonstrates a number of subtle points in building a comprehensive strategic plan:

- The OBJECTIVE should be a specific outcome that you would like to achieve. These should be high level (such as "Build a website for the project") but not too fluffy (such as "Make everyone feel warm and fuzzy").
- Each objective can have a number of GOALS, and each goal should have SUCCESS CRITERIA, IMPLEMENTATION PLAN, and (where appropriate) OWNER items.
- The SUCCESS CRITERIA should be a set of measurable assessment methods. You should be able to look at each SUCCESS CRITERIA point and definitively know whether it was achieved. Each item should clearly state what needs to be achieved to be considered complete. Avoid vague and general statements; instead, use specifics. Success criteria helps the wider community reach a consensus on what constitutes success and identify when progress is being made.

- The IMPLEMENTATION PLAN is a set of reasonably granular steps that indicate how you can achieve the goal.
- In some cases it makes sense to assign an OWNER to a goal, even if that owner merely has oversight in getting it done. OWNER does not necessarily map to responsibility for the goal in a community, although it typically does in the business world.

With this consistent structure across your plan, you will be able to effectively document the broad objectives and specific goals and measure and evaluate progress on your action points. Now let us use this structure to add some meat to the bones.

Filling Out the Plan

With a mission statement, strategic plan structure, and some notes about our aims, we have already made some great progress in getting organized. Many communities are built on vague ideas that are barely communicated and shared, and progress is scattershot. Their approach is often uncoordinated and without schedule. Our work so far already has our community off to a rocking start.

But the devil is in the details. We need to use our structure here to flesh out what we want to achieve in our mission statement. We need to take our mission; combine it with our notes about opportunities; and produce a set of objectives, goals, success criteria, and implementation plan items.

Unfortunately, in terms of your community I have no idea what you specifically want to achieve. Some of you will be working on software projects. Some of you will be setting up user groups. Some of you will be traveling to far-flung lands to help poor children. I cannot directly choose your specific objectives and goals, but I can advise you on *how* to choose them.

First, take your mission statement and pick it apart. Now use the high-level aims as a source of discussion for brainstorming sessions. We need to flesh out, discuss, and debate our ideas and their implications and requirements. These sessions will generate this set of ideas that you can merge into the strategic plan.

Brainstorming Ideas

The justification for collaborative brainstorming is twofold. First, no matter how intelligent you are or how worldly you consider yourself, collaborative brainstorming *always* uncovers new ideas, concepts, problems, and techniques that you never considered. This is hugely valuable. Second, collaborative brainstorming is an important step in building transparency and openness in your community. This sense of openness is critical at every step, including strategic development.

Getting people together to share ideas and opinions should be exhilarating. It should open everyone's mind up to flow in an environment that encourages the expression of ideas. Great

brainstorming sessions inspire their participants: members not only feel that they can contribute, but that they can control the implementation of the very ideas they propose. Unfortunately, many brainstorming sessions don't quite work that way.

Many sessions comprise one driving force (the leader of the session) moving the discussion forward, backed up by a core group of two or three primary contributors. The rest of the group often sits in silence, observing the melee around them. Many of the ideas from these sessions are often plain and obvious; they fail to uncover core opportunities and issues.

Fortunately, great brainstorming sessions have a core formula:

Define a purpose

Your session should have an aim, a goal, and a purpose. What do you want to achieve in the session? What outcomes would you like to generate? Would you like ideas, work assignment, or other elements? Make sure each of your participants are aware of the session's aim. Remember, an important source of focus for your sessions should be the objectives of your mission statement.

Organize and invite your participants

Make sure that the people you want in the session know about it and can attend. If the session is online, be conscious of time zone issues. Allocate enough time to run the session. Usually 1–2 hours is enough time without boring the pants off people.

Get your resources in place

If your session is face-to-face, make sure you have somewhere to note down ideas. A whiteboard or flip pad is a great resource for this. If the session is online, a wiki or note-taker is a great option. Another great option here is a collaborative text editor, such as the freely available Gobby (<http://gobby.0x539.de/trac/>).

Set some ground rules

Make it clear that everyone is encouraged to participate. Also make it clear that offensive discussion and nonconstructive criticism should be avoided. Ideas should be expressed in high-level detail, but not discussed in too specific detail due to time constraints.

Help people relax

For many people, a brainstorming session is a social nightmare. Try to make the atmosphere as loose and informal as possible.

With this recipe in place, we can explore some methods of generating the best ideas from the session.

Technique 1: Question assumptions

With every brainstorming session, there is a set of assumptions and perceived norms. Your first point of order should be to tear open those norms and question whether they are effective. If they are not, the road is open to alternatives.

If your session is about improving ease of use in your software, is your current design the best approach? If you are discussing how to lower crime in your area, consider your assumptions: is it as bad as you think? If you want to bring more people to your community, does your target audience read and reside in the areas you think they do? Questioning assumptions can often uncover some great talking points and areas of investigation.

While we are on the subject of questioning the norms, it is also healthy to regularly question your own perspective and approaches. As a community leader, people look to you for guidance. It is hugely valuable to regularly sit back and reconsider the approaches that you ordinarily take for granted.

It might sound a little unusual, but I personally conduct a performance review of myself twice a year. In it, I ask my community for feedback on my approach, language, work, responsiveness, and other aspects of my work. On many occasions this review has helped me uncover some great areas in which to improve how I work with my own communities. I recommend you do the same.

Technique 2: Think outside the box

While we are all wrapped up in self-reflective thought, you should also question what you could achieve if you let your imagination run loose. What are the subconscious limitations that you are placing on yourself when thinking of ideas? In other words, if all the barriers are removed, what is possible?

A great example of this is the Nintendo Wii games console. For years, console after console had shipped with a control pad that players used to control the on-screen characters. These pads went through many variations, but they all shared one common characteristic: the buttons controlled the action.

Of course, there were attempts at alternatives: light guns, dance mats, plastic guitars and drums. Most still had the assumed knowledge that the player controlled the action by pressing buttons. These alternative approaches were never a core part of the systems: they were novelty additions that often had limited appeal.

The Wii changed all of that. Shigeru Miyamoto, a renowned video game designer and cocreator of many games, including *Super Mario Brothers*, *Donkey Kong*, and *The Legend of Zelda*, sat down with other designers and questioned whether they should be limited to the existing norms of the game interface.

The result was one of the most significant developments in video gaming history: the Wii Remote, which allowed gamers to control the action by moving the unit itself. This enabled all manner of physical interactions, from 10-pin bowling to boxing to ski-jumping. The Wii changed how we think about video games, and is clearly having an effect on the design of other consumer products as well.

Thinking outside the box takes time and focus to master. The trick is in questioning everything around you. A great exercise for this is to spend an hour noticing the details in your world and questioning why they work the way they do. Try to find faults. Try to find justification for the way things are. Importantly, when you ask these questions, try to think of solutions. If you question why the heater control in your car is labeled the way it is, what would be a better label? If you question why you always need to pay your rent check manually, how could it be automated for both you and your landlord? These are great opportunities to enjoy being a cantankerous consumer but with the added benefit of expanding your own problem-solving skills. Meaningless buzzword fans may want to refer to this as *Cantankerized Consumer Problem Assessment* (CCPA).

Technique 3: Let's make it suck

A great technique for questioning everything and uncovering new ideas is one that I tried first at the Ubuntu Developer Summit (UDS) in Cambridge, Massachusetts, in early 2008. The idea is simple: reverse the aims of what you want to achieve.

As an example, imagine you wanted to design a cell phone. Traditionally, you would brainstorm the attributes of a great cell phone. Instead, turn everything on its head. What would make the worst possible cell phone? Maybe it ignores all calls? Or maybe it only accepts calls from telemarketing companies? Maybe the buttons are too small? How about really short battery life?

When you ask these kinds of questions in a brainstorming session, it almost always breaks the ice and gets people talking. Such ridiculous questions generate a lot of fun discussion, laughing, and ludicrous ideas. Make sure you write every one of these nuggets of madness down.

After your group has exhausted their initial pool of ideas, you should now invert each idea again. How do we make sure that our phone accepts all calls? How can it avoid calls from telemarketing companies? How can we make sure the buttons are the right size and not too small? How can we improve battery life?

Aside from the benefits of getting your group brainstorming, this approach is an excellent method in building defenses against the infuriation of normal life. It helps to identify frustrating attributes and protect against them, and this will in turn provide better results. I have used this technique for brainstorming websites, processes, products, and more, and it has always been fun, productive, and useful.

Pulling Together the Threads

At this point in our journey into the depths of strategy you have four very important assets. Each of these was developed from our structured approach to thinking throughout the chapter:

1. A set of TODO list items: essential concepts that are important throughout all aspects of your community building.
2. An initial set of MISSION, OPPORTUNITIES, AREAS OF COLLABORATION, and SKILLS REQUIRED notes that flesh out a set of aims for your community.
3. A more complete mission statement that helps determine high-level goals and ambitions and serves as a source of inspiration for brainstorming.
4. A set of ideas generated from your brainstorming sessions.

Our task is now to convert these assets into strategic plan structure that can be shared with the rest of your community.

First, produce a set of objectives from the high-level goals in your mission statement and the list of ideas from your brainstorming sessions. Create objectives that you feel you can achieve within a reasonable time frame. Again, the choice of time frame is dependent on your project, but I would recommend it not exceed 6–12 months.

Now put this list of objectives into the strategic plan and apply the consistent structure we discussed earlier. Here is a quick recap of this structure:

OBJECTIVE:

GOAL:

SUCCESS CRITERIA:

Item

Item

IMPLEMENTATION PLAN:

Item

Item

OWNER:

GOAL:

SUCCESS CRITERIA:

Item

Item

IMPLEMENTATION PLAN:

Item

Item

OWNER:

For each objective that you have decided on, divide it into goals and assign each the SUCCESS CRITERIA, IMPLEMENTATION PLAN, and (optionally) OWNER items. You should now go and flesh out each of these items for each of your goals.

Let's now take a look at our TODO list.

Community TODO List

- Identify how we can divide our community into teams.
 - Ensure that teams can communicate clearly and effectively.
 - Attract a diverse range of contributors to our community to get involved and contribute to our goals.
 - Build an environment conducive to our wider goals.
 - Define the scope of each team, and help team members understand that scope.
 - Understand the extent and range of collaboration between our teams.
 - Encourage diversity and opportunity in the community.
 - Produce a Code of Conduct.
-

Each of the items on this list is critical in building a strong, proactive, effective community. They are all important attributes that need attention when building a healthy environment for our contributors.

Some of these items are discussed later in this book. We will defer discussion of how to “build an environment conducive to our wider goals” and “encourage diversity and opportunity in the community” until our chapters on processes ([Chapter 4](#)) and building buzz ([Chapter 6](#)), respectively.

We have already added “produce a Code of Conduct” to our list of objectives. This leaves us with some outstanding TODO list items, all of which relate to dividing our community into teams.

Teams: Divide and Conquer

Earlier in this chapter, we rambled on about the importance of teams. We explored how meritocracy, diversity, and working together can help us build these units of belonging.

Since that discussion, we have identified a set of objectives, each with related goals. It is now time to divide our community into a set of healthy, motivated teams.

Although we are going to strategically develop a set of teams to form our community, you should encourage free-form team creation. Giving your community the ability to build its own

teams without prior approval can produce incredibly diverse and productive results. We have seen this in droves in the Ubuntu community, with many seemingly random teams doing great work in areas we never envisaged.

Although the door should be open to form new teams, a primary set of teams will perform much of the work in the community. Our four remaining Community TODO List items provide a set of guidelines for forming these core teams.

Identify how we can divide our community into teams

Earlier in this chapter we fleshed out some initial notes about our community and what general considerations we think about in terms of a mission, skills, and objectives. If you followed my advice earlier, you will have worked in an open and transparent manner with your community to flesh out this direction into a strategic plan. Your objectives and the work necessary to achieve them should be no surprise to your primary community members who were involved in constructing this strategy. To achieve our strategic goals, we need to form these teams that are so important to our community.

There are many approaches to setting up teams, and there is no standard recipe for how to do this. It varies among different types of community, and it varies among the experience and expectations of your existing members. When deciding on teams, consult again with your existing community. Have a discussion about which teams make sense. You can't build teams out of nothing, and as such, your first set of teams will almost certainly map to regularly contributing community members who want to engage in certain types of work. As an example, if you have contributors who want to work on documentation, they would be a great foundation for a documentation team.

When most communities set up teams, the approach is almost always based upon what similar communities have done. As an example, most open source projects will have development, documentation, and translations teams. This is largely because these teams are what we are familiar with in those communities. I highly recommend that you look into these existing communities and learn from their approach. This can offer valuable insight about the most suitable solution for you.

For most communities, it seems that teams typically map very well to primary skills. Earlier on when we built our initial notes about our community, we created our own list of these SKILLS REQUIRED. Many of these skills often map well to teams.

As an example, a software project will typically require these skills:

- Programming.
- Writing documentation.
- Testing.
- Bug triage.

- Advocacy.
- Website development.

Each of these skills could map to the following teams:

- Programming→Development Team.
- Writing documentation→Documentation Team.
- Testing→QA Team.
- Bug triage→QA Team.
- Advocacy→Marketing Team.
- Website development→Web Team.

Some skills (such as testing and bug triage) naturally map to the same teams. This is perfectly normal, and should be encouraged. Although you want to ensure a good range of teams cover the different types of contributions, you don't want to overload your community. Form new teams only when existing teams cannot sufficiently cater to new needs and requirements.

It is important that your teams have the ability to expand their mandate, adjusting their focus as they mature. As an example, a documentation team may expand to include general communications and education. As such, try to foresee this growth while keeping the team focused.

Define the scope of each team, and help team members understand that scope

Earlier we created a mission statement for the wider community, but mission statements can be useful for specific teams, too.

For each team you should produce a mission statement that outlines the specific work that the team performs. With this statement available for each team, you will better understand how your teams fit together, and so will prospective contributors: this statement will help you articulate the purpose of the team and help new contributors know which team to join.

As before, your definition should explain the typical activities that the team will do. As an example, the following could be used to describe a documentation team:

The documentation team identifies and prioritizes areas in which documentation is required, builds processes for producing high-quality documentation and best practice, and produces said documentation.

As a continued effort in transparency and openness, when writing team mission statements I recommend that you work together with other members of your community who are on those teams, in order to come to an agreement upon them.

Understand the extent and range of collaboration between our teams

Teams are not insulated from each other. Every team needs to work with other teams, and some will have more crossover than others. Identify how your teams will interact in different ways, and try to optimize for those teams that cross over the most.

Identifying these collaborative crossovers typically requires a significant amount of day-to-day experience in how the community functions. With this in mind, you should regularly observe how your community's teams collaborate.

For example, in the Ubuntu community, the bug squad (who triage and organize bugs) works very closely with the development team. By observing how the teams work, we were able to optimize the lines of communication, how both teams work together on the bug list, how meetings and events intersect, and other initiatives.

Ensure that teams can communicate clearly and effectively

You should always be cognizant of how your teams communicate together. How can the documentation team communicate effectively with the marketing team? How can the QA team work well with the development team?

The answer to these questions is complex and multifaceted. My recommended approach is to identify how you communicate tasks, issues, and goodwill:

Tasks

Enable teams to share and report what they are working on. When teams are able to identify who is working on what, this avoids duplication and confusion. Great approaches to this are regular meetings, notes, and articles (such as blog entries). This will be discussed in more detail later in this book, when we refine our communications facilities and processes.

Issues

Have regular, open opportunities for teams to communicate. Every team will have questions, concerns, queries, opportunities, and other topics to communicate to other teams. We need to ensure these lines of communication are open. Again, regular published meetings are the perfect opportunity to cover these kinds of issues.

Goodwill

Encourage and inspire strong and positive relations between teams. Always remember that your teams are part of the same mission. As such, they should celebrate victories together and console each other in hard times. If you don't focus enough on ensuring that teams feel part of the same machine working together in the right direction, the community will feel fragmented. This all boils down to bonding. Different teams should regularly bond together, either online, at physical meetings, and/or during other occasions. We will discuss this in more detail in [Chapter 10](#), when we discuss events. In the Ubuntu community, this occurs at every Ubuntu Developer Summit: different teams bond together, which helps the project as a whole.

As mentioned, we can achieve each of these methods of communication by producing a set of processes and events that every team makes use of. Reporting can help with tasks, regular meetings can help with issues, and social interaction and bonding can help with goodwill. We will talk later in this book about these different elements and how to implement them.

Setting expectations is critical in building community. When expectations are unrealistic or misaligned, the community forms its own theories. This can be a rocky road. The problem is that of Chinese Whispers (also known as *Telephone*, to some)—the game where a message passes from one person to another and changes at each step until the original meaning is garbled. When an expectation is surprising, often guesses and assumption are communicated as fact. This causes misinformation, concern, and typically negative outcomes.

The goal of this chapter is partly to clearly define these expectations. Your strategy is a means to get everyone in your community on the same page. If aspects of the strategy are unclear, there is the potential for mismatched expectations.

Again, this is an area in which experience plays a valuable role. After a short time working with your community, you will begin to understand better how they and you perceive the work that your community performs. This understanding will help you build a better strategy that will be more effective in achieving your objectives and goals.

Documenting Your Strategy

By now, a number of chunks of our strategy are in place. These include:

- A mission statement.
- A set of objectives and goals, each with success criteria, implementation plan, and owner details.
- A list of skills and how those skills map to teams.
- A list of teams, each with a definition of its scope.
- A set of TODO list items that we can utilize throughout our community-building activities.

Our final task in this chapter is to put the fruits of our labor somewhere that is accessible, easy to read, and easy to update. We need to communicate our strategy with the wider community and ensure that it can be referenced and updated when required.

As a bare minimum, your strategy should be available online, and at least available in the most common language that your community speaks. Multiple languages would be beneficial under the condition that every translation covers the full strategic documentation. Half-complete translations are not useful to anyone; the strategy must be complete.

The strategy that you publish should be made available in full and should not omit any sections. The purpose of this strategy is to unite the community behind one set of documentation and to form agreement around the objectives and goals outlined in that documentation.

There are a number of options available when it comes to publishing online, from a simple web page to a full content management system. The specifics of how you put your strategy online generally are not important; instead, open access to the content is the primary focus.

Many communities are using wikis as a method of publishing strategy online. A wiki is a website that anyone can update the content on, the most significant example being Wikipedia (<http://www.wikipedia.org>). The idea is simple: a community sets up a wiki to use as a general area for collaboration, and so it makes natural sense to put the strategic plan on there.

With anyone able to edit the content on a wiki, your community needs to decide how changes to this content are handled. You may welcome cosmetic changes such as spelling and grammar, but you should discourage random changes to the core agreement that is your strategic plan. Remember, the devil is in the details in strategic documents. To handle the tension between maintaining continuity and allowing fresh input, you have several options, some of which include:

- Put up the strategic plan as a static web page that changes only after a rigorous process of internal discussion and voting. New goals and other changes can be emailed to project leaders or brought up in meetings.
- Put up the mission statement and other rarely changing documents as a web page, but use a wiki or other malleable document for objectives and goals, so they can be rapidly updated by anyone involved.
- Put everything on a wiki, but make sure you subscribe to all changes through email, an RSS feed, or whatever other medium you like to receive news on.

Other options will certainly arise as new online media are developed.

Whichever approach you take, if the ability to change content is restricted, you need to be able to justify why those who have access do and why others don't. "I created the content" is not enough of a justification.

When some members of your community are given elevated responsibilities and privilege, it automatically puts a hierarchy in place in your community. The outcome is that those with this access are seen as leaders and governors. The production and development of these strategic documents is part of the responsibility of governing the community, and as such this governance may need to be formalized into a recognized body, such as a Community Council. We will discuss governance in detail later, in [Chapter 8](#) of this book.

As a final note, always remember that documentation and strategy do not solve every problem. Many people will be unaware that your strategy exists, and many who are will simply not read it. A strong and well-documented strategy does not replace the culture of communication and unity that we seek to grow throughout this book.

Wrapping Up

In this chapter, we have made some great progress in our journey. We have laid the foundations for an organized, well-planned strategy that will unite your community behind the same set of objectives and goals. With this foundation in place, we can expand and refine it as we progress through the book and explore the many other concepts further down the road. If you have followed through with each of the exercises in this chapter, it will have helped you to think structurally about your community and its goals, and share the fruits of this structured thinking with everyone involved. By definition this will help your community rock and roll that little bit harder.

When we started our journey into community, we talked about the social economy as a collection of processes to achieve social capital. These processes are essential in a simple, effective, and nonbureaucratic community. They affect how people join your community, how they collaborate, how ideas are communicated, how problems are communicated, and more. These processes can be the difference between success and failure.

The first set of processes that we need to consider is how we communicate effectively in our community. With this in mind, let's dive into the next chapter. All set? Let's go....

Communicating Clearly

“Before I speak, I have something important to say.”

—*Groucho Marx*

WHEN I WAS 11 YEARS OLD, my friend somehow obtained an LP he had “borrowed without asking” from his brother. He handed me the large disc adorned with cardboard sleeve, all carefully buried in a white plastic bag. As I retrieved said sleeve from said bag, my eyes widened and I skipped a breath: it was the first time I witnessed the sheer brilliance of an Iron Maiden record.

I listened to that LP until it damaged the pin in my record player. I loved the energy, I loved the look (including the spandex...I am not kidding), and I just wanted to be them. Unfortunately I had neither talent nor spandex but merely a bowl haircut and large white socks.

Inspired, I decided I was going to learn to play the guitar. My parents bought me an old acoustic guitar and I parked myself on my bed night after night trying to sound like my rock and roll heroes. Of course, I instead sounded like an incompetent 11-year-old with an acoustic guitar. I sucked, but I stuck at it.

As the years rumbled on, so did my guitar playing. Fortunately my skills were improving and I was reading more and more about music, guitarists, and the bands I loved. While flicking through a copy of *Guitarist* magazine, I came across a quote that I seem to remember was from Eric Clapton, but I’m not quite sure. Whoever it was, his words really resonated with me (pun intended):

It's not the notes you play, it's the notes you don't play.

While I am still making oodles of music today, this quote didn't really take hold of me until five years later when I started writing. What the little nugget of wisdom infers is that although we may obsess over the obvious components in an art (such as the notes in music), it is often the hidden and underlying messages that offer the most value (such as not playing certain notes).

This is a strong and relevant message for building communities. Community is absolutely about understanding the ether. Our notes are the processes, governance, tools, and methods in which we work together. The notes we don't play are the subtle nuances in how we pull these notes together and share them with each other. The space between the notes is *communication*.

He Said, She Said

Communication is essential in community. It is the metaphorical highway that connects the many towns and people in your world. Effective communication brings together your community members in a manner that is free-flowing, productive, and accessible.

You may spend hours putting together elegant and sleek processes, infrastructure, and governance, but if your community can't communicate well with each other, you may as well pack up your bags and go and pursue a career in cabaret on *The Love Boat*.

Our analogy with highways and towns maps eerily well to how we wire up our community with the communication channels it needs. To ensure our towns can work together, we have two primary tasks on our hands:

Create the highways

First, your community needs to build a set of resources to facilitate communication, discussion, and the sharing of ideas and best practices. In many cases these resources are online facilities, such as mailing lists, forums, and discussion channels.

Encourage great driving

Once your communication channels are in place, they can be used in all manner of ways. There will of course be some good drivers and some bad drivers; some will communicate exceptionally well, and some will irritate and agitate anyone who crosses their path. You want to inspire and encourage a baseline quality of communication. This is not about excluding people who are imperfect writers or speakers, but instead about providing a consistent example of simple approaches to communication that make the community easier to understand and more pleasurable for everyone involved.

In this chapter we are going to cast our sights on both of these topics, discuss a range of mediums for building the free flow of communication, make good use of those facilities, incorporate transparency, and avoid the common problem of a community preferring to speak rather than do.

So, let's get started by laying down the highways and putting some cars on the road. Now is the time to build our community's communication backbone....

Building Your Communication Channels

Earlier we created our TODO list of key aims and ambitions for building a strong community. We are now going to dig up that list and cast our sights on one very specific item.

Community TODO List

- Ensure that teams can communicate clearly and effectively.
-

Good communication serves many purposes in a community. It is the foundation of how your members work together, share goals and ambitions, and build social relationships between each other. It is communication that ensures everyone is on the same page, heading in the same direction, marching to the same tune.

Good communication is also a powerful security blanket. When communication breaks down in community, it can cause havoc. Volunteer communities are driven by members with a set of values that reflect themselves and the values of the wider community. These values are important to regularly reinforce in your community: they are the metabolism that fights off the threats and problems that can undermine the goals of the community. When your members feel like they are disconnected from the community, they lose their sense of value.

Communication can be divided into three primary areas:

Incoming

Receiving and processing feedback and viewpoints for the purpose of improvement. An example of this could include surveys to determine how well a part of your community is working.

Outgoing

Sharing news, stories, and achievements from the community with the rest of the world. An example of this could be showing off something your community has created.

Internal

Internal discussions and meetings in the community to discuss objectives, goals, conflict, and other issues. An example of this could include meetings that are designed to decide on how your community will work together toward its goals.

Each of these forms of communication is essential for a strong community. All communities need open and objective feedback. They should all share their achievements and what they

have produced. Finally, all communities need to have regular internal discussion and meetings to ensure everyone is interacting smoothly. Great communication should be a goal for the many different parts of your community, as opposed to one specific area. When we get great communication right, community feels vibrant, thriving, and accessible.

We are going to cover all three of these topics in this book. Incoming communication will be discussed in [Chapter 7](#), outgoing will be discussed in [Chapter 6](#), and we discuss internal communication here.

Striving for Clarity

As communication is critical to the success of your community, it is stunning how many communities just get it plain wrong. Many are plagued with long-winded, overly complex, and difficult-to-use communication channels, and it seems you need a degree in rocket science to understand how to join these channels. Then, it seems you need to go back to school to get a degree in computational linguistics to then fit into the culture and expectations of these communications channels. Many have an unwritten rule book stapled to the side of the channel, and if you are unfamiliar with its scriptures, the response can be terse, forthright, and sometimes outright rude. This is the last thing you want. Instead, you want to create simple, efficient, welcoming, and enjoyable methods of keeping in touch with other members of the project.

The greatest form of communication is always going to be sitting in a room with a real person, face to face. In this setting you can speak freely, your words flowing as quickly as your brain conjures them up, with body language, gestures, and facial expressions further augmenting the flow of conversation. Any technological form of communication is going to compromise some of these attributes. Of course, many who are socially uncomfortable in face-to-face situations will hail the muting of these additional attributes in conversation as a victory for accessibility and openness.

When laying down the lines of communication for your community, our goal is to strive for *clarity*. Imagine if you will a world in which every communication is clear, accessible, and well understood by your community. You need to think carefully about the culture in which your community communicates and strive to build a highway and driving style that achieves that culture. You first need to lay the foundations, which can be found in clarity and transparency. Your members want to be able to hear, read, or experience each communication and understand it straightaway. When clarity is in place, contributions will begin to flow shortly afterward. When confusion, misunderstanding, and opacity set in, your members will either spend their days seeking clarification or move on, confused and frustrated.

Clarity and transparency are also important in attracting new members. For example, most online communities' communication channels are indexed by search engines. How many times have you typed something into Google and some of the results that appear are mailing lists, forums, or other online discussions that are happening inside a community? Potential new

community members will read these discussions and it will affect their desire and willingness to join your community. If the communications are complex, socially fraught, or otherwise suboptimal, potential members may prefer to spend their valuable spare time playing *Guitar Hero* rather than joining you and your band of merry men and women.

Achieving clarity requires attention to two areas. First, a sensible choice of communication medium is required (mailing list, IRC, forum, etc.). This is relatively straightforward and actually fairly uninteresting. We will make some decisions about this over the following pages. The second, more complex part is picking communication channels that match the needs of the users while maximizing clarity. Let's spend some time talking about that.

Choices, Choices

Your community has oodles of communication channels to choose from, each with qualities that make sense in different scenarios and to different people. The goal is to match the right medium to your community and to understand the pros and cons of that medium to help the pros bubble to the surface and keep the cons well away from the kitchen.

Picking an appropriate medium is largely about understanding your contributors and their workflow. Each type of contributor will have different preferences. Software developers generally prefer content to be delivered directly to them. They are generally most comfortable with mailing lists and RSS feeds (updated content from websites and online resources) and don't like to have to refresh a browser to see if new content exists. This is part of why many (typically western) developers don't get on very well with forums. Some communities have bridged this divide by providing gateways so a mailing list post goes to a forum and vice versa (an example of this is the Banshee project at <http://banshee-project.org/support/>).

NOTE

Of course, many developers do love forums. The last statement was based upon general experience with developers over a range of projects and development cultures, largely in western countries. If you know developers who love forums, don't be alarmed: we are all friends here.

Users are (typically) different. Users often love forums for their accessibility and simplicity. The conversation flow is clear, the interface is friendly, and the web browser is a familiar window to that world. Users are used to having to refresh their browser to see if updates exist. They are used to visiting many websites to find content, and they generally feel uncomfortable about technical barriers to these discussions and content. Users just don't like to jump through hoops, particularly technical hoops that can easily trip them up.

Believe it or not, these roles are fairly set in stone. Trying to persuade a developer to use a forum can be like persuading a cat to chase after a stick. A developer may agree to give it a

shot, but I can almost guarantee you that it won't work out. Communication channels are highways of habit: people have their preferences and they generally stick to them.

At the beginning of a community you need to know which roles and personalities are most comfortable with which communication mediums; this is the very first step in building great communication. You can then make informed choices. We will explore some of these common mediums and some notes that can help inform these choices in just a moment.

Communication fetishism

Another key consideration when building effective communication channels is keeping discussion focused. This is a two-part process in avoiding *communication fetishism* and also *keeping all your eyeballs in one place*.

Communication fetishism is particularly prevalent in online and technical communities and points to the problem of new communities wanting to provide every possible communication channel under the sun. They set up mailing lists, forums, IRC channels, Second Life worlds, and more. This is the *wrong* thing to do. You should instead identify the key roles and personalities in your project and choose mediums that make the most sense to those roles. Let's look at an example.

When I set up the Jokosher project, I knew that the primary roles in the community were going to be users and developers. I wanted to ensure that the project had communication channels for technical discussion about the development of the application, but also a place for musicians to discuss Jokosher, share tips, and show off their compositions. I decided to set up a mailing list for developers and a forum for the users. These mediums reflected the respective developer and user roles well. The only other medium I set up was a #jokosher IRC channel for real-time discussion and meetings for the developers. Although the channel exists, we never point regular users to that channel; the forum is far more appropriate.

The second part of the process is "keeping all your eyeballs in one place." One of the mistakes a lot of new communities make is to fragment individual communication mediums too heavily. Let's look at another example.

When we started LugRadio, I wanted us to provide a place for listeners to talk about the show and the topics in each episode. Forums were the best choice, so I set them up.

In most discussion forums you can have a number of subforums. As an example, if you had a software project, you could have subforums for General Discussion, Development, Documentation, etc. When I set up the LugRadio forums I created three subforums: General Discussion, Ideas for the Show, and Mirrors. Each subforum had a clear purpose for discussion.

In contrast to many other forums, we had a tiny number of subforums. Many new forums have 10 or more subforums, and we had 3. This was deliberate. When you set up a new community, you want to generate discussion quickly. You want to initiate the discussion but encourage and inspire others to participate and get involved. If you have a forum with too

many subforums, you will fragment the discussion: you will get many tiny bits of discussion across the subforums, and little consistency. Keep the discussion in just a few places, and conversation will flow.

This happens because people waste time choosing the right forum instead of just posting. What is worse, some get confused and just don't post at all. Discussion gets going faster when you have fewer choices.

The Mediums

With some general best practice under our belts, let's now roll up our sleeves and take a look at the nitty-gritty of the range of mediums available to us. Over the following few pages I am going to provide a quick-shot summary of these mediums and many of their behavioral elements.

Before we begin, we need to discuss a key consideration in choosing an effective medium: how easy it is to record and retrieve conversations that have occurred in the past. When communicating together with others online, each medium places different opportunities in the hands of others to find these conversations. This attribute can dramatically adjust how people behave and converse in a given medium.

As an example, it is well known in many communities that mailing lists have their conversations archived and publicly available on a website. This website is then archived and retrievable by search engines, such as Google. With this in mind, participants in mailing lists often act in a more formalized and "professional" capacity: you never know when that conversation on a mailing list will show up when a prospective employer is Googling you for a job interview.

On the flip side, there is the real-time IRC chat medium. While many IRC channels can and are logged, it is far harder to find the logs and find relevant discussion in the reams of chatter that happen each day on an IRC channel. As such, the medium feels less recorded, and many of the participants feel more comfortable in communicating more socially.

You should consider these implications when building out your communications facilities, and ensure your participants are fully aware of where they stand. No one wants a situation where someone says something, assuming it is not archived, and then finds (to his chagrin) that it appears in a Google search.

Mailing lists

Mailing lists are an excellent medium for discussion. They are low bandwidth, a familiar interface (email), and fairly accessible, and conversation is delivered directly to your email client. The delivery of the conversation reduces the chance of new contributors forgetting about your community: each time they check their email, they are reminded that your community exists, and they may actually read the messages and respond.

Mailing lists are typically preferred by software developers, and of common interest in the open source world. Many nonprogramming contributors often use and enjoy mailing lists.

Mailing lists are not all sweetness and light, though. Joining a mailing list is a little complex. It requires people to know how to join, sign up with an email address, respond to the mail, and know where to send messages. Also, unforgiving spam filters can make the whole process icky.

Mailing lists also assume that you are interested in all discussions, and when you join, all discussions come to you. This makes mailing lists implicitly of most interest to those who plan a serious contribution or interest in a community.

Another issue to bear in mind with mailing lists is their lack of immediacy. As an example, if you are a community software developer working at 10 p.m. on a Saturday evening and you have a problem, you could post a question to the mailing list, but you may not get an answer until Monday or Tuesday and are thus stymied. If more immediate results are needed, real-time communication channels such as IRC are better (of course, assuming people are on the channel: silence sucks on all mediums).

Normal end-user and consumer reaction to mailing lists is mixed. Many users have single-shot queries and questions, and the requirement to battle through the somewhat complex sign-up process and then receive all conversations can be off-putting.

NOTE

An important consideration to make when setting up a mailing list is whether the list is archived on the Internet. If you want the conversations to be private (such as if the mailing list is for a governance body), you should close off the archives. We discuss governance in detail in [Chapter 8](#).

Discussion forums

Forums are a very popular, low-barrier-to-entry medium. They manifest in the form of websites that allow you to create an identity and have a discussion using that identity.

Forums have exploded in popularity in recent years, and some huge forums have developed across the Internet. At the time of writing, the biggest in the world is an Anime role-playing community with over 15 million members and over a billion posts. Now that is a lot of tentacles and oversized eyes. Even the far smaller LugRadio forum has over 1,000 members and 40,000+ posts. A common use for forums has been as a support channel for your community. Many software projects (including Jokosher) have found this useful.

Forums are a popular choice among less-technical users. They are easy to join and easy to use, and many forums allow users to inject personality into their profiles with avatar pictures, personal details, and signatures. Many forums also encourage discussion via the use of ranks—special community descriptions based around the number of posts that a member makes. Many forum users cherish these rankings and take great pride in making many posts to achieve them.

NOTE

Be careful with these ranks as a means of measuring community; they are often not all that insightful. We will discuss this in [Chapter 7](#).

Due to the user-focused nature of forums, discussion is often from the perspective of new users, and often those who are less familiar with netiquette. As such, you may attract some immature behavior and spam on forums. With this in mind, you should ensure you have a number of moderators who can tend to this kind of behavior. The best approach to moderation is to anoint a handful of well-respected and trustworthy contributors who use your forum.

IRC

IRC is a real-time chat medium that has become increasingly popular for communities. There are IRC providers all over the world, and many IRC networks cater to specific purposes. As an example, the Freenode network (*irc.freenode.net*) is specifically aimed at providing IRC channels and conversation for open source projects. Although an entirely open and accessible medium, IRC is still very much populated by technology-related channels.

The value of IRC is in real-time discussion, and there are many benefits:

Bonding

Communities bond effectively in real time. It provides a chance for people to have general day-to-day conversations that would be out of place on a more formalized medium such as a mailing list. Many people use IRC for socializing and often spend time chit-chatting; this all helps for bonding.

Speed

Discussions can often occur faster on IRC. It provides an excellent medium for discussing and debating, and it is smoother and easier to debate on IRC than on a mailing list or forum.

Meetings

IRC offers an excellent opportunity for real-time meetings. We will discuss this later in this chapter and in [Chapter 10](#).

Logs

IRC channels can be logged, which provides an excellent means of documenting discussions and meetings.

If you are setting up a community that is focused on technology or the Internet, an IRC channel could be a useful addition. It is important that your IRC channel is open, accessible, and well publicized. You should ensure it is listed on your community's website.

NOTE

Many IRC services are blocked from commercial work environments, and some of your members may therefore not be able to take a sneaky IRC break while at work.

There are some web-based IRC clients that may solve this problem for those users, and you may want to list some of these alternatives on your community's website.

As with any medium, IRC attracts its own kind of personalities, but the IRC personality profile is fairly mixed. On one hand, we see many developers making use of IRC every day to keep in touch with their communities. We also see many new users joining IRC. You should ensure that your IRC channel(s) cater to these different types of user. If you are running a development project, it is best to have separate channels for developers and users (such as *#myproject-dev* and *#myproject-users*, respectively).

An interesting personality trait in IRC that you should be aware of is the desire for power. Many communities have discovered that there is a lot of importance placed in IRC channel moderation and control. IRC allows people to become channel operators, and these privileges allow people to be kicked out and banned from channels. Many communities have experienced power struggles, arguments, and bickering over these privileges: being an operator is a badge of honor for many people. You should give out channel privileges sparingly, to those who will treat those privileges with care.

NOT SHOWING POWER

In all IRC channels, some people can be operators and have the power to kick out or ban people from the channel. By default, an icon appears next to the operators in your channel. Many channels decide not to have the operators display their privileges in general.

On one hand, this makes sense: if there are no visible operators, the channel feels very equal. On the other hand, having visible ops can be self-policing because it can ensure people behave properly. My recommendation: if someone is an operator, ask her not to display it, but if someone needs kicking out of the channel, have the operator switch on her privileges and firmly plant that boot in the offending user.

Leading by Example

However you decide to lay down the lines of communication and whichever tools and mediums you decide to use, you should strive for communication in your community to be friendly, clear, and productive. You want your community's members to not only feel engaged, excited, and enabled, but to enjoy their interactions with the rest of the community.

This may not be quite as easy as it sounds, however. Communities are by definition groups of people, and those people have different personalities, habits, opinions, and approaches. These nuances can sometimes complicate how we communicate.

People can and do get frustrated with each other, have mismatched expectations, engage in unclear and tense discussions, and indulge in other agitating agendas. Not only that, but people sometimes just get out of bed on the wrong side. Everyone has bad days. Confusion happens. Life will get in the way, and people get busy.

Many of these causes are entirely innocent and simply the nature of being human. With people being the catalyst of community, you could be forgiven for thinking that there is nothing we can do about these problems in communication.

Not so.

Communities are vessels of culture: every community has its own norms, and these norms are defined by community leaders—community leaders such as *you*. You can avoid many of these potential issues in communication by not only setting a strong example but also inspiring the wider community to follow a positive example in how they communicate with each other.

To lead by example, we can break the problem down into the following two bite-sized chunks:

Daily communication

The day-to-day discussions, ideas, and debates that are expressed across your range of mediums.

Longer writing

Longer, more considered writing, articles, and other pieces that are often used to inspire, advise, and direct your community.

Let's now take a spin through both of these different areas and explore some best practices, tips, and tricks for communicating well.

Daily Communication

Leading by example means communicating in a way that you feel is representative and appropriate for you and your community. It is about inspiring your community behind an ethic of good communication and ensuring that your community is exposed to that ethic on a daily basis. Life has taught us that consistent exposure to high-quality content can influence raised

quality in the onlookers. Great music often inspires great musicians; great art inspires great artists; and great community inspires great community members.

To lead by example, you need to be the living incarnation of the advice you offer to your community. As community leaders we often settle ourselves happily into the role of *Chief Advisor of Doing Things the Right Way*, and it is hugely important that we practice what we preach and get intimately familiar with the taste of our own dog food.

Being a great daily communicator is not about being an incredible writer, proficient speaker, or using words reserved for a Scrabble-winning smackdown. Great communication instead focuses on clarity, detail, objective thought, and a consistently high quality of interaction.

Teaching great communication is complex, and many books have been written on the subject. Fortunately, becoming a great communicator doesn't require an exercise in academically satisfying hand waving or an attempt to sound like a monocle-wearing intellectual, but to simply be clear, friendly, and straightforward in your communications.

To achieve this, there are two stages involved. First, you should familiarize yourself with some simple recommended rules of engagement. These are:

Be clear

Always try to communicate as clearly and transparently as possible. Speak frankly and use language familiar to your recipient. Try not to blind people with science, but don't patronize them either. Always try to craft your communications to your audience.

Be concise

Keep to the point, and don't weigh your communications down with babble. Don't use 1,000 words to say what could be said in 100. With many of us receiving so many emails, messages, phone calls, and other distractions every day, don't burden your community with unnecessary rambling. If an email takes longer than five minutes to type, you may be doing something wrong (or you are a really, really slow typist).

Be responsive

You don't have to be wedded to your computer, but try to get back to people within a few days of them getting in touch with you. If you are drowning in emails and work, just let people know you might be a little delayed, so their expectations are set correctly. This issue is applicable not just to personal communication direct to you and other community leaders, but also to mailing lists, forums, IRC, and other public channels. Put yourself in the sender's position: it is impossible to tell the difference between "nobody is answering my question because nobody knows the answer, meaning that what I'm trying to do is impossible and I should try something else" and "nobody is answering my question even though everyone knows the answer because they're all too busy, meaning that I should sit and wait longer rather than abandoning this approach." It helps to make your community seem more friendly to new members and outsiders if they can tell the difference between these two things.

Be fun

One of the biggest mistakes that people make when they become well respected in a community is to hide their personality in the interests of “looking professional.” Let your personality shine through. Make jokes and witty comments, and be sarcastic.

Communities are supposed to be fun, and this is an important part of leading by example.

Be human

We are all human, and we all make mistakes. If you screw up, say so and apologize. People will cherish your honesty and your integrity to hold your hands up when you get it wrong. This is a critically important part of leading by example: you want your community to also accept when they get it wrong. What we want to avoid is defensiveness, because it causes people to enter into a game of rebounding defensive statements, which is frustrating and damaging. If your hands are tied in being frank and open about your mistakes (such as if your employer would be less than thrilled), identify what went wrong and try to secure confidence in your community that it won't happen again in the future.

The second step is to learn from others. Observe email, chat conversations, phone discussions, and other interactions, and deconstruct them in your mind to find nuggets of wisdom and inspiration. You are sure to find some people who are always clear, concise, and detailed in their communications, and these are excellent role models for yourself.

Netiquette

In many online communities there are often localized technical cultural norms that are in play. These are small conventions and methods of communicating that are typically localized to a community, be it online or physical. Respect for these conventions and (often unwritten) rules is known as *netiquette*.

Although most of you will be familiar with some of the common conventions of communicating well on the Internet, you should be familiar with some of the local conventions in your specific community.

As an example, many communities use mailing lists as a primary method of communication. In many technical communities, some members can get rather agitated when new members unwittingly use an approach called *top posting*. Let's illustrate this with an example. Imagine you are subscribed to a public mailing list. You get the following simple email:

```
Hey All,  
I just wanted to ask what is the answer to Foo?  
Thanks!  
Bob
```

Top posting is when you reply above the quotes message. As an example:

```
Hi Bob,  
The answer is Bar.  
Jono  
> Hey All,  
> I just wanted to ask what is the answer to Foo?
```

```
> Thanks!  
> Bob
```

This is often frowned upon. The reason is because in online discussion threads with a series of messages replying to pieces of previous messages, top posting makes following this thread of discussion difficult to follow. The following is an example of when someone replies to a top-posted message but below the message:

```
> Hi Bob,  
> The answer is Bar.  
> Jono  
>> Hey All,  
>> I just wanted to ask what is the answer to Foo?  
>> Thanks!  
>> Bob  
OK, thanks for the reply, Jono!  
Bob
```

The conversation is already a pain in the nether regions to follow. If everyone responds below the quoted text, the conversation would instead look like the following at this point (particularly when cutting out the bits of unneeded conversation):

```
> Hey All,  
> I just wanted to ask what is the answer to Foo?  
>> The answer is Bar  
OK, thanks for the reply, Jono!  
Bob
```

Though this might seem like a minor detail, many online communities get surprisingly frustrated at these kinds of mistakes. Unfortunately, communities often fail to document these expectations around discussion, and many innocent bystanders get short shrift when they accidentally get it wrong.

You should always try to document these expectations around communication on your community's website. If you are new to a community, it is recommended that you first observe the communication for a few days before you participate. This will help you to get a firm idea of these social conventions.

Avoiding bikeshedding

Almost every technical community I have seen makes the same two common mistakes: obsessing over resources and discussing topics to death. The latter of these two problems is rather excitingly titled *bikeshedding*.*

In many collaborative communities, one of the first communication resources to be set up is a mailing list or forum. As with any new community, excitement is high, and the thrill of communicating with your new community members from all over the world gets everyone pumped up and fills these communication channels with discussion.

* To learn why, visit http://en.wikipedia.org/wiki/Parkinson%27s_Law_of_Triviality.

With the communication lines open and an unbridled sense of excitement around the shared goal of the community, plans get made posthaste. Ideas, thoughts, commitments, and other agreements get discussed. The community gets its first opportunity to dream together, and it feels like the world is your oyster. The result of this exciting brainstorming period is a stack of incredibly exciting plans and ideas. All that needs to happen now is to make them reality, right?

Unfortunately, it's not quite that simple.

Talking is always easier than doing. It is far easier to discuss a great idea (often in exhaustive detail) than to implement a great idea. Reams and reams of discussion from many interested parties can explode on your mailing list, but this in itself can be a blocker to producing the very projects you discuss. While up-front specifications can help communities build something, over-specification and information overflow can have the inverse effect.

Part of the problem is that many communities focus a little too much on having every "i" dotted and every "t" crossed before work begins. The issue here is twofold. First, too much discussion will typically result in a hugely complex plan that is difficult to keep track of if not well documented. This can be off-putting for volunteers who just want to get started with something fun and manageable. When you overdiscuss and overengineer an idea, the complexity can be daunting.

The second problem, particularly in the early stages of your community, is that your new members need to see results or they will grow bored and leave. Great communities are often built by a small set of capable and fast-moving contributors right at the beginning of its evolution. If a community spends three months discussing an idea yet produces little or nothing, many potential contributors will grow impatient with the discussion and develop a sense that the community is nothing more than a talking shop. This is exactly what we want to avoid.

There are two approaches that can help avoid this problem. First, you should avoid discussing topics in too much detail without connecting them to specific plans. Ask who is going to work on what, when the work is likely to be completed, and how you can help. As an example, if you are working on a software project, you should aim to have some code written within a week of the project starting. Always keep your eyes on the prize: reaching the goals of the community, be it software, an event, a campaign, or whatever else.

Another solution is to build a culture of experiments in which your members channel their energy into a specific project that can be proposed to your community at a later date. This has been a popular approach in the open source GNOME desktop community: many developers have worked on their own pet projects that involve a handful of developers. These projects are created, enhanced, and refined, and when they are mature enough, they are proposed for inclusion in the main GNOME desktop. This is an excellent, productive, and engaging approach to creating things together.

BUILDING A “CAN DO” CULTURE

It is also important to set your own expectations correctly when it comes to your new community members.

The vast majority of people who join your community will be passive and contribute little more than opinion and commentary. This is perfectly normal. You should go into your community and expect bikeshedding to be a very real threat. You should expect that most of your new community members will contribute little in the way of practical results.

You instead need to build a culture around doing things. You need to inspire your contributors to make things happen. You should speak positively about your goals and encourage everyone to contribute to their implementation, no matter how small the contribution.

Longer Writing

In addition to day-to-day discussion, email conversations, online chats, and other elements, great communication often calls for longer, more considered pieces and articles. These longer chunks of writing are often required in a number of different places, including:

Blogs

With the continued growth of blogs as not only a personal publishing medium but also a community publishing medium, you may need to write articles, stories, and features about your community.

Emails

Many community leaders send out longer, more considered, and more strategic emails that help to guide the community in its direction.

Magazine articles/web articles

Your community may get the opportunity to be featured in a magazine or website, and you will need to be able to present your thoughts well.

Documents

Your community will produce documents, guides, specifications, help, and more, and these should be clearly written and easy to read.

Writing well happens at both a mechanical and stylistic level: you first need to produce properly written language and then build atop this with words, expressions, phrases, and approaches that are interesting to read. Let's explore some of these topics now.

The mechanics of writing

The first step in being a good writer is to learn the mechanics of great writing. At this level we are absolutely focused on the nuts and bolts of writing: spelling, grammar, and punctuation.

For some people this is the be-all and end-all of communication: producing “perfect writing” that follows strong editorial guidelines. Using our previous “it’s not the notes you play, it’s the notes you don’t play” quote with which I opened this chapter, the mechanics of writing are our metaphorical notes. Though it is absolutely not the be-all and end-all, writing in a “correct” way does offer a baseline that all writers should build on. Producing fantastically literate, illustrative, and textured writing just doesn’t work if your spelling and grammar suck.

To achieve this baseline, I recommend grabbing a good book on the topic. There are plenty to choose from, but don’t feel you need to devote your life to it: pick a short book that covers the main points, digest it, and use some of the advice it presents. A great read that I recommend for those of you who want to brush up on this is Strunk and White’s classic *The Elements of Style* (Longman). While you read and learn the topics, also look at other people’s writing and try to learn what makes their writing structurally interesting and simple to read.

If there is one piece of advice I would recommend more than any other, it is to proofread your work extensively. Proofreading offers an excellent chance to read the piece and ensure it structurally fits together.

Let’s use this chapter as an example. If you take a look at the range of content that is covered, it has been laid out in a very specific way. The content has been structured to discuss different topics, each building on what went before. As I write these words right now, I have already written a significant chunk of content for the chapter, and although I could proofread this specific section on “[The mechanics of writing](#)”, I need to proofread the entire chapter to ensure this section fits in and flows correctly.

When many people write, they first create the main chunk of text and then read the whole piece again as their proofreading pass. This is fine, but I would encourage one more step before the proofread: read the entire piece aloud. Although this may sound a little crazy for a piece that is intended to be read silently, reading your content aloud can be hugely valuable in finding badly written sentences that don’t flow well.

CULTURAL DIFFERENCES IN WRITING

One additional element you should be aware of is just how much writing varies around the world, even within the same language. Aside from some of the spelling differences (e.g., color and colour, prioritise and prioritize), there are also changes in language use itself.

One example of this is how active the voice is. Some countries speak in a very passive voice (e.g., “Today we will do xyz”), whereas some use a more active voice, which is often interpreted as more conversational (e.g., “Go and do xyz”).

You should ensure you write in the manner that most suitably targets your audience. If it is an open-ended audience, simply pick one approach and use it consistently.

Don't write like an institution

When I was at university, I used to write articles about Linux and technology. The work was great for furthering my career while also funding my somewhat ludicrous lifestyle of working and partying.

I eventually went on to make writing my career when I graduated. Each day I would write countless articles for various magazines, and throughout this time I focused my radar on all manner of different topics. While most of my research was performed online or on the phone, much of my in-person research time was spent in press conferences.

Press conferences and engagements were interesting beasts, mostly following the same formula. We the press would represent the cynics of the world, and there the marketing folk would represent the optimists of the world. They not only fed the press coffee and small sausage rolls, but also a diet of unparalleled marketing nonsense.

Throughout my career as a journalist I witnessed example after example in which concepts and topics were dressed up with buzzword-laden diatribes that were often so jam-packed with marketing fluff that it would be barely understandable for us tech media hacks, let alone the average Joe.

One of the finest (read: most disturbing) examples of this happened when an organization (that shall remain nameless) sent me one of their new press releases. As I sat there one morning sipping my coffee, desperately trying to wake up, I was presented with this humdinger of a no-nonsense buzz-fest:

Program Focuses On Helping The Open Source Ecosystem Grow Sustainable Businesses By Implementing A Community-Leveraged Model

Unfortunately, the literary smackdown didn't end there. It went on to say:

XXXXXXXXXX, a leading provider of commercial open source middleware solutions for database high availability, today announced XXXXXXXXXXXX. The program is focused on creating a rising tide for the broader open source ecosystem, and is focused on leveraging community-driven development and frictionless distribution to extend the ecosystem.

The language in this press release was by no means unusual. Plenty of companies would send paragraph after paragraph of overcooked verbiage pouring into my inbox across press releases, briefings, product announcements, and other material.

Of course, there was a lot of sense in targeting writing to a specific audience and using language and references that are meaningful to that demographic (many of which were businesses and buyers). Unfortunately, in many of these cases the abundance of completely meaningless bluster manufactured in marketing meetings with the intention of sounding professional merely complicated as opposed to clarifying the matter.

You should be careful to not fall into this trap. If you bring out the big literary guns with your community, you not only face the risk of confusing people but also agitating. Remember, communities thrive on clarity and transparency: if you start talking like a slick-haired marketing copywriter, some members of your community are likely to interpret your words as dressed-up claptrap. Speak with your own voice and you avoid this problem. If you are a community manager who reports to a marketing department, this may seem like quite a challenge. Instead, see it as part of your responsibility and talent in helping to translate between these two sets of language so you can deliver the message in the right way.

One of the widest-respected writing teachers, William Zinsser, speaks extensively on this topic in his seminal *On Writing Well* (Collins), a book devoted to writing clearly. When I started writing for O'Reilly, the nice folks there kindly sent me a copy of the book. Curled up on my couch reading it one afternoon, one paragraph in particular on this topic leaped out at me:

Just because people work for an institution, they don't have to write like one. Institutions can be warmed up. Administrators can be turned into human beings. Information can be imparted clearly and without pomposity. You only have to remember that readers identify with people, not with abstractions like "profitability," or with Latinate nouns like "utilization" and "implementation," or with inert constructions in which nobody can be visualized doing something: "pre-feasibility studies are in the paperwork stage."

Zinsser is a staunch advocate of clear writing. He believes the enemy of great writing is the modern expectation that writing must be laced with buzzwords and meaningless terms to make it sound "more professional." Earlier, when we discussed that "it's not the notes you play, it's the notes you don't play," the professionalism that many seek to achieve in writing is absolutely the notes you don't play. These meaningless buzzwords are the equivalent of slamming your hands on the piano to get attention.

Untwisting the tail

As I step down from my high horse, I have to acknowledge that distilling complex processes and concepts into simple language without blinding people with science or meaningless words is really quite hard. It takes skill to explain these topics in a way that is clear, easy to read, and devilishly simple to understand. While I am eager not to turn *The Art of Community* into a creative writing textbook, let's look at one quick example in which complex issues are communicated easily and effectively, in a world intimately familiar with verbosity: *licensing*.

The Creative Commons is an organization that produces a set of free licenses that allows content producers to freely license their work. Founded by highly respected law professor

Lawrence Lessig, the Creative Commons has gone on to achieve remarkable success, and thousands of artists are licensing a range of music, video, art, and other creative endeavors under their range of licenses. Each of their licenses promotes and encourages a so-called Free Culture—that is, the freedom to share, remix, and derive from existing work. Their range of licenses provides different levels of these freedoms depending on the desires of the artist.

The creation and free availability of these licenses is a wonderful gift from the Creative Commons, but an even more impressive contribution is the ease with which the licenses can be understood. An example of this is the Creative Commons Attribution-Noncommercial-Share Alike license. This is the very license that *The Art of Community* is published under. Figure 3-1 shows what the license looks like.

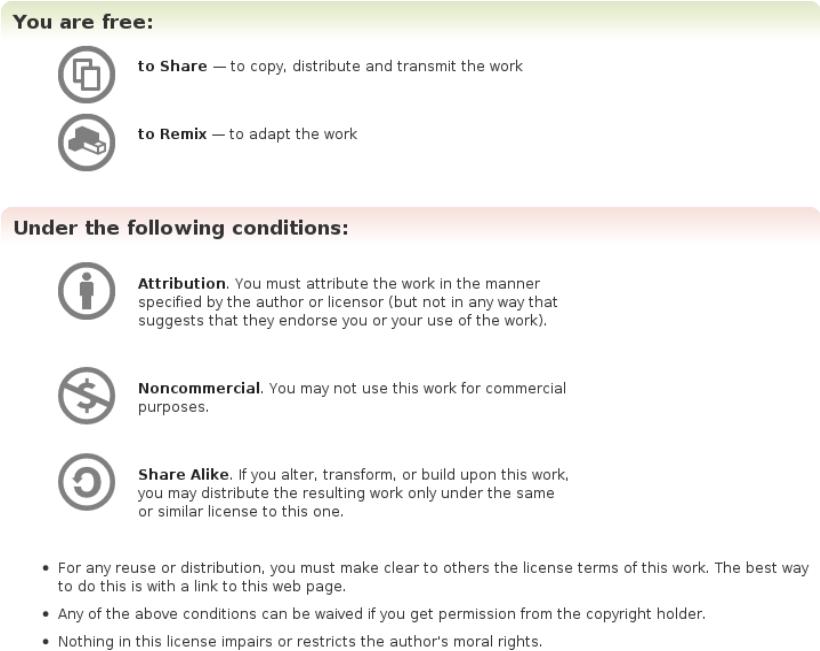


FIGURE 3-1. Simplification of licensing from the Creative Commons

Traditionally, licenses have been ugly beasts. They haven't evolved far toward readability, and they include reams and reams of complicated legal mumbo jumbo to confuse the poor artist.

The Creative Commons developed the concept of the Commons Deed, shown in the figure, which showcases the primary features of the license in a simple, easy-to-read, and visual way. They managed to transpose these attributes of a license in such a way that is simple for an artist to learn and apply to his work. It made perfect sense: if the licenses were too complicated to understand and use, artists will either (a) not use the licenses, or (b) more worryingly, pick a

license that they don't understand and suffer negative consequences around the use of their work in the future.

To help improve the accessibility of their licenses, the Creative Commons produces two documents for each one. First is the full-blown legal document with the nitty-gritty of how the whole deal works. If a case ever got to court, this legal smörgåsbord would be the primary point of reference. That document is great for the legal eagles, but not for artists. As such, the Creative Commons then went on to convert this 3,000-plus-word document into the simple Commons Deed shown in [Figure 3-1](#). The Creative Commons realized that licensing simply needed to be easier to understand for the real consumers and propagators of the licenses: an exercise in clear and targeted writing.

Zinsser's earlier observation about not writing like an institution is an important lesson when documenting all aspects of your community, but particularly processes. When producing documents that you want your community to read and act upon, don't be lured into filling them with meaningless buzzwords, overly complex descriptions, and other fluff. Take Zinsser's advice and try to write and produce documents that are simple, easy to read, and designed for the audience who should read them. Simplicity in writing will ensure your community can do great work easily.

ME, MYSELF, AND I

A useful tip for writing clearly and humanly is to write as if you're talking to a colleague or friend sitting next to you. That kind of clear but informal banter makes the reader feel comfortable and cared for. This combined with proofreading aloud, as we discussed earlier, is an excellent way to bring a conversational finesse to your writing.

Setting tone

Tone is an important element in all writing. Tone is the impression and "feel" that you take away from a piece based upon how it is written. Tone can often be subtle and entirely subjective in how it is interpreted. Let's look at an example. Here are two different ways of writing a piece asking people to write reports about their work, each with a very different approach to the tone:

Way 1

Hi, everyone! As our community continues to grow and rock the world, it is difficult to keep track of what we are all working on. As such, I think it would be awesome to produce reports that can help us all to know what we are doing. It should only take a few minutes, but it could be a great contribution and well worth the time. Thoughts? :)

Way 2

Hello. Could everyone please produce a series of reports for their work to facilitate the community in understanding which work is being tended to? Thank you.

With the first piece the tone is very open, friendly, and participatory. The language is loose, and the writing openly asks for comments and feedback. The second piece is far more formal, and a little more demanding in asking people to provide reports. Although both are clearly written, the choice of words and structure affects the tone.

Tone is hugely important when writing content for your community. Following on with our “notes you play” quote, it is tone that is the finest example of “the notes you don’t play.” The tone you choose can hugely affect how the piece is interpreted.

There are two different elements to consider when thinking about the tone of your writing:

General tone

One of the most important considerations to make with your writing is the general tone that you use throughout your communications. For most of us this is baked in: your tone is expressed each time you write, and it largely reflects your general personality. You should endeavor for your tone to be loose, open, friendly, and inviting. You should feel entirely comfortable about mixing in humor, and you should write in a way that makes your community feel like equals.

Context-specific tone

Depending on the context, you will need to adjust your tone to be appropriate. The choice of tone varies significantly between contexts that involve conflict, getting people excited, and governance topics.

Your choice of tone and how you apply it will significantly affect how you are perceived in the community. This choice does require balance, though: being too happy-go-lucky can have the same effect as being a grumpy curmudgeon.

Imagine a line drawn in the sand in front of you. On the far left of the line is an overly chirpy, overly friendly, and potentially disingenuous level of bright smiles and happy thoughts. On the far right side of the line is a grumpy, monotonic, miserable, and snappy approach. Although it is easy to see how those on the far right will rattle people’s cages, those on the far left create not only potential annoyance and frustration, but also mistrust: if someone is over-the-top happy in their communication to everyone, it could be interpreted that the person is doing it for effect and not being honest with the community. As ever, a balance is needed. Be yourself: err to the left side of the line, as we like friendly people, but use your best judgment to determine how far toward Happyville you edge.

Within the realm of tone, many different people take different approaches; some work well and some don’t. Compare and contrast the frankness and pointed tone used by a New York investment banker and the friendly and submissive tone of a San Francisco charity worker.

Let's now look at some ingredients that can be used when considering your tone and how they can affect your community:

Seriousness

Communities react very differently to seriousness in different scenarios. As a general rule, you should keep your community loose and informal, but at times you will need to be serious and focused. You should fully expect a more serious tone when discussing conflict, governance topics, or personal problems with a specific community member.

Humor

Humor is a huge boon when used right with community. If you are funny, your community will likely take to you, but your humor needs to (a) actually work (yes, you need to actually be funny), and (b) not distract from the topic. You need to always ensure your humor matches the context: as an example, humor in an emergency meeting will probably go down like a lead balloon.

Witty

Closely related to humor is wit. If you are quick-witted, particularly in an amusing way, it can bring a loose and fun atmosphere to your community. One risk you should be aware of, though, is to not use your lightning-fast wit to embarrass or show up a fellow community member. Also, wit is typically not very welcome in contentious or argumentative scenarios, so tread carefully there.

Frankness

Being frank and to the point can be received in many different ways. Some people who receive this kind of to-the-point approach will respond with a very defensive and confrontational tone, whereas others will relish your frankness. Take the pulse of the people you are talking to, and adjust this element of tone based upon how well you think they will react.

Your choices and approach to tone will adjust, grow, and mature over time. As with everything else when working on your communication, it is always valuable to look below the surface of other people's communications to learn how they have approached situations. As an example, if you are dealing with a contentious situation, look at the tone of how others have responded and learn from them. You may also want to directly ask people who have been there before for advice. We will be looking into conflict resolution situations extensively in [Chapter 9](#).

SOMETIMES WE ALL SUCK

One attribute of tone that I have always found compelling is those who are willing to be slightly self-deprecating. This is a particularly welcome trait in leaders: when a leader is willing to joke about herself or her situation, it can often help warm her reception to others.

Naturally, you should do this only if you feel comfortable doing so: faking self-deprecation just sounds patronizing.

Inspiring your community

One of the most fundamental qualities that a community leader can bring to a community is inspiration. At their heart, communities are collections of people united by a shared ethos and most typically by a shared opportunity and goal. Inspiration does not only bolster everyone's commitment and excitement around that goal, but it can build a sense of unity and togetherness that is essential for a thriving community.

Unfortunately, there is no recipe or formula that teaches you how to inspire. Inspiration is a hugely complex topic to teach because it is heavily dependent on your approach and the kind of community that you have. Part of the reason for this is that your approach is in itself also highly dependent on you as a human being.

Being able to inspire is based heavily on whether you are believable as an inspirational figure, and your approach needs to reflect you as a person. As an example, if you are a generally fun and jovial person but try to deliver a Winston Churchill monologue, it will be laughable at best and seen as a desperate attempt for validation at worst. On the flip side, if you are a serious and forthright person and you deliver a George Carlin-style comedic slap-down, your audience will probably feel the same way they do when they see their dads edging toward the dance floor at a wedding. You need to match the approach to inspiration with what qualities and attributes you as a person already have. When you make inspiration your own, it becomes other people's inspiration.

The best advice I have for writing inspirationally is to discover who and what inspires you and learn from it. Everyone has his own individual answer for this question. Some are inspired by musicians, authors, politicians, and people related to their field of interest or expertise. For me, one of my most significant inspirations is the unique combination of Aaron Sorkin and Martin Sheen.

Sorkin is the writing genius behind *The West Wing*, a show written about the goings-on of an American administration in the West Wing of the White House. Every time I watch the show, I feel inspired by the dialogue. I love the way the characters speak to each other. I love how their directness of conversation is underlined with humility and humor. Being the ultra *West Wing* nerd that I am, I have every episode on DVD, and those shiny little discs are littered with incredible examples of inspiring writing and the masterful delivery of that writing by Martin Sheen as the President. Whenever I watch the show, my own use of language changes a little and my own approach to writing adjusts a little more. It inspires me.

We should, though, keep things in perspective: *The West Wing* was written by a skilled team of writers who spend every day working to make those little hairs on the back of your neck stand on end. It is important not to judge yourself too heavily in what you produce. When you find your inspiration, pick it apart and try to discover what about it inspires you, but ensure you merely use it as a guide. Don't expect to become the next Aaron Sorkin overnight. Just focus on the most immediate goal of learning, sharing your thoughts using what you have learned and seeing how your community reacts.

FINDING YOUR INSPIRATION

You should spend some time trying to figure out where you find yourself thinking and writing inspirational thoughts and ideas. For some it is in the shower, when taking a walk, or while lounging around in bed. Everyone is different. Everyone has times when they feel inspired and creative.

The next time you feel inspired, note what you were doing when it happened. It is likely you will see a correlation. When you know what the magic scenario is, you can trigger it and get some more quality inspiration time!

Summary

Throughout this chapter we have explored some of the many topics involved in building a culture of simple, powerful, and effective communication. While all of the topics covered in this chapter are firmly seated in the "communication best practices" category, it should be noted that effective communication is something we should exercise across all of our community-building efforts.

With this solid chunk of great communication best practice under our belts, let's continue our application of simplicity to community by applying it to an area often bastardized by utter complexity: building *processes*. Fasten those seat belts, friends.

Processes: Simple Is Sustainable

“Light is the task where many share the toil.”

—Homer

RAY KROC WAS A FAIRLY SIMPLE GUY. As a small-time salesman in the 1950s, he had moved on from the thrill-seeking paper cup world to sell multimixer drink machines to restaurants across the U.S. His machines allowed anyone with rudimentary operating capabilities to create one of those delicious creamy milkshakes that required the power of a thousand vacuum cleaners to suck it up the straw.

Kroc’s machines were a hit, and it wasn’t long before his path crossed with Dick and Mac McDonald, owners of a small restaurant chain called McDonald’s. A partnership was born, but Kroc’s ambitions went way beyond milkshake-making machines. His ambitions also went way beyond those of Dick’s and Mac’s. Before long, Kroc purchased the chain and started growing his empire.

From such humble beginnings, McDonald’s now serves over 47 million people every day. Kroc, starting with his inventive approach to milkshakes, died with a fortune of over \$500 million. Boy, that is a lot of milkshakes.

Love it or hate it, McDonald’s has been hugely successful. Its food is cheap, readily accessible, efficient (in an increasingly busy world), and of typically acceptable quality. Of course, some of you will disagree with the last part, but let’s put any gripes to one side for a moment. I am more interested in the mechanics behind the meals.

In 31,000+ restaurants in over 119 countries, each McDonald’s restaurant serves an average of 1,500+ customers every day. Its menu is packed with products, each of which has its own

preparation method, ingredients, and presentation. But let's be honest: the average McDonald's employee is not exactly a master chef. Most of the people who work at McDonald's are untrained, and the training they do get is rudimentary. McDonald's low cost of staff is a key area in which it competes effectively—but how does it do it?

Like Henry Ford before it, McDonald's has broken complex processes into sets of steps that almost anyone can replicate. If you were going to cook fries at home, you'd have to consider a lot of variables: the time it takes to heat the oil, how small you need to cut the potatoes, cooking time, draining, and making sure not to inadvertently plunge your hand in the oil (an ever-present risk when I am in the kitchen). McDonald's has eliminated the need for this range of knowledge by converting the cooking process into a series of simpler steps: unpacking boxes, filling containers, pressing buttons, etc. McDonald's has perfected process simplification.

The inverse of simplification is complexity, and we don't like complexity around these parts. Complex processes are ugly beasts, and their effect is to merely build *bureaucracy*.

You need to avoid this 11-letter word at all costs. Bureaucracy is the enemy in this chapter: it is the vitriol that breaks down the opportunity, potential, and belief that we celebrated so strongly earlier in this book. Great processes blend into the background, functioning as required and as expected. Great processes let people get on with doing real, human, interesting things. Bad processes serve as nothing more than a dartboard for your contributors to throw their frustration at.

Great processes are our aim in this chapter. We are going to assess the many and varied processes involved in your community, break them down into steps, and make them as simple and effective as possible. Processes are everywhere in communities, determining how:

- New people join.
- People submit contributions.
- People collaborate together.
- People deal with conflict, and so on.

It is no coincidence that the word “people” is in each of those examples: people are the foundation of community, and for us to ensure these people can work well together, we need to focus on people at the foundation of our processes.

When we (a) know which processes we need to create, and (b) make them delightfully simple, our community members can get on with enjoying the community that they signed up for.

Eyes on the Prize

One of the challenges of writing a chapter about processes is that they are so abstract. The word “process” is unspecific: it describes a set of steps that achieves something. For newcomers to the subject, this is difficult to get your head around without specific examples of processes. This

could include the examples we just spoke of: how new people join your community, submit contributions, collaborate, and deal with conflict.

When thinking of how to approach this chapter, as well as the wider book, I am keen to not merely provide you with answers to specific problems. This reminds me of the old Chinese proverb:

Give a man a fish and you feed him for a day. Teach a man to fish and you feed him for a lifetime.

I want to teach you all how to fish. I want the content in this book to help you grow and build best practices that can be applied in a range of scenarios. This absolutely applies to processes.

But this brings us back to our original conundrum: processes can be a little abstract and difficult to understand if not provided with an example. As such, as you read this chapter always consider how you can apply the content to the examples we discussed earlier, such as encouraging new members into your project.

Throughout this chapter we will discuss how to build processes that are smooth and effective, explore what is important to consider when building them, identify our needs, and document and communicate them in your community.

Keeping Things in Perspective

Right now I can imagine that some of you are not exactly exhilarated at the idea of reading a chapter all about processes. I admit that even I, while planning this book, was initially hesitant about devoting a chapter to processes. Some of you may see processes as a waste of time, an overly formal mind game that makes life less fun for everyone involved. Some of you may even go as far as suggesting that to define processes is to limit community and entangle its members in rules. But just invert this oversimplification, and you will be closer to the truth: well-thought-out, simple, and well-documented processes enable your community members to do their real work easily and effectively.

Here is the crux of how we frame this perspective: processes are useful when they become a means to an end. I don't want to overplay the importance of constructing processes, but I do want to stress the importance of the activities that they implement. In other words, making it easy for new members to join your project is really important, but the process that makes this happen should never overshadow the end goal. A consciousness of processes is good; obsessing over them is bad and breeds bureaucracy. We always need to keep our eyes on the prize.

I raise this distinction for an important reason: building processes is a core activity in governing a group of people. Unfortunately, all too often the act of governing can overtake the goals of the governance. Always bear this in mind when building your processes, and always ensure you are not building processes for the sake of building processes. Not all problems can be solved with documentation and rules.

Building Great Processes

Processes are everywhere. There is a process to withdraw money from your bank account. There is a process to get the oil changed in your car. And of course, any interaction with the government...well, I am sure you can end that sentence for yourself. Unfortunately, many people who interact with processes experience little reward for a lot of frustration.

Processes are like television news: we only ever hear the negative stories. Television news favors murder and accident stories over local charity success stories and animal rescues. In the same manner, the only processes we ever hear about are often problematic ones: people queuing for two hours to mail a package, piles of forms we have to fill out to file taxes, and the hoops we have to jump through to buy insurance. Believe me, even though I have written this chapter, I hate bad processes just as much as you.

Processes that don't work suck, plain and simple. Some process failures are far more dramatic than others, though, and can have very severe human ramifications. One such example is the product recall. In *Process Improvement Essentials* (O'Reilly), James R. Persse talks up product recalls and the insights that they offer:

As a rule, corporations like to keep their troubles quiet. They prefer to keep their problems from showing up on the street. It can be embarrassing. And it takes a lot of explanation to a lot of people who probably aren't prepared to appreciate that these things happen. Nobody's perfect. Defects happen.

We are all familiar with product recalls, those little notices in supermarkets and newspapers that indicate that something went horribly wrong when the product was created. Recalls are bad for business because they say, very clearly, that the manufacturer's quality assurance processes don't work. Whether nuts contaminated a nut-free zone, pieces of glass broke into a large vat of baby food, or a batch of eggs was exposed to salmonella, product recalls point the giant finger of blame in the general direction of the manufacturing process. These recalls not only destroy faith in brands, but cause concern in customers and sometimes even panic.

Bad processes are one thing, but a process that nearly costs people their fingers is something entirely different. Persse shares with us the curious case of the Chevrolet Venture minivan debacle:

Take the Chevrolet Venture for example. This popular minivan ran into something of a quality problem in the late 1990s. At issue was the questionable design of its passenger seat latch. The company received data that indicated maybe the latch had an operational defect. After some study, Chevrolet determined that a general recall of the 1997 Venture was in order. Here is an excerpt from the text of that recall notice: "RECALL NOTICE (1997 Chevrolet Venture): Passenger's fingers could be severed by latch mechanism that moves seat fore and aft (Consumer Reports 2005)."

Now, as a general rule, severed fingers should not be a part of anyone's driving experience. What manufacturer would want that picture etched into the brand image of their minivans? Chevrolet no doubt brought its forces to bear. That latch was probably redesigned and re-manufactured against the highest standards, with inordinate attention to detail: test after test, exhausting verification runs, complete assurance of unquestionable integrity. It wouldn't surprise me if that latch today—almost 10 years later—stands as the world-class epitome of efficiency and safety in seat latch technology.

This was a pretty serious problem, to say the least. The outcome of a bad process (quality control) could have been disastrous. Aside from the obvious safety problems with many recalls, the nightmare goes much further. Products are expensive to replace. Advertising the recall is expensive too. Legal issues are likely to arise from embittered customers. Whether the bad process involves a thousand and one tax forms to fill in or a latch that may sever your fingers, bad processes cause significant harm.

Similar risk applies to communities. When we take a laissez-faire approach to processes, we put confidence in our community at risk. Processes are the conceptual buttons that your community members push to make things happen, and when those buttons don't work as expected, people get bored and frustrated and move on. On the other hand, if we craft smooth, efficient, and effective processes, our community feels nimble, responsive, and a pleasure to be part of.

Breaking Up the Puzzle

Building a quality process is like taking a road trip. You know where you want to go. You want to take the shortest route, and you want to avoid as many bottlenecks and problems as possible. When you plan your perfect route, you are careful to take the fewest number of roads, take advantage of available freeways/motorways, avoid rush hour, regularly check on current road conditions, and ensure that an In-N-Out Burger restaurant is on the route at regular points (that's just my personal criterion...).

You should take the same approach to efficiency with your processes. How can you achieve what you want as quickly and efficiently as possible? How can you communicate the journey as easily as possible to new members? How can you ensure your processes are always amenable to current conditions?

Great processes are beautiful creatures, but they need care and feeding to thrive. Our goal in this chapter is to identify these needs and produce processes that exhibit the following criteria:

The goal of the process is achieved as quickly as possible

The quicker a process ends, the quicker your community can get on with more interesting things.

The fewest possible steps can achieve the goal

Redundant steps merely make the process feel long and drawn out; let's avoid that.

Each step is simple, well documented, and clearly communicated

Each step should be absolutely necessary, and performing it should be simple. A quick technical example: if you need someone to type something in, don't demand case-sensitivity; it only complicates that step.

The process is as friction-free as possible

We want to avoid confusion and annoyance, not only with each step in the process, but in the process as a whole.

Quality is maintained at every step

We need to identify and maintain the different types of quality involved in a process: its accuracy at achieving the outcome, how efficient it is, how well documented it is, how current it is, and how open to change and improvement it is.

Although we will talk about it shortly, *simplicity* is the foundation of all good processes. As an example, consider how members join your community. How can you develop a process that is as easy as possible to follow and simple to explain to others? Put yourself in the position of a new community member who has no idea how to join your community. How can you learn how to get involved and get there in as few steps as possible? When this is as efficient as possible, you have a much better chance of recruiting new members.

Continuing the example, some communities have incredibly elaborate processes for how people join. They require applications, supporting evidence, membership council votes, signed documentation, agreement to terms and conditions, and more. Some communities are far simpler, merely requiring someone to participate in a discussion. Your community will be somewhere between these two points, and although only your community can really decide where this is, our discussion throughout this chapter should help you determine this.

Building a process

Earlier in this book we created our strategic plan. In it we produced a set of *objectives* and *goals*, each of which we broke into steps and documented. This approach is also excellent for building processes, albeit with a different set of criteria.

When you need to build a process for something (such as how members join your community), note down the following criteria:

Goal

What is the goal of the process? What does it seek to achieve? What is the outcome when the process has been followed?

Target participants

Who is the process designed for? Is it intended for a particular kind of contributor, such as a developer, documentation writer, translator, or advocate?

Requirements

What tools, knowledge, and experience must the contributor have in order to follow through with the process? If she does not possess these requirements, how can she obtain them easily? Are these requirements a barrier to entry (such as costing money or limited availability)?

Steps involved

What are the chronological steps involved in achieving the goal? What could go wrong? Is it possible for people to accidentally ruin a step? How is feedback provided about each step? Who provides the feedback?

Verification

Who makes the decision about the successful completion of the process? Also, how is it communicated that the contributor has achieved the process?

Let us now apply these criteria to our previous example of a contributor joining a community.

In our example we are the members of a small open source project that is working together on a video game that pits open source community managers against each other with the stylings of *Street Fighter II*. Naturally I would expect my own representation to be somewhat more muscular and toned.

As an open source project, we want to encourage people to contribute code (such as a patch to crank up my muscles). For developers to contribute, though, we need to give them access to the code repository. Before we give them this access, we need a process that helps them contribute some code that we can assess first, so we can ensure that only capable developers can contribute directly to the repository.

Let's now apply the set of questions we discussed earlier to this example. Here are some answers:

Goal: To assess developer contributions and if approved, permit direct access to the repository.

Audience: Developers with some experience of the project.

Requirements: Development experience, interaction with the community with regards to software development.

Steps Involved:

Developer can email patches to existing developers or attach them to bugs.

Developer creates an application page on the contributor wiki with:

Name.

Email address.

List of patches contributed and a link to their commit email.

Any additional notes.

Applicant waits for a decision within one week.

Verification: The core development team.

If you are not technically savvy, don't be thrown off by terms like "patch" or "commit email"; they're not really important. The value in this example is how it outlines the core structure behind a typical process. Also note that this process does not cover specifically how the existing development team assesses these applications from new developers: that would be its own separate process.

NOTE

Later in this chapter, we are going to dig into more detail about how to make it as simple as possible for contributors to join your community.

In a few pages we will determine what process needs you have for your community and then apply this formula to those needs. First, though, I want to talk through a few additional important best-practice considerations when thinking about processes in general.

Process Considerations

The art of building effective processes is in not forgetting the big picture when fleshing out the details. Processes get very granular very quickly. It is easy to sometimes get lost in the details and forget some of our wider ambitions.

To complement our approach to fleshing out process criteria, let us now explore some of the key themes that make good processes. These key themes should be at the forefront of our minds while we work.

Simplicity is key

Life is pretty complex, and modern technology seems to exacerbate as much as relieve that complexity. Take the now-somewhat-archaic example of the VCR. For years, thousands tried to set their VCR to achieve a seemingly simple task: to tape a show at a specific time. Unfortunately, many of us failed. We heard endless horror stories of missed tapings of *Falcon Crest*, *Days of Our Lives*, and other such travesties (sounds like a blessing in disguise to me). VCRs seemed to just confuse the heck out of many of us. There are similar examples: car entertainment systems, computer software, automated ticketing machines in parking garages, and online banking, to name a few.

Part of the problem is that design and engineering have traditionally not been particularly good bedfellows. This is not a new problem. Way back in 1847, Sir Henry Cole, one of the forefathers of industrial design, expressed his concerns over this problem to The Society of Arts:

Of high art in this country there is abundance, of mechanical industry and invention an unparalleled profusion. The thing still remaining to be done is to effect the combination of the

two, to wed high art with mechanical skill. The union of the artist and the workman, the improvement of the general taste of our artificers, and of the workmen in general; this is a task worth of The Society of Arts and directly in its path of duty.

Cole was suggesting that we approach mechanical inventions with the mind of an artist and vice versa. He understood that artists and designers have a fresh approach to how we think and interact with things. He also understood that engineers can build reliable and useful inventions. When we combine this understanding of the human psyche with the ability to build devices, machines become easier to operate and invariably look less like something that I myself would bodge together in my garage.

Examples of simplicity are everywhere. Andy Oram, who has lovingly edited this book alongside Simon St.Laurent, shared one such example of this with me. Andy had a conversation once with a user interface expert about a quiz program he had developed. His interface was good, but could have been better. It worked like this: first he had people create an account and then create and save a quiz. Andy's user interface mentor suggested he let people create the quiz right off the bat on the first screen. This meant the user could sign up for an account if she decided the quiz was worth saving. This suggested change was a simple example of optimizing a process and removing a barrier to participation. This is exactly what we want to achieve.

Avoiding bureaucracy

While writing this book, I experienced something of a disaster. While talking to my wife one evening about some Ubuntu work I was doing, I was gesticulating and inadvertently threw wine all over my laptop. Seemingly not satisfied with my quota of laptop abuse, a few days later I managed to throw tea all over the poor blighter, too. I, my friends, am a clod. Said laptop complained and refused to switch on. As such, a replacement was needed, so I hopped online and ordered myself a snazzy new machine. And thus the pain began....

In the space of the following week, I was tortured by customer support. Issues with a credit card payment, and need to use another credit card? Problem. I need to register a card and wait 24 hours. Entirely incompetent staff. Want to speak to a supervisor straightaway? Problem. I am entered into a queue, and will get a call in 24 hours. Laptop arrives and fails to switch on.... Can a replacement be ordered automatically? Problem. I have to first wait for a refund, manually reorder the laptop again, and wait for it to be built and dispatched. I was not a happy boy.

I am sure all of you can report similar experiences. Each of these frustrations was caused by bad processes. We learn from such experiences that an organization must not only create a simple process, but set up ancillary processes that kick into action when something goes wrong. The problem during my laptop order sprang from a company that was thinking like a company and not a human; in other words, it set up processes that were convenient for its organizational structure and resources, not responsive to customers.

Many of these problems could have been avoided with feedback. The most important ingredient when building processes is an understanding of people and their expectations. And this understanding requires you to solicit feedback, along with a culture devoted to always improving and refining your processes. When we understand and react to participants' expectations, processes behave as they expect. Part of the reason why feedback is so important is that it prevents bureaucracy: rules that are maintained because "*they are the rules.*"

Spread the message in your community that tomorrow may always bring a better way to carry out a current process. Processes provide an excellent opportunity to simplify tasks, tend to needs, and help your community focus on innovating more easily.

Encourage and enthuse your community to question your processes. This feedback will keep your processes on their toes and protected from the dreaded B-word.

Transparency

Another distinctive benefit of feedback is to bring an air of transparency and openness into your community. When people can contribute their thoughts and opinions, and those views get rolled back into community processes, openness is achieved.

All volunteer communities thrive on a sense of openness, because they are *associative*. These communities are built by people who choose to live their lives there. Everyone who participates in a volunteer community does so because they enjoy it: it is not a job or a requirement. As such, to keep them involved, there needs to be a sense that their input is valuable, and this absolutely applies to their input on how well community processes are working. Ask yourself this question: would you rather live in a community where you can have an impact on the governing rules, or a community in which other people decide?

Unfortunately, not all communities get this right. Back in March 2003, the xFree86 project was struggling with transparency. xFree86 was a free implementation of X, the core windowing system that many Linux, BSD, and Unix systems relied on. Everyone with a graphical interface on those systems was almost certainly using xFree86.

Back then, the processes and procedures that outlined how developers joined the project were minimal, undocumented (or unknown), and restrictive. The directors of the xFree86 Corporation dictated these processes, even though the technical direction and leadership of the project had traditionally been governed by the xFree86 Core Team (comprised of the main xFree86 developers).

One person who was deeply unhappy with this was Keith Packard, a core X developer who was working at Hewlett-Packard at the time:

While the xFree86 Board of Directors is nominally in charge of xFree86, they have absented themselves from governing the project and left that to the xFree86 Core Team. The community is left wondering who is actually in charge of xFree86. As a result, community trust in xFree86 leadership has suffered. Decisions appear to be arbitrary and are not seen to reflect the will of

the community. The leadership has no accountability to the community: thus community members have no ability to change project direction and the Board has little incentive to do so. In addition, the lack of clear formal policies has made it difficult to resolve disputes when the usual consensus breaks down.

This breakdown in governance and lack of clear, community-led processes had very real ramifications for the project. Fewer developers were able to join. xFree86 development was slower than expected. Those projects that depended on xFree86 could not progress further due to technical limitations in X, so free operating systems were failing to benefit from the features and performance offered by new generations of graphics hardware. As a critical component in the Linux desktop, xFree86 was holding everyone back. It was a real mess.

Unfortunately, the bylaws of the community stated that any changes in governance had to be initiated by the directors of the xFree86 Corporation. This was a problem. The board members were evidently not particularly responsive or reactive to the concerns of the community, as Packard noted:

xFree86 has the trappings of democracy, but the community has no voting rights and no elections are held.

Ultimately, the desire for openness and transparency won out. On January 22, 2004, the X.org Foundation was formed. The new foundation brought the open governance and open processes that are common in open source to X. Since then, the X Window System produced by the X.org community has thrived.

These risks with transparency can happen to any volunteer community. The solution to this is simple: involve your community at every step of your community growth. Involve them in the strategy, the processes, the governance, the execution of these decisions, and more. Have public communications channels and public meetings, and instead of questioning whether something should be public, question whether it should be private. When we work together, the world feels a very open place.

Assessing Needs

Earlier in this book we explored many of the best-practice aspects of community building and compiled a TODO list of these elements. One of these items rather conveniently relates to our current discussion of processes.

Community TODO List

- Build an environment conducive to our wider goals.
-

Effective environments are built on effective processes. As a community leader, your goal is to build an environment that offers rich opportunities and is simple to engage with. So you need to figure out which processes you need in your community and apply your recipe and best-practice concepts from earlier in the chapter.

Processes can be broadly divided into three primary categories:

Environment and strategy

This is a continuation of the strategy that we built earlier: the general concepts that apply to everyone in your community, such as milestones, direction, processes that define how people collaborate, and external feedback.

Infrastructure

The technical nuts and bolts and tools of your community: the day-to-day facilities that your community members require to get things done.

Governance

The governing bodies, legislation, rules of engagement, and other more political components required to organize and govern your community.

Each of these three areas is of significant importance in building a strong community, and I have devoted a chapter each to infrastructure ([Chapter 5](#)) and governance ([Chapter 8](#)) later in this book. Each of those chapters helps you build effective processes in those respective areas using the best practices described in this chapter.

So let's now talk through some of the environment-related concepts that are important in our community. This will give you some handy examples that you can apply to your own community.

Community Cycles

Most communities, particularly collaborative ones, require some kind of planning cycle. Whether you are working toward a software release, a conference, a local event, or something else, planning is key.

Many of these objectives are reoccurring. Software regularly has releases. Events regularly occur. In addition to this regularity, there are regular milestones that need to be completed to achieve the objectives.

One such example is software. Most software releases adhere to the following broad set of milestones:

1. Development begins.
2. Feature freeze.
3. User interface freeze.
4. Beta freeze.

5. Beta release.
6. Translations freeze.
7. Release candidate.
8. Final release.

Another example consists of the milestones involved in planning a local community conference:

1. Planning begins.
2. Call for papers/exhibitors opens.
3. Final speaker list chosen.
4. Final exhibitor list chosen.
5. Schedule published.
6. Venue equipment finalized.
7. Event begins.

For each of these two examples, the milestones combine in chronological order to define a cycle. If you have similar milestone-leading-to-goals requirements, I recommend that you produce your own cycle.

Leading by example: Ubuntu

Ubuntu releases a new version every six months. Each release cycle involves a number of tasks: updating the toolchain, syncing and merging with the Debian release upon which Ubuntu is built, deciding on direction at the Ubuntu Developer Summit, scheduling freezes, releasing alphas and betas, and other elements.

At the beginning of every release cycle, our release team documents the release cycle on the Ubuntu Wiki. As an example, the 9.04 Jaunty Jackalope release cycle was documented at <https://wiki.ubuntu.com/JauntyReleaseSchedule>.

The release team broke the cycle down into a number of weeks, dividing the weeks into sections by month. Each week is placed into a table that has “Week Number,” “Date,” “Status,” and “Notes” columns. Here is a snippet from the March part of the Jaunty plan.

March 2009			
Week number	Date	Status	Notes
18	March 5	User Interface Freeze Artwork Deadline Two	
19	March 12		Alpha 6
20	March 19	Beta Freeze Final Artwork Freeze	Rebuild Test
21	March 26	Beta Release Docs String Freeze	Beta

An important component that enables the planning cycle is knowing how long each task should take to complete. When you build your own cycle with your community, make a fair assessment of how long your tasks should take, and build in some redundancy for delays.

When your cycle is complete, you should ensure that it is published and available to all your community members. An excellent way of doing this is to place it on a wiki.

The Gates of Your Community

Now let's take a look at another item on our TODO list.

Community TODO List

- Attract a diverse range of contributors.
-

Attracting new contributors is an essential part of any community. Earlier we used this item as an example in fleshing out a description of our process. I want to now explore this important item in more detail.

In attracting contributors, our goal is not merely getting the word out; it is about getting the word out to the *right people* and ensuring that they can join us in achieving our goals.

In this work we have two primary outcomes that we want to achieve:

- To ensure that the on-ramp from consumer of buzz to full-fledged community member is as smooth as possible.
- To attract community members who will demonstrate commitment and a willingness to work toward the goals of the community.

The first point may seem obvious, but the second is more subtle. What we want to avoid are drive-by contributors. These are people who join a community for a few weeks, you help them get up and running, and they then get bored and move on.

Drive-by contributors are expensive, not necessarily financially, but in terms of time. Contributors often require a certain level of (wo)manpower. The community responds to their questions and provides the help, guidance, training, and mentorship needed to get them on their feet. If these contributors receive this assistance and fail to deliver anything of value, they become expensive propositions.

Animal welfare communities have dealt with this for years. Cristina Verduzco is the volunteer/outreach manager for the East Bay and Tri-Valley SPCA (<http://www.eastbayspca.org/>). Cristina is responsible for recruiting, training, and managing volunteers. She is responsible for

all facets of the volunteer program, coordination and planning of multiple events, and coordination of mobile adoptions.

Retention is important for animal welfare communities. Not only do volunteers need to be trained in how to care for the animals, but bonds are developed with the animals they care for, too. It is in the interest of the animals to maintain sustained contributions: they feel more comfortable around familiar faces. Cristina is fully aware of the biggest risk to retention:

Retaining volunteers will always be a problem for any animal welfare organization. While the work is rewarding, it can also be difficult. We are fortunate that our euthanasia rates are lower than almost all other shelters, as we don't euthanize animals for lack of space. While it is common that issues around euthanasia tend to be a big reason for people not to return, we don't see that quite as often at our shelter.

Every community is strained by push and pull forces that will affect your ability to retain volunteers. This will apply equally in your community. For the SPCA, euthanasia is clearly a blocker for many to get involved. What is your equivalent blocker? This could be difficult personalities, suboptimal working conditions, hugely complex processes and technical procedures, or anything else.

You should identify these blockers and ensure you have resources to help people over the hump. Importantly, you should not downplay or cover up the difficulties in your community, but until you can fix the problems (which you should seek to do as a matter of priority), it is entirely reasonable to make those difficulties as pleasant to navigate as possible.

You should also focus on the attracting forces that bring people to your community while again ensuring that you communicate a balanced perspective. At the SPCA, the clear attracting force was kittens and puppies, but Cristina is keen to balance this with the reality:

Let's face it—playing with kittens and puppies sounds fun until you realize there's cleaning and training involved, and sometimes that deters volunteers. Sometimes people have every intention to commit, and down the line realize their schedules don't work out.

These long-term contributors are the real rock stars in your community. Cristina and her colleagues rely on their "angels" heavily:

Fortunately, there are always those people who come in, fall in love with the work, and stay for the long haul. Some of our volunteers have been with us over ten years, and they come every week without fail. We refer to them as our "angels". Our volunteers are the reason shelter animals don't go crazy in confinement, and for that reason we are indebted to them.

You have an enormous opportunity to find your own "angels" if you offer enough enthusiasm and excitement to pique interest in your community, give them a realistic set of expectations about the work involved, and ensure their experience is positive.

Before we move on, I want to be entirely clear when I say "long-term contributors" here. You should not expect your contributors to be working full-time for your community. Your focus

should be on sustained contributions, no matter how long each contribution lasts. If your contributors put in three hours a week every week, that is an excellent gift to your community. This gift is also far better than two days of frenetic full-time contribution and then nothing else.

Retaining contributors sounds complex, but is thankfully pretty simple. If you want people to stick around, you need to offer them *a regular sense of achievement*. Your community members need to feel that (a) they are productive, (b) their contributions are appreciated, and (c) that (a) and (b) happen repeatedly.

This is particularly important for new contributors. When someone first expresses interest in joining your community, they need to follow what I call the *contributor ramp*:

- Identify an area in which they can contribute.
- Learn the skills required to contribute.
- Know which specific task to work on.
- Know how to submit their work.

When you start attracting people to your community, you are encouraging them to begin interfacing with this process. As such, you need to make sure you have all of the above in order before you even make a peep. The most logical home for this information is on your website.

Assessing Contributors

Many collaborative projects embody some kind of access control. In a typical open source project, those who can contribute code are restricted, and developers are assigned a username and password to contribute code to the repository. The reason is simple: you don't want just anyone changing the project. The integrity of the code base is essential. You always want your code to run, be as error-free as possible, and maintain consistent coding guidelines. If access were entirely open, those with malicious intentions or error-prone (in)abilities could introduce bugs, defects, and security flaws to your project. Having a vetting process is important to ensure that your developers can be trusted and will demonstrate this level of quality.

As we briefly discussed earlier in this chapter, it is important for us to figure out how new potential developers can gain access to the project. If I want to contribute to an open source project and am unknown, how do I prove my abilities?

The first task is assessing what your expectations of quality are. For many open source projects, they are relatively simple:

- A reasonable knowledge of programming; the contributor should not be making the obvious mistakes that new developers make.
- A familiarity with the code base and coding conventions.
- A regularity of contribution.

Although the implementation of these expectations involves infrastructure technology (bug trackers, source control systems, etc.), which we defer discussion of until later in this book, we can take a high-level view of how some of these processes work.

THE ELEVATOR PITCH

Before we continue, again remember that simplicity is key to good processes. One of the most effective methods of determining whether your processes are really as simple as you think they are is trying an “elevator pitch.” This is simple to test.

Imagine you are out having a drink with some friends and talking about your community. One of them asks how they get involved. To pass the test, you should be able to recite your process and have them understand it within a minute.

If the person you are speaking to stumbles on any part of the process or needs it reexplained, your process needs some work.

Reviewing new developers: In depth

Ad Hoc Peer Review is common in the open source world. Here is how it works: you have an existing member of the community review a new contributor’s proposed contribution, and if it is acceptable, the existing member will commit it on the new contributor’s behalf.

As an example, imagine Adam would like to fix a bug in the OpenOffice.org project. He grabs the code, programs a solution to the bug, and then sends an existing OpenOffice.org developer the fix. That developer (let’s call him Michael) then reviews the fix, and if it is suitable, he commits the fix to the main OpenOffice.org code base.

Typically this process would be repeated until each proposed contribution from Adam is almost certainly going to be accepted by Michael. At this point, Adam would be given direct developer access to the source code repository.

A range of projects use this approach. One such project is Jokosher, which was mentioned earlier in [Chapter 2](#). When the project first started, a new contributor called Laszlo Pandy wanted to get involved. He joined the IRC chat channel to express his interest and was asked to submit some patches (small bundles of code changes). To make this as easy as possible for new contributors, we had written a web page that explained how this worked.

I myself reviewed some of his patches. With each one, I provided him technical feedback: areas that worked, elements that needed fixing, etc. When Laszlo’s patches were of sufficient quality (which required only minor adjustments), I committed them for him. After three or four patches, every one was excellent. At this point we gave him direct access to the Jokosher

repository. As a bonus piece of Jokosher trivia, Laszlo went on to become the leader of the Jokosher project.

There are a number of requirements for this process to work:

- You need to clearly explain in what form new contributors provide a contribution. Should this be a patch? An email? A document? Explain what is required.
- Where do new contributors send their contribution? To a mailing list? Mention it on IRC? Email a specific person? Attach it to a bug report?
- Your community must be able to access these new contributions and have existing, experienced contributors provide feedback. It is important that this feedback is not held up due to having only a limited set of contributors who are able to provide it. Every contribution should be seen as a gift, and as such should rightly expect feedback, even if this feedback is just “this is excellent work, I have committed this.”
- There needs to be a method of tracking these recurring contributions. If a new contributor has contributed three or four excellent patches that have been committed with few or no changes, you need a means of identifying this recurring quality. This could be as informal as some core contributors keeping a mental note of the new contributors, or as formal as maintaining a written log of contributions.

Although this approach has been described here within the context of providing code contributions, this could equally apply to other types of contributions, such as documentation, art, and translations.

For nontechnical projects, a similar approach can involve simply reviewing prior work by the contributor. If you are organizing a local community event and you want a contributor to do the art for the brochure, you could ask him to produce some initial drafts, make sure they’re acceptable, and then accept the contributor as the brochure artist.

Unfortunately, for some larger communities (particularly large open source projects), the previous approach is unsuitable. This can be due to a number of reasons:

- The technical requirements involved for the new contributor are too complex to be assessed in a loose and ad hoc way.
- There are too many new contributors submitting content for the existing developers to assess in a timely manner.
- The project is hugely popular, touching thousands or even millions of users, and as such, granted access needs to be more rigorously assessed.

Although the core technique of reviewing proposed contributions does not change with a more formalized approach (having people check a contribution before it is committed), what does change is how you track and review those contributions.

Let's look at an example. In the Ubuntu community, we have a little more formality in how new contributors can gain access to one of our key repositories, Universe. Our guiding policy has always been that new contributors should demonstrate a *significant and sustained contribution*. That is what we seek to assess in new contributors.

The Universe repository is a large collection of applications that are packaged by volunteers. These applications are not officially supported by the Canonical technical support service, and they don't receive security updates. The team that works on these packages is called MOTUs (short for Masters of the Universe). The MOTU community is led by the MOTU Council, a small group of well-respected MOTU contributors.

Although more formalized, the process is still relatively simple:

1. New contributors make their contribution by attaching a patch to a bug report. They then subscribe the ubuntu-universe-sponsors team to the bug report. The collection of bugs subscribed to that team is called the Sponsorship Queue.
2. The team looks at each of the patches, reviews them, offers feedback, and, where suitable, uploads the changes.
3. This process repeats a number of times to demonstrate the sustainability of contributions from a given contributor.
4. After a number of contributions have been made, an existing MOTU will often recommend that the contributor apply for MOTU approval.
5. Candidates prepare a wiki page outlining the contributions they have made to the project and also some endorsement statements from existing MOTUs to approve their application.
6. The application is reviewed by the MOTU Council and then approved where appropriate.

The MOTU community's approach has been refined over the years. At its heart, new contributors add their contribution to a queue, go through review, and then apply to the MOTU Council. The only differences are the sponsorship queue and the approval process, which make the approach more formalized.

Managing Feedback

Gathering feedback from the outside world is an often-overlooked but important facet of a strong and healthy community. Feedback is important for providing you with others' perspectives, and these perspectives can help identify opportunities, problems, and areas that need renewed focus. It is this feedback that tells you what people outside your community think of your goals, ambitions, and progress. As such, feedback is a positive double-edged sword: not only does it provide pleasant confirmation of things we're doing right, but it also reveals and focuses our minds on areas to improve. Finally, a good process for handling feedback can improve all your other processes, as I explained earlier in this chapter.

Unfortunately, many community leaders view feedback as a nuisance, and disregard it if it challenges their work and the norms of the community. This is the wrong perspective to take. Feedback instead offers us an incredible opportunity for improvements in how our community functions.

None of us are perfect. None of us get things right every time. Even the greatest community leaders in the world—and the most intelligent and hard-working community members—get it wrong sometimes. But being wrong is not black and white. In many cases we craft interesting and inspiring initiatives, but we sometimes focus on the wrong things or overcomplicate matters. Feedback helps zone our minds in on which elements to fix.

This happened to me recently. Back in February 2008, my team and I were thinking of methods to improve how we handle our list of bugs in Ubuntu. As Ubuntu has grown and is now used on millions of computers around the world, the list of reported bugs has grown, too.

After fleshing out a number of different possible approaches and plans, we announced a new scheme called 5-A-Day. The idea was inspired by the popular meme of eating five portions of fruit or vegetables every day to keep healthy. I wanted to apply this concept to software bugs: encouraging everyone in the Ubuntu community to work on five bugs every day. This would make our bugs list and resultantly Ubuntu healthier, too.

The response was stunning. Many contributors got involved, and thousands of bugs received attention. Many of our contributors didn't stop at five bugs, though; they were often nailing upwards of 20 or 30 a day. Some were even hitting 100 a day. These people were bug-squashing machines.

Ten months later, at the Ubuntu Developer Summit in California, I scheduled a session to review 5-A-Day. I was keen to gather feedback and opinions about the initiative and how we could improve it. In the session, a friend of mine called Robert Collins raised a valid point: in measuring 5-A-Day, we were measuring the wrong things.

Robert argued that 5-A-Day should not be recognizing those who were smashing past the five-bug target for each day and hitting the high bug numbers. He felt that in keeping with the Ubuntu ethos of *significant and sustained contributions*, 5-A-Day should instead cap the number of bugs at five per day and instead track *how long* people were maintaining their daily 5-A-Day. Robert felt that it was more valuable if we knew that a contributor had consistently worked on five bugs a day for a month, instead of doing 155 bugs in a day. He was right. Sustainability is an essential component in community: more bugs get fixed, people consistently set a good example for others, and we build a long-term contributor base that is regularly working on bugs as opposed to localized spikes of interest.

In this example, our request for feedback at the Ubuntu Developer Summit resulted in a valuable piece of insight, one that may never have occurred to my immediate colleagues or to me.

Gathering feedback

Feedback can be collected in many ways. The simplest and often most effective way is to set up an email address that forwards email to a number of core community members. A simplification of this approach is to set up an account with a free email service and give each of your core members access to the account. This approach is excellent for gathering general thoughts, concerns, and opinions about the community that often reflect on processes.

Another alternative is to set up a public mailing list that people can submit feedback to. This solution is not the most suitable for single-shot chunks of feedback from someone. Mailing lists are designed for discussion, so you should really use mailing lists only if you want people to submit feedback and have a discussion afterward. Another downside of mailing lists is that they require people to sign up, and these members will also receive all email to the list. Someone who just wants to give you a quick chunk of feedback likely won't want to see what everyone else is submitting to the list.

Another possibility is to use a blog. You could post entries to the blog to request feedback and allow readers to provide their feedback in the comments. Remember that all feedback here will be public. You may not want to have everyone and their dog see this feedback (although it is an impressive message of transparency if you do). What's more, remember that search engines will index this potentially negative feedback, showing it to everyone who searches on your project name, even if you've since addressed the problems.

The most productive approaches I've used for gathering feedback have been surveys and one-on-one discussions. We cover surveys in more detail in [Chapter 7](#); they are an excellent means of gathering focused feedback around specific areas of interest. I have used them extensively in my work, particularly online surveys, and they have always gathered excellent results.

Whichever of these techniques you decide to use, you should always augment them with some direct one-on-one discussions. Find people who have an opinion about the process you are investigating, and get them on the phone, send them some specific questions over email, or get together to discuss their views over a coffee. This direct feedback is often very valuable; just remember that feedback will always be tinged with opinion and potential exaggeration, and one-on-one discussions are the most likely candidate for this.

PUTTING CRITICISM IN CONTEXT

One final note about feedback is related to expectations. You should prepare yourself to receive more criticism than praise. This is simply the nature of community and human interaction: we often feel compelled to write an email to criticize, but we rarely get an urge to send email to praise. As such, if you receive what appears to be a large amount of criticism, it may not be reflective of your community. It is likely that there are a lot of silent yet happy members out there.

Getting Buy-In for Your Processes

So far we have discussed many of the best practices for building simple yet effective processes. We have talked about simplicity, transparency, avoiding bureaucracy, assessing needs, and more. Processes don't mean anything, though, unless your community (a) knows about them, (b) understands them, and (c) uses them.

At the beginning of this chapter, we likened processes to machines and waxed lyrical about how interface and industrial design have made devices easier. We noted that we should apply the same logic to our processes. We can continue our analogy beyond the development of processes (akin to the devices) and also apply it to their packaging and distribution. How can we make our processes as easy to understand, discover, and use as possible?

Document Them

The first step in making a process work for your community is to ensure it is documented. The goal here is efficiency. Sure, anyone can write a detailed list of steps outlining how a process works, but who wants to read paragraphs and paragraphs of text? The documentation behind a process should be as close as possible to a cooking recipe: do this...do that...get this result. The emphasis here is on quick, clear, straight-to-the-point directions.

Processes are fundamentally a collection of steps with an outcome. Documenting them is similar to Ikea's instructions for putting together a shelving unit. How can you tell the user how to achieve the outcome as easily and effectively as possible?

When Ikea does this, it uses only pictures, presumably to avoid translating instructions into multiple languages. Although I would recommend using text for most communities' processes, the Ikea model is cathartic and helps focus your mind on the right goal of reducing clutter.

To get you started, let's get introspective for a while and document your own steps for writing a process:

1. First, write down the end goal of your process. What does it achieve?
2. Now, in numerical and chronological order, write down each step in your process, using a single word to describe each step.
3. Finally, for each word, write a single sentence that clearly explains what is involved in the step.

As an example, the following could be used to develop a process for a new developer contributing a patch to a project:

GOAL: Achieving contributor access in the FooBar project.

PRODUCE: Produce a patch with a new feature or bug fix that can apply to the latest development version of the code.

SUBMIT: Send the patch to the FooBar project mailing list outlining what it does.

WAIT: Wait for feedback on the patch and, if required, make adjustments. Otherwise, the patch will be submitted.

REPEAT: Repeat the above steps until a developer considers your contributions consistent enough to offer you direct access to the repository.

When following through with this approach, always read and reread each step, and assess how easy it is to understand. Is it written concisely? Does it use too much jargon? Would it be suitable as an elevator pitch?

Make Them Easy to Find

Processes don't have any value when no one knows that they exist. In addition to ensuring that your processes are clearly written, you should work hard to ensure that they are discoverable. Our goal here is to ensure that community members can find our documented processes easily.

This is a two-step approach:

1. You need to put your documentation somewhere online that people can refer to.
2. You need to inform your community about additions and changes to the documentation when they happen.

The first step involves putting your process somewhere accessible. If it makes sense, you should put it with the rest of the strategic documentation that we discussed earlier in the book. Discoverability is the key here, though.

As an example, if you have a process that new contributors should use to gain repository access, you should make sure that new contributors can find it. This kind of documentation should not be buried in a wiki somewhere. You should assess where your potential contributors are reading documentation and where they are likely to look to find your processes. Again, plenty of feedback can ensure you make the best decision here.

LINKING FOREVER

When putting process documentation online, you should ensure that its location never changes.

The link to the location will be referenced extensively, particularly in online communities.

If you need to move your documentation, you should ensure that the old link redirects to the new documentation. The referenceable integrity of your documentation is something you should always endeavor to maintain, even if your documentation moves.

The next step is announcing the new process documentation to your community. Earlier in this book, we discussed communication channels within your community, and you should use

these channels to announce this documentation and encourage your community to make use of it.

When you announce your processes, you should include the following details:

- The problem that you sought to solve.
- A single paragraph overview of the process and how it works.
- A link to the online documentation that explains how the process works.
- A final paragraph that strongly encourages the community to make use of the process.

When announcing new processes, you should tread carefully. You should expect that your members, suffering from the same fear of rigidity and rules I mentioned at the beginning of this chapter, will roll their eyes at the concept of a new process being announced. Your goal is to have them entirely on board by the end of the announcement. Use loose and friendly language, and make them understand that the process is really necessary and not merely an exercise in bureaucracy.

Using Your Processes

With the process documented and announced, the final step is to encourage your community to make use of it. Documentation and announcements are no guarantee your community will make use of a process. In my experience, every process needs a certain amount of manual pushing and poking to become the norm.

Communities generally follow by example: members look toward other people to engage with processes before they do it themselves. You need to put a few examples of successful use of each process in place as a head start to get the community to accept the processes.

A useful approach is to pick four or five key community members and ensure that they are fully behind the process that you have announced. You should regularly check in with these members to ensure that they are making use of the process, and when they are not, you should check why and remind them where needed. You should also encourage these key members to spread their best practices throughout the community.

You should also identify incidents that act as opportunities to reaffirm the purpose of the process. This could be handled in two ways. On one hand, you should find success stories: examples that used the process with very positive results. These examples are always fantastic to show off. On the other hand, when someone doesn't follow the process and things go wrong, you can use it as a chance to remind your community about the purpose of the process. Do tread this line with caution: you should absolutely *not* show up your community members in front of others, and you should try not to climb up on your high horse and send out an "I told you so!" message.

Process Reassessment

Processes are living, breathing organisms. They are typically based around current conditions: the current level of contribution, demand, expectations, and goals. They are the clear plastic film that wraps around the assets and members of your community. As the assets and members adjust, so should the processes.

At the start of this chapter, we firmly established that unchanging processes cause bureaucracy. Processes that no longer map effectively to the people who need to use them cause bureaucracy. Our whirlwind tour of process best practice would not be complete if we didn't discuss how we can best evaluate our processes and make changes where appropriate, thus avoiding the dreaded claw of bureaucracy.

Process reassessment has become a staple part of each Ubuntu release. The Ubuntu community has a huge range of processes, initiatives, governance structures, and more. Each of these facilities was developed to serve a specific purpose, but as the community has grown and changed, these purposes and processes have needed to change, too.

An example of this was a change in how Ubuntu members have traditionally been approved. Anyone can be a member under the premise that they have demonstrated a significant and sustained contribution to open source. Traditionally, the way this assessment was made was that the candidate would first produce a wiki page outlining these contributions. The next step would be to attend a Community Council meeting online in which the council would discuss the application and vote. A majority vote would approve the member.

As the Ubuntu community continued to grow, this process became increasingly overstretched. The vast majority of Community Council meetings were devoted to approving members, and meetings were dragging on for nearly three hours. At an Ubuntu Developer Summit we identified the problem and proposed that we set up a series of regional subcouncils. We planned to set up a council in the Americas, Europe, and Australasia, and these would take over the task of approving members.

The new councils were formed, and subsequent membership applications have been uniformly more efficient. This has not only benefited prospective members, but it has meant that Community Council meetings have been more focused on governance issues as opposed to merely approving members.

These kinds of process reviews used to occur on an ad hoc basis in the Ubuntu world, but we have since tied them to regular timeline. We now reassess all processes at the beginning of a cycle, and many of these reassessments occur at our biannual Ubuntu Developer Summit. This provides an excellent opportunity to gather feedback (of which we identified the importance earlier in this chapter) and discuss better alternatives.

NOTE

The problem of overwork and delegating that work to other bodies in your community is part of governance, and we discuss this in detail in [Chapter 8](#), cunningly named... *Governance*.

Building Regularity

With your community, you should schedule regular process reassessments. You should schedule a time in which your community can come together to determine how to improve on these underlying structures and processes.

How you do this is really dependent on your community. For a small local community, why not organize a series of meetings in a coffee shop or at someone's house? For an online community, a series of scheduled sessions on an online chat network such as IRC is often effective. Alternatively, you could organize a more formalized event for your community, akin to what we do with our Ubuntu Developer Summit.

However you choose to organize this reassessment, you should ensure the following:

- The events should be accessible to your community members. They should preferably incur as little cost as possible, and be within reasonable reach. Organizing a reassessment in Jamaica when your community is based in Northern England is not practical.
- The events should be open to all of your community members, and you should explicitly state this when promoting them. You should clearly communicate that everyone is welcome to join in and provide feedback about how to improve how the community runs.
- Ensure a sensible level of representation. Feedback sessions with 2 or 200 people are not valuable. Sessions with 10–30 people offer real opportunity to achieve some conclusions.
- Ensure that you begin organizing these sessions with plenty of notice, particularly for physical meetings.

When advertising these sessions, you should make them as attractive and practical as possible for your members. Don't describe it as a "governance and process review" but instead as "how to improve our community." Make sure that your announcement welcomes everyone, and ensure it underlines how everyone can have an impact. Here is an example:

Improving Our FooBar Community

1st–3rd Sep 2009 – 5pm–7pm UTC – #foobar on Freenode IRC

Ever since the beginning of the FooBar project, we have all worked hard to produce a rock-solid implementation of Bar, and we have seen many new contributors join our stunning community.

This range of sessions is designed for us to come together, share experiences and feedback on our processes and methods of working in our community, and see how we can make improvements that benefit everyone. Everyone's participation is not only welcome, but encouraged!

When you have scheduled these reassessment sessions, you can build an action plan of what to discuss and how you discuss it. In many of these sessions there is a potential for the discussion to wander off at a tangent. It is therefore important to have a firm idea of what to discuss and what the core issues are.

I recommend following these steps to carve out a method of handling these sessions:

1. First, make a list of all of the processes that are involved in your community. This should cover topics such as how people get involved, how you work together, how teams work, etc. I recommend you do this in an online document, such as a wiki page.
2. For each process, note down bullet points for the feedback that you have heard, both good and bad.
3. Now look at the full list and reorganize it into what you consider priority order. This is most typically driven by which processes are causing bottlenecks and problems in your community.
4. Now inform the wider community of the list and ask people to either edit the list directly (if they can, such as with a wiki) or to submit feedback for you to add to the list. You should ask for feedback about the processes and also which processes they consider the most important to discuss. Give them a deadline for feedback that is before the reassessment sessions. Merge this feedback into the document and reprioritize the list based upon consensus.

This list is your agenda for the meetings. When you get together with the community, pick the top processes and discuss them in turn. Communicate the good and bad feedback with the community, and focus the discussion on coming up with modifications to the processes that may be an improvement. Always keep these discussions focused on achieving outcomes; they can quickly degenerate into fruitless conversations.

When you reach consensus on changes to these processes, you should ensure that the changes are noted down in the session. These notes provide a TODO list of changes to your process documentation. Remember that each time a process changes, you should announce it to the wider community, just as you did with new processes.

Moving On

At this point, we are making some real progress on our journey through community. We have a strategy and a direction, and we have identified how to build strong processes for many aspects of our community. Now it is time for us to get the tools of collaboration in our community's hands. We need to put the foundations in the ground that our community can use to build great things. It is time to talk infrastructure....

Supporting Workflow with Tools

“We shall neither fail nor falter, we shall not weaken or tire...give us the tools and we will finish the job.”

—Winston Churchill

WHEN I WAS 16, I KNEW I WANTED TO PLAY WITH COMPUTERS. To access this world of exploration, though, I needed a bundle of what I had precious little of: *money*. One option was to save, but like many 16-year-olds I was more likely to fly to the moon on a potato than save any lucre.

It wasn't just hardware at the mercy of money; so was the Internet. Back in those dim, distant days, dial-up Internet access in England was charged at 10 pence a minute. As a blossoming Net junkie, I had a (seemingly reasonable) proposal for my parents. For each minute of Internet access, I would put the requisite 10 pence in a cardboard box next to the computer. At the end of the month when the phone bill came in, the money in the box could cover my usage. Simple. Well, at the end of the month a bill for £190 arrived, and there was six quid in the box. The philosophy of “it's easier to ask for forgiveness than permission” didn't wash with my parents.

Over in the U.S., things were different. Many parts of the country had flat-rate local access and established Internet access at universities. The growth of these cheap networks made it easy for online communities to form. Early on, these communities shared knowledge as text files, packed with ideas, techniques, and recipes to make things. Each text file was shared freely, and people could add new content, modifications, and changes. Eventually, it seemed like

information covering every conceivable subject was available online. I myself contributed, writing guitar lessons and putting them on my website.

This global body of knowledge was in many ways the great grandfather of Wikipedia. The Internet not only enabled the sharing of information, but enabled the sharing of tools to create this content. A new era of collaboration was bubbling to the surface. As that collaboration became more widespread, however, new challenges arose. While the cost of computing and the Internet declined, coordinating people across the Internet proved more difficult than just opening the doors.

Understanding Your Workflow

Tools are a means to an end. They are used to produce something more interesting than the raw material and tools used to manipulate it. To select the right tools for the job, we need first to understand what we are trying to achieve. We need to know what our *workflow* is.

I am a firm believer that every brain is different. Each interprets the world in a slightly different way, and each defines our differences and individuality. Take music as an example. For some, the purest definition of beautiful music is blues, for some it is rock, and for some it is crushing death metal. Each style has the same instruments and core ingredients, but we all perceive those styles in very different ways.

The same happens, for example, with programming languages. My brain just works with Python. When I started to learn Python, it felt natural. I was productive almost immediately. Some languages, on the other hand (such as Perl), just confuse the heck out of me. No matter how much I try, the square Perl block just doesn't fit into the circular hole in my head. Programming languages are curious vessels of logic and workflow. Each one has a set of semantics, rules, and procedures that must be learned and executed with perfect precision for it to work. In addition to this, different programming languages have different technical approaches, cultures, and expectations. Each of these attributes must be learned and mastered for the programming language to be a useful tool in helping you to create something. The reason why Python works with my brain is because the workflow and semantics of Python help me to be productive quickly.

These differences are everywhere, and become more important as people work together. It's not just programming languages, though. We see differences in how we keep TODO lists, how we manage email, how we organize our important documents. Our methods of working and organization styles vary. A successful workflow requires understanding what you want to achieve and making the steps in which you achieve it as simple as possible for as many people as possible.

One of the reasons why the Firefox browser has been so popular has been its extensibility. A large and growing collection of Firefox add-ons is available. These small bundles of functionality, basically small pieces of software, integrate tightly into the browser and can do

anything from block ads to allow web developers to dynamically adjust the layout of web pages. Traditionally, installing software has been possible using a variety of approaches: installers, package archives, compiling code, and more. The Firefox team made it as simple as selecting an add-on and clicking Install. Firefox takes care of the rest. Workflow and usability are close companions.

Another successful workflow example comes from Bytemark Hosting (<http://www.bytemark.co.uk/>). This small hosting company provides virtual hosting machines that run Linux. If my machine was unresponsive for some reason, they offered a command-line-driven administration console that I could log into. That console let me restart the machine and gather diagnostic information. Although it might have seemed complex and technical to the average Firefox user, the console was fast, efficient, and intuitive for most Linux users. If a machine was down, I could be into the console within minutes and have it back up again, all from my familiar command-line interface.

The connection between both of these examples is the first task in understanding workflow: know your audience.

Roles

People always have expectations. Our job is to understand, predict, and cater to fair expectations in our target audience. To understand these expectations, we need to understand our audience. Although communities are a breeding ground for diversity of personality, experience, and background, we can often see similarities in expectations, skills, experience, and approaches between people who have a shared interest in a particular type of work, be it programming, documentation, testing, advocacy, or whatever else.

Consider programmers as an example. We know that we can assume a certain amount of technical knowledge: programmers indulge in a technical art form. They spend their days engaging in technical conversations and manipulating technology to their needs. You can't push these assumptions too far, though. A Windows power user and a low-level assembly device driver writer are both "technically trained," but in different ways. Technical experience comes in many forms. It is important for us to see the correlations, but also to keep track of subcategories.

Roles are critical in identifying preconceptions and experience. Using programmers again as an example, it is reasonable to assume that a programmer will know how to use an operating system reasonably well, know much of the jargon associated with computers, and be fairly self-sufficient in solving technical problems. Each of these attributes is common in programmers. We draw these parallels from two primary methods: *observation* and *experience*.

Understanding the audience requires observing them in their natural environment. If you want to make bug triage as simple as possible, sit down and watch someone triage a number of bugs. In fact, watch a range of people triage bugs. If you want to understand how to run a booth at

a conference efficiently, watch how people set up booths and how they talk to people. As you observe your community engaging in their workflow, ask them to comment on it, telling you what frustrates and annoys them and what works well. This feedback is always valuable.

Identify the parallels between the multiple people who perform the same type of task (such as programming), and you begin to build a mental picture of their expectations and experience.

Your own experience will also guide how you build strong community and identify with these different roles. Repeated experiences in particular can be critical for understanding what is going on.

I had a colleague who did sales to big companies. On his first big deal, he went through an intense set of emotions. He first built confidence while the client was excited about the product. As the negotiations moved forward, he was cautious but confident. All was going well until he hit a roadblock in the negotiations. It was an emotionally difficult time: he was about to close a big deal, he was excited but stressed, and he was losing sleep and was constantly anxious to check email. What my colleague didn't know at the time was that this was part of practically every deal: every deal hit a roadblock and every deal had moments of doubt. After a few more deals his experience taught him what was the norm in his world. From then on he knew how to manage his expectations. Your own experience will uncover similar secrets about the roles that you are seeking to understand.

Importantly, your community members can help you to understand these roles too, but you really need to figure out their current workflow and ask them for specific feedback about its different parts. Have a series of frank and honest conversations about the overall workflow. This provides community members with an opportunity to share how the workflow benefits their lives and how it complicates them. You should also get to know their general perspectives and opinions that surround the workflow. As an example, if you are talking to a programmer, ask for her opinions on related topics such as testing, bugs and triage, translations, code commenting, documentation, and other topics; this will help you build a better sense of her views and expectations. While having these conversations, take copious notes: the devil is almost always in the details, and you don't want to lose anything.

Building a Simple Workflow

After establishing roles, we now need to sit down and flesh out a workflow. In this chapter our focus is on technical, tool-based workflow, so we will assess the technical steps involved in achieving a goal. To do this we will identify an "OBJECTIVE," the "AUDIENCE," and then a number of "OUTCOME" steps that can achieve the objective. An example will best illustrate how that works.

When I was involved in Jokosher, we wanted people to test the application and provide feedback. We wanted to take a snapshot of the application for members of our community to test and tell us when things went wrong. This would give us an opportunity to fix remaining

bugs. That helps us produce the first item in our workflow specification, the objective that you want to achieve:

OBJECTIVE: To have more users install and test a development snapshot of Jokosher and provide feedback on bugs.

The next step is to identify the audience. We did not want developers to test Jokosher; we wanted real users to test it. We wanted people to use Jokosher for real music production and make assumptions based on real use cases.

THE RISKS OF AUTOPILOT

A common problem that can occur when observing how people use software is when the user knows of a particular quirk in a product and works to naturally avoid triggering the quirk. This is common with software developers, and before release, the software typically is not used in the same manner as it is by normal users after release.

A good antidote to this problem is to simply put new users in front of the software who are entirely unaware of the quirks and oddities. They can often present the most valuable feedback.

Let's now add our audience to our spec:

AUDIENCE: Musicians or audiophiles who have basic computer knowledge.

The next step is in describing the primary pieces in the workflow. At this point we're just identifying the major steps; we'll add the details later. Add these as a series of **OUTCOME** items. For example:

OUTCOME: Install a current snapshot of Jokosher and all required dependencies.

OUTCOME: Use and test primary features in Jokosher.

OUTCOME: Provide feedback about bugs.

These three outcomes are the major steps involved in achieving our **OBJECTIVE**. The next step is to combine our **AUDIENCE** with each **OUTCOME**.

This is the most important part of the process, where we assess how to make our workflow as simple and attuned to our audience as possible. Let's look at our first **OUTCOME** in the preceding list as an example. We need to combine this with our audience of new users. How can we make it as easy as possible to install Jokosher for a user base that is not technically sophisticated?

When we were faced with this challenge, we considered a range of options. We heard some suggestions to provide simple documentation that would show people how to compile code and install dependencies. That was too technical and complex as an option. There were also suggestions to make packages available for each distribution. That required skills that were not present in the team, and would involve sourcing external help.

The ideal scenario was that someone could simply download a file and Jokosher would run: this matched our audience profile. It would be simple, easy, and efficient, perfect for our users. Stuart Langridge produced a script that downloaded the Jokosher code, installed the relevant dependencies on the user's system, and ran Jokosher with a click of the mouse. The script checked to see whether the dependencies were already installed, and skipped their installation if they were. The user could use the same script for the first run and subsequent runs, and not even think about the concept of installation. Simple.

In the workflow spec, you should document each step from the perspective of your AUDIENCE. For the Jokosher installation example:

OUTCOME: Install a current snapshot of Jokosher and all required dependencies.

Download a script from jokosher.org.

Make the script executable (explained with documentation).

Double-click the icon to run the script and install Jokosher.

To re-run, double-click the icon again.

This process was as simple as we could make it for the user. The hardest part was making the script executable, and there is no safe technical solution to automate that process. Our solution also required few resources to develop, just the time for one person to produce the script instead of lots of packagers to package Jokosher for every conceivable Linux distribution.

Once you have written your workflow steps into your functional spec, you should now go through each step and ask the following questions of it:

- Is this step really required?
- How easy is this step to understand? Could it be simplified?
- Are the requirements easy to obtain or access?

When you have applied these questions to each step, you can now apply these questions to the entire workflow:

- Is this workflow as efficient as it could be?
- Is this the most intuitive approach to the workflow?
- Is this workflow scalable?

If you have satisfied each of these questions, you should have a pretty rock-solid workflow in place.

The Mechanics of Collaboration

Earlier in the book we added a critically important item to our Community TODO List.

Community TODO List

- **Build an environment conducive to our wider goals.**
-

The operative word here is *conducive*. What does that mean, though? How exactly do we optimize our environment for the purpose of achieving our goals?

Collaboration is all about working together for a common purpose, opening up a channel in which content can flow between interested parties for the purpose of building something interesting. This content and the tools that make it flow are the *mechanics of collaboration*.

Collaboration is a form of conversation, a reciprocal back-and-forth of communication around a common topic. At the heart of all conversations are at least two people and a:

- Shared language.
- Shared topic.
- Communications channel.

Imagine a normal spoken debate taking place in a London pub. The shared language may be English, the topic could be how effective the prime minister is, and the communications channel is the spoken word (lubricated with a pint of something cold). These simple attributes combine to create a thread of responses, each building on the previous statement. This combination of responses comprises a conversation.

An Example: Ubuntu Bug Workflow

Technical collaboration has similar components and similar results. The mechanics of collaboration are so important because when you understand what conversations occur, you can optimize how people converse. Take bugs, for example. Every piece of software ever written has bugs. Software is written by people, people make mistakes, and bugs are human mistakes formed as bits and bytes.

Most software projects use a special piece of software called a Bug Tracker to have a bug conversation. Someone files a bug. Someone else asks for more information on the bug. Those who are affected by the bug offer clues and information. Someone else may propose a fix. People try the fix and provide feedback. The essence of conversation is the same: a shared language (English), a shared topic (exploring and fixing the bug), and a communications channel (the Web).

When considering how to build your community’s infrastructure and tools, you need to identify what these mechanics of collaboration are. You need to think carefully about how people have conversations, both in natural language (such as discussions) and technical (such as with bugs). When you understand the driving forces behind how conversations occur, you can make some really interesting things happen.

A while back my team spent some time working on optimizing the conversation around bugs. Before I explain the solution, let’s take the problem for a spin.

In the open source world bugs are a complex topic, and they are particularly complicated for Linux distributions. Consider the problem. Erica is using Ubuntu and experiences a problem in the Text Editor: when she tries to spellcheck, the editor crashes.

As an experienced open source citizen, she is familiar with the mantra that when something goes wrong, you should report it. She clicks the Help menu in the text editor and clicks on “Report a problem...” to report the bug. At this point she now expects someone in Ubuntu to tend to the bug report and hopefully fix it. So far, so good.

Now it gets a little more complicated. The Text Editor in Ubuntu is actually called GEdit. It is an open source project that is part of the GNOME (<http://www.gnome.org/>) desktop. The GNOME project is an independent project that produces software, and the open source parlance that describes this kind of project is an *upstream* project.

Ubuntu developers take the upstream GNOME source code (including GEdit) and build packages to ship it in Ubuntu. When this code is built, some modifications are made to make GNOME (and as such, GEdit) integrate well in Ubuntu. This includes consistent themes, file dialog boxes, folder locations, and more. In a nutshell, Ubuntu takes the original upstream source code, adds some changes, builds it, and ships the final product as part of Ubuntu.

This raises our first question: who is responsible for the bug? Was the bug present in the original upstream code from GNOME? Was it in the changes made by Ubuntu developers? Does the bug exist outside GNOME and GEdit? Does the bug even exist at all? Fortunately, this is a fairly simple question to answer: some quick checking of the upstream code can usually identify the source of the bug.

The second question is the one that is pertinent to our bug conversation. How do we deal with the bug report? Erica rightly reported the bug, but should she report it in the Ubuntu bug tracker (called Launchpad) or the GNOME bug tracker? In addition to this, many other people may have experienced the bug. Some will report it in their distribution’s bug tracker and some will report it in the upstream bug tracker. It is not entirely inconceivable that there is a bug report in each distribution’s bug tracker and one in the upstream bug tracker all pointing to the same bug. This is hugely wasteful.

Regardless of all this duplication of effort, if the bug exists upstream and the distribution ships it, the bug *does* still exist in the distribution and it *does* exist upstream. The challenge is to bring each of these different bug conversations together so everyone who experiences the bug can

share their experience to fix it. The solution to this problem is to identify how the flow of conversation and collaboration happens between these different bug reports and to optimize it.

Fortunately, software can address these issues. Launchpad, the web service that Ubuntu uses as a bug tracker, already had the ability to link bugs, whether they're in the same tracker or a different one. When Erica filed her bug, she or someone else could connect her bug to an existing bug in an upstream bug tracker. It works like this:

1. If the developer knows the bug is an upstream bug but does not know which bug it is in the upstream bug tracker, he can add an *upstream task* to the bug report. This upstream task indicates that the bug should be linked to an upstream bug when it is found.
2. If the developer knows the bug in the upstream bug tracker matches the Ubuntu bug, he can link the two bugs. This involves finding the Ubuntu bug and using the Link feature in Launchpad to enter the URL of the upstream bug.

When a bug is linked, any changes made to the upstream bug are synchronized in Launchpad and vice versa. This feature connects the two separate conversations.

Getting to know the problem

Although this is a hugely useful and valuable feature, we got the impression that not enough bugs were getting linked. The problem that I wanted my team to solve was to explore why these bugs were not being linked and help to increase the linkages.

Although we had a hunch that not that many people were linking bugs, we really had no idea. We had no concrete statistics to back up our assumptions. Our first task was to learn more about the problem.

To do this we produced a tool called the Ubuntu Upstream Report (<https://launchpad.net/ubuntu/+upstreamreport>). The report mined the Launchpad bug tracker to show the top 100 upstreams shipped in Ubuntu ordered by the greatest number of open bugs. We knew that these 100 upstreams were likely to be the largest and most significant projects: more bugs in the open source world typically means larger projects with more users using the software and therefore filing more bugs.

For each upstream in the report, we showed which bugs were open, which had been triaged to determine they were actually bugs, which were known to be upstream bugs (marked as upstream tasks), and which had linkages associated with those tasks.

The differences between the numbers generated interesting conclusions. As an example, if a project had a significant difference between the number of *open* bugs and the number of *triaged* bugs, we knew which projects needed more help with bug triage.

A key conclusion that we discovered was the difference in many projects between the *upstream tasks* and *linked bugs* numbers. As an example, one project had 229 bugs with upstream tasks but only 101 of those bugs had upstream links. This left a total of 128 bugs that were known to be upstream bugs that didn't have linkages. This told us that the project was not linking bugs

well and needed help. It also told us which specific bugs were known to be upstream bugs without linkages. We could use these bugs as a target for our community to make those linkages.

Breaking down the conversation

The raw data in the upstream report was not only the foundation of exploring the problem, but also solving it. The data confirmed our suspicion that a lot of bugs weren't being linked. On a practical level, it helped us to understand which upstreams needed help, and therefore which contributors to speak to.

To raise the number of linked bugs, we needed to identify the three key aspects of the bug-handling process (reporting, triage, and linking) and improve them. Our first area to focus on was linking, as it was the most immediate goal. We examined and optimized each step in the process of linking a bug. We discovered, for example, that finding bugs in the upstream bug tracker took the most time, so we explored methods of syncing bug data across the Launchpad and upstream bug reports automatically, automating duplication searches, and simplifying the user interface for the whole shebang.

Our next step was to optimize the reporting and triage process. This was an area in which we had already poured extensive work, and our workflow here was generally pretty good. Where there appeared to be a disconnect, though, was with documentation and people reading and making use of that documented best practice. Although there was plenty of documentation, it was scattered all over the Ubuntu wiki, difficult to navigate, and complex to read. We reorganized and improved it, dividing it into better sections and making it easier to understand. We organized documentation days; encouraged community participation; celebrated the improvements; and encouraged our bug triagers to review, expand, and make use of the new documentation that was produced.

Lessons learned

This example reflects a useful approach to help you “build an environment conducive to our wider goals.” Its key themes likely apply to your community:

1. Identify the primary ways in which you collaborate. These are the areas in which your tools and workflow should be as flawless and intuitive as possible. In our example, we identified bugs as a key area.
2. Understand the problem. How does your community understand and engage in the process you are exploring? Which parts of the problem are more problematic than others? What do you want to achieve? In our example, we knew that linking was a problem, and focused on it.
3. Break down the workflow. When you understand your areas of focus, examine the process and ask whether each step is entirely necessary or could be improved.

These three simple steps are all based on the principles of understanding the problem and making a solution that is as simple as possible. *Simplicity* is a key goal in everything we do with community.

Building Great Infrastructure

Consider the humble can opener. Years ago, to use a can opener you had to place a very small cutting blade on the lip of the can, ensure it hooked over correctly, and then twist the handle, invariably making your fingers ache. As you twisted the handle, the blade often slipped off the can. Even when you had managed to get the lid off the can, you might cut your finger on the lid or the sharp top of the can. We now have electric can openers in which you simply press a button and the lid is cut and taken off for you. Even manual can openers are effortless and go as far as to blunt the lid and the top of the can. Manufacturers were well aware of the problems and explored the workflow and potential solutions to evolve their tools.

Collaborative online projects need a diverse set of supporting tools to flourish. Even the simplest community will use tools for communication, storing work, and sharing information. Many communities build on top of these staples with a variety of tools and facilities for different functions in the community. Different contributors have different needs for their types of contribution. Developers need bug trackers, patch systems, and version control; documentation writers need wikis; text processing tools, and editors; translators need translation tools; testers need test suites; and everyone needs to communicate with each other.

It's tempting to go out and set up a wide range of tools, but think carefully about how your tools integrate. An efficient and integrated set of tools will be far more pleasurable to work with than a completely disconnected set of tools.

For years I have been playing music, and writing and recording my own songs in my home studio. As a strong believer in a Free Culture music industry, Severed Fifth (<http://www.severedfifth.com>) was my contribution. I wrote, produced, and recorded an entire solo metal album and released it under a Creative Commons Attribution-Share Alike license. I set out to build a community around the project and the music, building many features into the site: news, a blog, forums, static web pages, and more. I wanted the entire site to feel as integrated as possible. My friend Stuart implemented a single user account that worked with all features on the site. Although the site consisted of two WordPress blogs, a Vanilla forum, and some static web pages, having an integrated account and look and feel made all of the separate components feel like a single well-oiled machine.

Integration offers a range of opportunities for collaborative projects. Workflow is the foundation for identifying these avenues of integration. Workflow is a larger process that can contain many individual components.

Take, for example, the workflow to fix a bug. A simplified process could involve finding a bug in the bug tracker, checking out the code, writing a patch to fix it, attaching the patch to the

bug report, and informing the developers of the fix. This workflow involves three separate tools: the bug tracker, the source repository, and the mailing list. It could arguably involve additional tools, such as those required to produce the fix and generate the patch. There are many areas in which the tools could be integrated:

- The bug tracker and source control system could use the same login credentials.
- The bug tracker could display recent branches in the source control system that refer to a specific bug number in the changelog.
- When a patch is attached to a bug report or a bug status changes, that could generate an automatic email to the project mailing list to inform the community that a fix has been made available.
- When a fix is sent to the mailing list with a bug number, additional information from the bug tracker could be automatically added to the message.

Every project has the opportunity to draw similar integration opportunities between the different tools in the workflow. The key is in identifying how different tools can talk to each other in useful ways.

The first time I experienced this firsthand was with Jokosher. When the project started, we needed the following facilities:

- Source control.
- Online commit log history.
- Source viewing.
- Bug tracking.
- Wiki.

One approach was to set up a number of separate tools to provide these facilities. We could have set up Bazaar, WebCVS, Bugzilla, MediaWiki, and other tools. Each of these tools would be largely insulated from the others, and each with separate accounts. Our final solution was to use a system called Trac (<http://trac.edgewall.org/>).

Trac is a popular project management and software development system that provides an easy-to-use and integrated system for source control, issue tracking, commit logs, and a wiki. We used a hosting company called Python-Hosting (now <http://www.webfaction.com/>) that offered a free Trac instance for open source projects. Trac was simple, integrated, and hugely productive.

Trac was by no means the only option. Systems like SourceForge (<http://sourceforge.net/>) existed, but with Jokosher development so new, SourceForge felt like a very large hammer to crack a very small nut. Integration can go too far, and some software development solutions can include oodles of features and facilities that simply distract from what is important. Back then, SourceForge was vexed with this problem. Eventually Jokosher outgrew Python-Hosting and settled on [Launchpad](#).

Software As a Service

The Jokosher example raises an important question when it comes to integration. Do you want to run the toolchain facilities yourself or use an existing online system?

Integration has always been a challenge in IT, and subsequently many companies have sprung up to provide solutions. I have already mentioned Sourceforge and Launchpad, but there are many more. These solutions put integration at the center of their offering: they can give you most of the tools that you need in a single integrated system. These online services are often referred to as the *cloud* or *software as a service*.

Many new projects make the mistake of setting up their own tools for the sake of just controlling the tools themselves. This is a shortsighted perspective. But it's equally shortsighted to simply use an existing online service without thinking through your needs now and for the immediate lifecycle of the project. Let's think through some of these choices right now.

We should first look through the different attributes of both approaches. Let's start with benefits. Self-hosted systems offer a number of benefits and disadvantages, listed in [Table 5-1](#).

TABLE 5-1. Benefits and disadvantages of self-hosted systems

Benefits	Disadvantages
Control—you have complete control of the toolchain. You can always add features, facilities, and additions where you see fit.	Maintenance—having control is a double-edged sword. It requires maintenance, security updates, and the installation of updates.
Choice—you can choose which tools you want to use in your community.	Cost—if you host your own tools, you need to provide the hosting and bandwidth.

Now let's look at the benefits and disadvantages of a cloud-based solution, listed in [Table 5-2](#).

TABLE 5-2. Benefits and disadvantages of software as a service-based solutions

Benefits	Disadvantages
Ease—setting up your entire toolchain could be as simple as registering with a software as a service solution.	Data—when you use a software as a service solution, you put your data on someone else's system. You have to continually back up your data outside the cloud, in case that cloud provider goes away intermittently or permanently.
Maintenance—in the cloud someone else looks after security and updates so you don't have to.	Bandwidth limitations—some cloud providers limit disk space, bandwidth, and other attributes in a project. You may hit these limitations.

Benefits	Disadvantages
Potential reliability—software as a service solutions are typically hosted on high-bandwidth networks with a powerful backbone. This could potentially offer improved reliability over a self-hosted solution.	Tool choice—if you pick a software as a service provider, it may provide 80% of the tool choice that you want, but may choose a different tool for another part of your workflow. This could potentially limit what you want to do.

At this point your head is probably spinning with options, and you may be unsure of what to choose. My recommendation is almost always to choose a software as a service-based solution.

Software as a service-based solutions offer one major benefit: lower maintenance. The last thing you want to be doing in your community is spending time fiddling with tools. You should instead be focusing your efforts on growing community, building a team, and achieving the objectives and goals that you outlined in your strategic plan. Online resources such as Launchpad and Sourceforge are an excellent solution to this problem. They give you everything you need, and they put the maintenance and bandwidth responsibility in someone else's hands.

This recommendation comes with a caveat, however. We are still in the early days of the cloud, and it's difficult to know how stable the cloud will prove to be in future years. Many of the Internet companies that were prominent five years ago are either no longer around, have been acquired, or have otherwise faded into obscurity. When you put your data in the cloud, you face the risk that your provider may suffer these consequences.

I also worry about software as a service security perceptions. Storing data on a third-party service requires trust: trust in the provider and trust in the concept of the cloud. When that trust is compromised, the entire industry suffers. The airline industry has experienced this. When the attacks on the World Trade Center happened on September 11, 2001, all air travel felt unsafe. People avoided flying like the plague. The entire industry suffered due to a perception that air travel was no longer safe.

If one of the large cloud providers faces privacy or security issues, the concept of the cloud will feel unsafe and the industry will suffer. The risk to you is that if the cloud industry suffers and your data is on the cloud, your data could potentially go away. You need to assess the current industry and keep abreast of the risks involved.

BACK UP THE CLOUD

If you do decide to go with a third-party online software as a service provider, I again strongly recommend you have a backup solution in place, just in case you face these problems. Always ensure that if your provider goes away, it still can be business as usual in your community.

Avoiding Resource Fetishism

Before we move on to look at some of the technical specifics of different tools, I just want to spend a little time talking about a problem that faces a range of new collaborative communities.

The problem looks a little like this:

1. A new community is born around a goal. This goal could be to produce software, change the political landscape, or anything else.
2. The founders of the community set up the communication channels, and some early members join up.
3. Additional tools are required. This could involve a website, blogging tools, technical development facilities, or anything else.
4. Using a website as an example, a discussion kicks off asking which content management system should be used. A long and drawn-out debate starts over which system to use. These debates can drag on for a long time.

I have seen this same scenario happen over and over again. There is nothing wrong with wanting to choose the best tools for your community, but you need to get your priorities straight.

When you set up a new community, one priority is more important than all others: *building a team*. Your #1 goal is to get people involved. You need to get people inspired to join your crusade. If you don't attract people and get them up and running and productive as soon as possible, your community will feel like it is treading water.

Debates over which tool to use should be kept as short and sweet as possible. You should seek to establish requirements and gain consensus around a chosen tool as quickly as you can. Always remember that these discussions will garner strong opinions for different tools. You will never make everyone happy, so you need to identify the requirements and pick something as soon as viably possible. The sooner you get your tools up and running and get people focused on doing useful work, the better.

Technical Considerations

Most of this chapter has focused on best practices for assessing optimal workflow for a project, staying detached enough from specific technologies so that it can apply to any kind of community and workflow. The next few pages delve into some technical specifics of particular interest to software projects, such as bug tracking, source control, and collaborative editing.

As we cover these topics I am going to deliberately avoid recommending specific solutions or pieces of software. This is because software changes a lot more quickly than books do, and the last thing I want to do is recommend a piece of software that has since become unsuitable for your requirements. Instead, the following pages will advise on some key considerations that

you should keep in mind when evaluating which tools you want to use to solve a particular problem.

When evaluating specific software solutions, you should meld the advice here with your current requirements. You should also factor in predicted requirements for the next three years. If you can find a solution that will keep your project productive for three years, you have hit the jackpot. Our goal here is to keep migration to new systems to a minimum: migration is disruptive, complex, and time-consuming.

Bug Tracking

Bug tracking is a critical part of any software project, particularly open source projects. It is one of the core mechanics of how your project works. It should be a central part of your development process and a daily component for each of your developers.

There is a range of bug-tracking solutions available, from those your project hosts (such as Bugzilla) to a third-party-hosted system (such as Launchpad). Bug trackers vary hugely in terms of features. Some are simple, designed for small projects with a small number of developers. Some are hugely complex, infinitely customizable behemoths of functionality.

Before you choose a bug tracker, you should evaluate your requirements. Do you need a simple bug tracker to merely maintain a list of defects in your software, or do you need a complex solution with custom fields, multiple projects, teams, automation, and other features?

A common temptation for new open source projects is to install the most feature-packed bug tracker available. The reasoning behind this is usually “just in case we need those features.”

Whatever your needs, always keep usability as a key requirement in picking a bug tracker. If you use a solution that is too complex, it will annoy potential bug reporters and most certainly annoy those performing triage.

TRANSPARENCY IN BUG TRACKING

You should ensure that your project’s bug tracker is publicly available and that anyone and everyone can access it. As such, you should not restrict viewing the bug tracker to username and password access.

There are many reasons for this:

- There is really no need to restrict bug information if it affects a collaborative community project.
- Users who are experiencing problems can search quickly and easily for information that might help them fix the problem.
- Eric Raymond, a well-known open source proponent, once uttered, “Given enough eyeballs, all bugs are shallow.” Open bug tracking provides an excellent opportunity for community

members who are blighted by the same bug to explore, share notes, and possibly even fix the bug together.

The only time I consider a closed bug tracker or private bug reports to be suitable for community projects is when Non-Disclosure Agreement (NDA) work for a specific vendor is occurring, but in a manner that does not compromise the wider project. Bugs that affect the wider project should never be private.

Bug reporting

You should make every effort to make bug reporting as simple as possible. Some projects (such as Ubuntu) embed a link into an application that eases reporting a bug. Ubuntu also uses a special tool called Apport to bundle up debugging information to send to the bug tracker automatically when something crashes. This is a useful feature, as traditionally many developers doing triage would ask bug reporters to enter a series of commands that gather debugging information and add it to the report. Apport (<https://wiki.ubuntu.com/Apport>) automates this process.

Other projects make the act of reporting the bug on the bug tracker as easy as possible. An example of this is Jokosher, which only asks for a Subject and Description of the bug, leaving it up to the Jokosher triage team to manage the rest of the information. It is important to remember that your bug reporters are unlikely to know most of the requirements for a bug report.

Another project that has taken an interesting approach is GNOME. They produced a simplified frontend to their Bugzilla (<http://www.bugzilla.org/>) bug tracker, and the tracker asks only a few simple questions. It is recommended you keep your own bug reporting frontend as simple as possible.

Bug triage

For each bug that comes into your bug tracker, various decisions need to be made. This includes confirming whether it is actually a bug, asking for additional information from the reporter, assigning the bug to people or teams, and classifying the urgency of the bug.

Triage is a process that your developers will typically need to engage in, and few developers want to spend any more time on it than is absolutely necessary. As such, triage should be as straightforward and painless as possible. Understanding bug workflow, ensuring it is simple and effective, and picking a suitable bug tracker is a critical part of this process.

As your project grows, you likely will become inundated with bug reports. As the bug reports flow in, it is also likely that your developers simply won't have time to triage the bugs as well as fix them. As such, you may need to consider setting up a dedicated triage team. This, my friends, is a path that you should tread carefully.

Triagers straddle a delicate middle ground in your community. On the far left are your users who have been gracious enough to report a bug in the interests of it being fixed and benefiting the wider community. This is a generous gift on their part, and this gift should be rewarded with the attention it deserves. On the far right side are your developers who have only so many hours in the day to fix bugs. If the developers don't have enough time to triage the bugs, a triage team is expected to sit in the middle and help ensure the reported bugs are assessed and then routed to the right developers. If this process is successful, all bugs get triaged and the most critical and important bugs get fixed promptly by the developers who have the most appropriate experience in that part of the code base. If the process is *not* successful, the original users will grow resentful that their bug was "ignored," and the developers will either drown in bugs (not enough triage), never see bugs (lack of visibility in triage), or focus on the wrong bugs (wrong triage).

Setting up a triage team needs extensive work and resources to be successful:

Documentation

You'll need extensive documentation about how to triage, what is involved, where to find help, how to respond to users to gather more information, etc.

Mentoring

New triagers need to learn the ropes, and mentoring can help ensure these new contributors get up and running as quickly as possible.

Events

You should organize bug days, meetings, triage events, and other methods of inspiring participation.

Communication

Your triagers should never become an island: they need to be in regular contact with the developers and ensure communication is flowing.

Although it is certainly not impossible to set up a triaging team, and it seems like a fantastic method of contributing, many people often start with triage and grow bored. You should expect this: it is part and parcel of this kind of contribution. Have faith, though, because there will be some people who will join, enjoy, and stick with triage, and you should cherish these contributors and help them to inspire others to get involved.

Source Control

Source control has become an increasingly hot topic in the software development world. Once upon a time, the only real option was a tool called Concurrent Versioning System (CVS). We then had an "easier CVS" come along in the form of Subversion. Since then, we have seen the birth of Distributed Version Control Systems (DVCS), which add even more functionality to source control. DVCS systems make it easier than ever for new contributors to produce features and fixes that can be merged into the main branch.

I don't want to get into a debate about which version control system to use: other resources can provide a far better assessment of that. One piece of advice I do offer, however, is that you should strongly consider some of the integrated tools that exist around these systems. As an example, the Launchpad system integrates tightly with the Bazaar DVCS system and makes tasks such as reviewing merges and viewing changes a point-and-click affair. These kinds of features can be hugely useful with such an important component in the software development toolchain.

Collaborative Editing

Back in the early days of open source, we all lied to the world about documentation. “*Want to get involved in open source but can't write code? Sure you can help! Write documentation!*”

Back then, if you wanted to write documentation you needed to install a complex menagerie of text processing tools, write your documentation in a markup language, ensure the document properly validated, and then convert the documentation into something the user could read. In a nutshell, it was way too complex.

Most people who write documentation for open source software projects would fall into the category of “power user.” They are technology enthusiasts who are not interested in the super-technical avenues of programming, but want to help out. Many of these people have good writing skills and a good knowledge of using the software, so the documentation fit is natural.

With Jokosher we wanted to acknowledge this profile of user. As such, instead of focusing on complex text processing tools, we encouraged our documentation contributors to use a wiki. Those who did not know the wiki markup would use the graphical wiki editor. We got some great contributions because we identified the workflow with the type of contributor.

Building and Maintaining Transparency

Transparency is not only a key theme throughout this book, but throughout community building in general. Openness is an important ingredient in healthy communities, and your community needs to feel that there is transparency throughout its governance, processes, communication, and workflow.

In terms of workflow, transparency can be divided roughly into three areas:

Tool access

How open and accessible are the tools that form the foundation of your community and workflow?

Communication

How open and accessible are the communications channels in your community?

Reporting

How well do you report what your community is doing?

Each of these three areas is important for a healthy community. Let's now spend some time looking into each of them, and cover some of the key points in keeping your community open and accessible.

TRANSPARENCY FOR NEW CONTRIBUTORS

An important area in terms of transparency is how people who are not yet trusted contributors can contribute to the project to gain trust. This is covered in [Chapter 4](#).

Tool Access

You should always ensure that the tools that you choose to use for your community are easy to access, well documented, and freely available.

This has historically been an important factor for most open source projects. If you look at the majority of open source projects, the tools and dependencies that they use are also entirely free and open source. An example of this is the GNOME desktop environment. Not only is GNOME free and open, the tools used to build it also are free and open. This includes the toolkit (Gtk), C compiler (gcc), debugger (gdb), GUI designer (glade), and more.

Although most open source projects use free tools, many projects do use proprietary tools. Some years back I spoke at a Microsoft DeveloperDeveloperDeveloper! Day in the UK. As an open source guy, I was expecting to be greeted with a healthy dose of suspicion. Not only was everyone incredibly welcoming, but there was a surprisingly strong sense of community. There were a number of user groups and communities based around Microsoft technology, and in addition to this, many Microsoft fans released open source software, but they used proprietary Microsoft products to produce the code.

Although this is fine, I recommend using freely available tools. Contributing to a community where you can download the tool immediately for free is far easier than having to spend \$300 on a tool before you can participate.

When you have decided on your toolchain, you should ensure it is well documented on your community's website. You should specify:

- The tools that are used in your project.
- Where you can obtain the tools required.
- Instructions for how to use these tools to obtain the work that your community produces (such as the source code for an application) and how to run it.

Another useful goal to aim for, particularly with regard to software development, is to ensure that each contribution is documented. Ideally, you want any contributor to be able to point to

any of her contributions by referencing a URL on the Internet. If every contribution can be referenced, your community will be transparent by definition.

An example of this is how many open source projects have mailing lists that get posted to automatically with each new commit to the source control system. This simple solution ensures that anyone can subscribe to a list of source code updates. One alternative to a mailing list is an RSS feed of updates.

Communications

Openness in communication is essential. Your communication channels are the very lifeblood of how ideas, problems, and solutions flow between the different members of your community. The golden rule here is to ensure that anyone (including those who are not currently part of your community) can reference every communication online after it has occurred. I would like to be able to go to any community and read conversations that I was not a part of so I can evaluate the decisions that were made.

This is happening today in most open source projects. Most projects have open mailing lists, and these lists have publicly visible archives that are automatically made available on the Web. Many communities also have small programs called “bots” that log IRC conversations and make the discussions available on the Web.

I would highly recommend making these kinds of archives and logs available. In many cases, it is as simple as switching on a configuration option. The only time I consider it reasonable to have private archives is when a communication channel is being run by a Community Council. For many communities, sensitive and private discussions, particularly those around conflict, are taken to the council. There are also conversations that pertain to an individual, such as if a contributor has had some complaints made about him. These kinds of discussions need honest input from the council members, and if archives of the conversation are available, many council members will feel uncomfortable making statements that they would otherwise make in private. Just compare and contrast the statements you yourself make in public and in private.

Transparency in communications is not purely about the free availability of archives, though. It is also about ensuring that everyone in your community is welcome to participate and communicate in an open manner. There is one enemy to this kind of participation—*cliques*.

Cliques exist everywhere. Some of them are pronounced, such as the thousands of invitation-only clubs and associations across the world. Some of them are less pronounced, such as the informal, unwritten, yet obvious groupings that occur when we are all at school. There is nothing wrong with groups; they are natural functions of human social interaction. But they can be destructive in communities that are explicit about their openness to new contributors.

I noticed this for the first time years ago when I set up my Linux User Group in Wolverhampton. As the group grew and regular members attended, a sense of familiarity set in. There were insider jokes, everyone knew the regulars, and the group was generally bonding. Although we

welcomed new members, we occasionally heard of people being put off due to “cliquishness.” Although you should not have overt cliques in your community, you also should not inhibit the ability for people to bond and form personal relationships.

As you continue to engage with your community, there are some members who will naturally group together due to similar perspectives, sense of humor, or other attributions. Encourage and welcome these groups, but always be cognizant of the risk that these groups will be off-putting to new users. Everyone knows what it feels like to feel unwelcome, but not everyone knows what it feels like to be a possible cause of that feeling.

Reporting

The final aspect of transparency with regard to workflow is keeping everyone on the same page. This is all about reporting.

Some time ago, we wanted to improve how different parts of the Ubuntu community communicate what they are working on. We have hundreds of contributors all over the world, each working on different teams, and I was keen to see a single web page from each team with bullet points containing what they had achieved that month.

This was something of a challenge. Reporting is not a natural by-product of community, and most communities struggle at producing metadata such as this. Communities are great at doing the work that interests them, but activity reports and additional information does not flow naturally. The only chance of making this work was to make the process as simple as possible, and to encourage as many people to engage in the process as possible.

The process I developed was about as simple as I could get. I had a set template on the Ubuntu wiki, and each template was named for the given month. I asked all teams to get their feedback in by the 20th of every month. This would leave a few days to tidy up any missing pieces of the report and announce it on the 22nd of each month. For each team, I asked someone to be a team reports contact, and every month I would nag them to get their content in. Some teams would be as reliable as you could imagine. Some less so. In general, though, the team-reporting framework generated some useful content.

The value of the team reports was obvious. They were well received, and it was fascinating and inspiring to see what everyone was working on. The approach I fleshed out could easily be rolled out to your own community. You should seriously consider it.

Another little story I will share about reporting was one that we faced at our Ubuntu Developer Summits (UDS). At every UDS we would have a large number of discussion sessions, and we had some facilities in which people could listen in to the sessions through an audio stream and communicate with us via IRC. Despite these methods of engagement, we still wanted to release a set of proceedings from the summit. This would be a summary of decisions made at UDS that would affect the development of the next version of Ubuntu.

Our first shot at this was to produce a wiki page for each track at UDS and ask attendees to update the page at the end of each session. A simple process, but it got limited uptake. There simply was not time at the end of sessions for people to summarize the agreements. As such, we got relatively sparse proceedings for each track.

For the next UDS, we had a different idea. In recent years microblogging platforms such as Twitter (<http://www.twitter.com>) and identi.ca (<http://identi.ca>) have become a common part of many people's workflow. It is not uncommon to send out a quick "tweet" or "dent" to let the world know what you are doing. We figured this could be an interesting approach for our proceedings at UDS.

To do this, we conducted an interesting experiment that worked rather well. Using identi.ca we registered an account for each track at UDS. In each track room we had the login details for the account on the whiteboard. We would then encourage everyone to microblog as the sessions continued. The final part of the process was to write a script that took each of the microblog entries and divided the messages into the relevant sessions. As an example, all the messages on the community track between 1 p.m. and 2 p.m. would be automatically grouped under the heading "Making LoCo Teams Rock," the name of the session at that time on the community track.

The moral of this story is that to get effective reports, the method needs to be as low-friction as possible. People were already used to microblogging, so asking people to microblog sessions was entirely natural for many people. This is the essence of effective reporting.

COLLABORATIVE TEXT EDITING

Another useful tool that has been a common staple at UDSs is a collaborative and cross-platform text editor called Gobby (<http://gobby.0x539.de/trac/>).

In the Gobby text editor, multiple people can edit the same document at the same time. You see the changes that other people make in real time in your instance of Gobby. It is a clever little tool.

Gobby is useful when multiple people are working together to record notes for a session or working together on a specification or strategic document.

Regular Workflow Assessment

Like any process, structure, governance, or other agreed method of working, workflow should always be subject to change. Your workflow and the tools that are crystallized in it should always reflect the optimal way in which your community can work. If the tools become too complex and too laborious, you should consider adjusting your workflow, and possibly your tools as well.

An example of this occurred when I was involved in Jokosher. When the project started, we wanted a simple means of tracking issues and problems, and so we used the issue tracker that was part of our Trac development forge. We used the issue tracker for a variety of purposes, and it was simple and effective.

As the project grew, we were building up to shipping a major release. The project had more than doubled in size, and there was a significant chunk of functionality built into the application. When our community was reporting problems in the issue tracker, they had to enter a lot of manual information, and tracking and prioritizing this increasing number of issues was becoming a problem.

After a short discussion on IRC, we decided to move to a full-fledged bug tracker. We chose Launchpad as our solution, and we never looked back. Launchpad was more suitable, more flexible, and the right choice at that point in the Jokosher history.

One of the most useful lessons that I have learned in my community work is that you should regularly reassess everything: your workflow, processes, governance, and anything else. A community is built on a set of rapidly changing people, needs, and requirements. Regular reassessment is important to ensure that your workflow is matching the day-to-day work of your community.

I recommend that you set this regularity now. The length of time before reassessments is really up to you and your needs, although I would make sure to do a reassessment at least once a year.

Gathering Structured Feedback

When performing your reassessment, your aim is to find solid feedback about the good and bad aspects of your current workflow. The real value that you should be seeking is constructive, objective feedback from those who really dig into the workflow. Those community members who are using the workflow day in and day out will have the most valuable feedback for you. Before you consider changing any aspects of your workflow, you should gather an extensive level of feedback and use it to identify correlations in aspects that work and don't work.

A useful technique for getting good feedback is to produce an online survey. There are a number of free survey sites that enable you to produce a set of questions and have anyone fill in the survey. Many surveys also allow you to choose between invitation-only responders and a free-for-all public survey. This is an important decision when running the survey. There are benefits and disadvantages to both. For open public surveys, there is always a risk of getting a lot of inexperienced feedback that is not particularly useful, and this noise can skew the results. On the other hand, public surveys feel more community-spirited: they are open and accessible, and anyone can participate. Invitation-only surveys allow you to choose who responds, and if you choose wisely (read: not just people who will give you good feedback), you can get some great objective and practical results. On the flip side, invitation-only surveys feel closed, cliquey, and restrictive.

I recommend you do both. Start out with an invitation-only survey with at least 10 respondents. When you have this feedback gathered, open the survey up to anyone. Keep the public survey open for a set period of time (a month is a good figure), and promote the survey in places where your community reads and resides. You can then use both sets of results to draw conclusions. I would recommend that you put more faith in the invitation-only survey results, as they come from your key community members, but the public survey will undoubtedly uncover some useful results.

When you have performed your surveys, you should schedule some meetings with your community to discuss the results, propose solutions, and share more experiences. As with normal meetings, these meetings should be very focused on driving to a conclusion: finding solutions to the problems.

Moving On

Building effective workflows is complex. It involves a combination of understanding people, technology, and usability. As is a common theme throughout this book, you should look to simplicity as your inspiration. At every step of the way you need to ask questions. Is this more complicated than it needs to be? Is this the most effective way of doing this? Is my community going to be able to follow this workflow every day and not get frustrated?

Although we covered a range of concepts in this chapter, don't push yourself too hard, either. Experience and learning from your successes and mistakes is all part and parcel of being a great community leader. Some of the concepts in this chapter will take a little while to settle into your flow.

We are now going to move on to an essential part of leading a community: getting people fired up! We have some great structure in place so far. Now it's time to assemble an army....

Building Buzz

“A wise man will make more opportunities than he finds.”

—*Sir Francis Bacon**

ON FEBRUARY 17, 2000, BEN, MY PAL FROM UNIVERSITY, PICKED ME UP from my student halls of residence in Wolverhampton. Delicately held under my arm was a large cardboard penguin and a spindle of Linux CDs. Today was a special day: Microsoft Windows 2000 was released.

Just a few weeks before, on the other side of the world in Arizona, Deepak Saxena had an idea. With the looming release of Windows 2000, Deepak wanted to see Linux user groups and open source enthusiasts take to the streets to spread the word of open source. Deepak saw the opportunity, and was determined to make it count. He decided to convert a major milestone for Windows into a “Linux Demo Day” when thousands of Linux supporters would evangelize their love of free software.

Within a few weeks, Deepak had persuaded many of the companies in the Linux space—including Caldera, EST, Linuxcare, Linux Central, Linux Mall, Red Hat, SGI, SuSE, Turbo Linux, and TuxTops—to contribute to his cause. He had publicity from some of the largest online tech news sites, and over 150 Linux user groups interested in joining in. Deepak’s little idea had become a coordinated worldwide effort.

My group was one cog in this global machine. Five of us climbed into Ben’s car, which he drove at disturbing speed to nearby Birmingham. We proceeded to visit computer and software

* No, he is not my uncle.

retailers, bookshops, and libraries, and in some cases, walk up and talk to complete strangers. Unlike those strangers, we were unfazed.

Throughout the day, we explained and reexplained why open source was important. We talked about the ethos, the belief, and the community. We applied our enthusiasm to the slightly bemused subjects before us. It was a long day, but as we drove back to our native Wolverhampton, we were proud of our efforts. Our unity warmed us on that cold day in England.

Deepak's Linux Demo Day demonstrates the potential of community advocacy. Driven by a compelling reason to get out of bed and fueled by a combination of simple tools and elbow grease, the result was stunning. Whether or not the actions of February 17, 2000, had any impact on Microsoft is irrelevant. What was fascinating was the mechanics: the way in which a volunteer community rallied together behind a shared ethos on the same day, all around the world.

Deepak created *buzz*. He generated interest, excitement, and a willingness to contribute to a shared campaign.

This chapter dissects the science of buzz. Buzz is an important part of our community, and it is important that we get it right because missteps can often be problematic. As such, we will first discuss the key attributes of great buzz, what to avoid, and how to articulate ourselves. We will then move on to discuss the *buzz cycle* and the practical implications for different methods of building buzz around your project.

Mindshare

Every year without fail, an article bubbles its way to the surface claiming that this is the “*year of the Linux desktop*.” Each time, a new journalist steps up to the plate, and each time she gets it wrong. Although once an exciting and testy headline, today it is one big cliché.

The so-called “year of the Linux desktop” is commonly understood to indicate when Linux seriously threatens the market share of Microsoft and Apple. Year after year, we see these claims, and to be honest, I don't blame the journalists. We are bombarded with articles claiming huge growth in Linux. We hear about people migrating; new deals; large companies and governments switching over; and Linux features in magazines, books, and even television shows. It is easy to feel like those of us on the Linux side of the fence are rocking the world.

In our scramble to find party hats and shot glasses, it seems the analysts want to rain on our parade. Some cite 1% market share, some 5%, and some 20%. Whichever you choose to believe, the analysts are far more conservative. This rather predictably sets off a lengthy debate about whether you can count software that can be freely copied, over and over again.

You know what? In the end it doesn't matter.

Some vector within the great mythic noosphere, the “sphere of human thought,” probably knows exactly how many Linux desktops there are in the world, but actual usage doesn’t mean a bean in the scheme of things. What really matters is *mindshare*.

Mindshare is perception. It is a global gut feeling. Mindshare and perception is the magic that wins hearts and minds. It is also the explosive, seductive substance that clears the path for change, and it is mindshare that enables communities to have a voice.

The Mindshare Opportunity

Jamie Oliver is a television chef. Known in England for his young and fresh approach to cooking, he has always been popular with foodies and amateur cooks alike. Despite career success, Oliver was not content. He wanted to change something very specific: school meals.

The UK faced serious health problems in school kids. Obesity had doubled. Eighty percent of kids predominantly ate junk food, and the food parents were giving them was not exactly healthy. In a typical lunchbox, 55% of kids got potato crisps, 40% got a chocolate bar, and 33% got a carton of drink.

A significant part of the problem was the meals served to kids each lunchtime in school cafeterias across the country. Every day, 3.2 million kids were served on a budget of merely 68p. Unfortunately, that 68p was seemingly funding fried food and other junk.

Oliver wanted change. Although he had a television show about the topic and a reputation, he knew that he couldn’t change this alone. He worked hard to make the issue one that the nation cared about. Articles, blogs, radio shows, and other media were behind Oliver’s campaign. After a few months, it seemed that healthy school meals were the hot topic all over England. Oliver achieved mindshare.

With mindshare came change: meetings with the Prime Minister, new government guidelines, and better school meals for kids. Oliver got the entire country behind his crusade, and kids now have a better shot at a healthier lifestyle.

Mindshare has two potential outcomes, depending on which side of the topic you sit. If you agree with the campaign, it reinforces and strengthens your will and perspective. For those who are not on board, mindshare often causes them to reevaluate.

This harks back to our earlier example of Linux. In recent years, Linux has generated tremendous mindshare. The software and its formation have inspired a generation of technologists, who have in turn inspired others. This mindshare has caused many to reevaluate their choices in computing.

This was particularly the case around the time Windows Vista was released and a new buying cycle opened. Many organizations faced stiff hardware requirements for Vista, so it was a great time to evaluate alternatives, and both Linux and Apple prospered.

The Building Blocks of Buzz

In this chapter we want to build excitement and mindshare around our communities. We want to motivate, inspire, and encourage people to join us and achieve our goals together. If we get this right, we have the opportunity to build a great team and make some incredible things happen.

While some of you reading the book may come from a marketing background, it is important to stress that traditional marketing strategies probably won't wash with your community. You can't rely on press releases and advertising; you can't depend on buzzwords and slogans; and often you can't easily measure the impact of your message. Whereas marketing traditionally requires a budget, no budget is required here. Buzz in this world is subtler and more organic.

While we are making comparisons to traditional marketing, it is also important to note that building buzz in community does not need to be centralized. You don't need to appoint a marketing manager for the community and have everything go through him. You should instead encourage everyone in your community to learn the skills behind building great buzz.

This harks back to business expert Peter Drucker's often-cited lesson that great customers help create great customers. You should likewise help your contributors encourage and enthuse other contributors to join the community. I would highly recommend that you send this chapter to your contributors in your project to help them learn the building blocks of buzz.

The Mission

Every community has a mission. Back in [Chapter 2](#), we kicked off our strategic plan with our mission statement. In it we documented what we are setting out to achieve with our community. Your mission is critical to building buzz. First of all, you can't expect to generate excitement around an idea you can't articulate, so by developing a mission statement you sow the seeds of buzz. Second, developing a mission statement trains you to describe your goals and excitement succinctly. The mission is what will pique the interest of your next generation of community members.

The ideal mission statement is concise, specific, and functional; we worship vigorously at the altar of conciseness. Unfortunately, being concise, specific, and functional is not going to excite your potential community members. They are not going to get excited about strategic plans, either. They are not going to get excited about governance, and they are most certainly not going to get excited about processes. Our mission needs to dim the lights, throw on some Barry White, and get...sexy....

Uniting Together

Buzz is all about excitement, and excitement is about dreams. We need to present a dream so compelling that it will inspire people to join you to make it happen. Let's begin by taking some inspiration from politics.

I know what you are thinking. Politics? Sexy?

Although too many politicians focus on themselves, politics is an activity engaged in by communities. Political parties in democratic countries cannot achieve their goals without widespread support from those they govern. They gather this support by presenting a dream of the future that appeals to the electorate. Each of these activities has parallels with community building.

The similarities go further. Politics is, in essence, not all that interesting. Political parties spend their days arguing over policies, manifestos, taxes, inflation, and the other details of government. Around election time they have to transform these eye-glazingly mundane concepts into exciting statements that inspire a nation.

An interesting example of this happened in 1996. Putting political persuasions to one side, let's explore how New Labour built momentum around their political initiatives. It all started when a rather young and grinning Tony Blair became the leader of the opposition party in the UK.

Back then the country was deeply dissatisfied with the ruling Conservative Party. Prime Minister John Major had failed to enthuse the country, and the party was riddled with allegations of sleaze and political unrest. But despite multiple attempts to gain power, the Labour party had failed to unseat the ruling Tories for 18 years. Labour was branded by many with a "second-prize" political reputation.

Then, Tony Blair gave birth to his concept of New Labour. He was personable and different, an instant antithesis to the plain and stuffy political norms of the day. Blair formed his team and came bounding into visibility with his mantra of "New Labour: Because Britain Deserves Better."

Blair wanted to enthuse the electorate. Reams of inspiring, motivational language were unleashed on an unsuspecting British public. Consider these examples from the 1997 election manifesto:

I believe in Britain. It is a great country with a great history. The British people are a great people. But I believe Britain can and must be better: better schools, better hospitals, better ways of tackling crime, of building a modern welfare state, of equipping ourselves for a new world economy.

I want a Britain that is one nation, with shared values and purpose, where merit comes before privilege, run for the many not the few, strong and sure of itself at home and abroad.

I want a Britain that does not shuffle into the new millennium afraid of the future, but strides into it with confidence.

Pretty stunning stuff.

Although younger than his opposition, Blair was a savvy political player. He knew that the British public had lost faith not only in the Tories, but also in politics itself. He knew that the first time someone read, watched, or heard about New Labour, he had only seconds to grab them and build their support. He needed to build buzz, quickly and efficiently.

As part of this, he developed what Labour called a “Pledge Card.” This small card was carried by politicians and listed five concrete promises. Back then, the combination of “concrete promises” and “politicians” was not typical in British politics. Blair’s approach was honest and frank, and the nation responded. On May 1, 1997, Tony Blair’s New Labour party was elected into government in a landslide victory. Labour achieved the highest number of seats in the party’s history. I remember when it happened. I had a bowl haircut. The country felt invigorated and full of hope for a bright new future.

Inspired Words

A critical component in Labour’s victory was their ability to inspire. Words were carefully chosen to be visionary but not tacky. However, these words would have meant nothing if it were not for the underlying message and confidence in Tony Blair that promised a brighter new future.

Labour’s words offered huge hope for many who were disillusioned with politics. They spoke of “renew[ing] our country’s faith,” they talked of “our contract with the people,” and Blair dreamed of “a Britain which we all feel part of, in whose future we all have a stake, in which what I want for my own children I want for yours.” Blair’s ability to level himself with the rest of the country and inspire a joint feeling of change was a work of genius.

When delivered well, this kind of motivational writing can make those little hairs on the back of your neck stand on end. The use of words such as “vision,” “faith,” and “future” all help to paint the dream of freedom, opportunity, and in some cases, healing old wounds. This is the language that will inspire your future community members, and you should endeavor to use it where appropriate.

The greatest inspirational writers are those who have been exposed to great inspirational writing. There have been many great orators throughout history: Winston Churchill and Martin Luther King, to name just two. Many websites have recordings and transcripts of their speeches. I highly recommend listening to or reading them and identifying the parts that you find most inspiring. Watch out for that tingly feeling up your spine. It is that feeling that we want to create.

Inspiration is not limited to real politics. Many movies, TV shows, and books include powerful, inspired speeches written by hugely talented writers. I would personally recommend watching Martin Sheen as the President of the United States in *The West Wing*. He delivers some exceptional neck-hair-raising monologues.

There is no secret recipe for creating community enthusiasm. There is no blueprint. It is as organic and unique as the very community you wish to grow. The most important tip to remember about inspirational messaging is that it must be genuine. Whatever path we follow to build our communities, our leaders must instill a sense of trust and representation. The enemy here is in appearing contrived and disingenuous. Our inspiration must be natural and genuine. If you are real, your community will see that.

Unfortunately, it's easy to risk being seen as a fake. The word "community" has been twisted and contorted to mean many things over the years, some of which don't exactly fall in line with what many of us would consider a community.

The reason for this is that the word "community" almost always conjures up a positive mental image. It alludes to togetherness, compassion, and equality. As such, it has been a target of hype and hot air. If an organization, company, or individual wants to be seen in a positive and engaging light, liberal use of the word "community" makes sense.

There will always be a natural tension between the inspiring speaker and the audience. The speaker will always have a goal or agenda of her own, but needs to strike a balance between honesty around her personal ambitions and an assurance that she takes the wider values of the community to heart.

If your definition of community does not extend to the common expectation of volunteer environments, you will receive short shrift quickly. Readers seeking to build communities around commercial products should be particularly cognizant of this risk. Always remember that whatever their focus, communities are organic units of interest and collaboration. They live and breathe on an understanding that community efforts benefit the community as a whole. Anyone who tampers with that ethos will face problems. This is a critically important point, and I recommend you read this paragraph again twice. Go on, I will wait.

Becoming the Advocate

When we build buzz, we enter into a relationship with the audience. Advocates and salespeople each enter the same approximation of the relationship, but the rules of engagement vary. Both of these different roles encourage adoption of a product, technology, or lifestyle via positive messaging and reinforcement. How they differ is in how the person who is performing the messaging is perceived.

I have always considered myself an advocate. I define advocacy as the putting forward of a positive message and investing one's personal reputation in that message. Advocates recommend only products, technologies, and lifestyles that they personally subscribe to. Those

people on street corners who want to talk your ear off about human rights are advocates. They may be annoying, but they genuinely live their message. For them it is not a job; it is a lifestyle.

Sales are different. Some (not all) salespeople exemplify the philosophy of “I could sell ice to Eskimos.” In other words, they take pride in their ability to make the sale. The pride is in being convincing and charismatic. As such, salespeople are often able to detach themselves from the product they are selling. They can effectively work for any company that sells something.

I used to work at OpenAdvantage, a government-funded organization tasked with spreading open source software in the West Midlands region of England. Every year my colleagues and I worked with hundreds of businesses, charities, educational and governmental institutions, and individuals. OpenAdvantage was a vendor-neutral playground. We could recommend whatever solutions we liked to our clients. This became a trial by fire. In my two years there, I tested hundreds of different tools, systems, and applications. In that time we developed preferences that were personal and not mandated by any other agreement or contention. Like any government-funded project, it had a time limit. We had two years. Around six months before the sand was due to empty out of the hourglass, I started looking at other opportunities.

At this point I discovered I was really an advocate: I found it near impossible to consider working for a company that made something that I did not 100% choose, believe in, and make part of my life. I did some stringent analyses on the products I used every day, and this is how I came to work at Canonical: Ubuntu was my choice of operating system.

This makes life tough for an advocate. It means that in your heart of hearts, you can get out there and shout from the rooftops only about what you truly believe in. It also requires that implicit honesty in both directions: if your product sucks, you don’t cover it up but instead try to fix it.

The upside is that when people know you are an advocate, and know this is how you tick, your opinion *really* counts to them. Advocacy requires trust: you are putting your belief behind your words. If people like you and trust you, they are likely to trust and like what you are advocating.

Getting It Right by Not Getting It Wrong

Trust is complex to achieve and easy to lose. Months of positive moves are required to gain it, and one misstep can shatter it. With trust comes responsibility, and you have a responsibility to approach your buzz in the way that you would want others to approach you.

Making a lot of noise is easy. Spammers do it. They send out millions of emails advertising a product without any concern for whether the recipient might be interested. The last time I checked, I really didn’t need Viagra. Out of the millions of emails that they send, some tiny proportion of people will buy spammers’ products. With email’s low cost to send, the cost of annoying the world is relatively cheap and the financial rewards are real. Unfortunately, becoming the scourge of the Internet is a price some people are willing to pay.

Of course, the major problem with spam is that it is unsolicited. I don't choose to receive emails about Viagra and Cialis. I never signed up for an announcements mailing list. I never informed anyone that I wanted to receive that email. I received the email purely because I have an email address. Spammers don't care who receives the spam, and anyone who can, will.

To avoid the disgust that this provokes, we need to always ensure that our buzz is relevant. When we seek to excite and inspire people to join our community, we must target the right demographic. Our buzz always needs to be honest.

We are all human, and we will all be tempted at some point to target those outside our demographic. Don't. Sending unsolicited messages to people, no matter how you do it, will not only frustrate many people, but it could harm your community. Who would want to join a community with a reputation for spamming? I certainly wouldn't.

Honesty

Unfortunately, honesty is more complicated than you may think. It is not just about the black-and-white approach of not lying. Even the kindest hearts can sometimes get a little carried away with the buzz-building and promote an image and a culture that is not entirely representative of the reality. It is tempting to lavish a more exciting spin on your community.

Don't be tempted into building a false prophecy: it never works. There is nothing wrong with getting excited and enthusiastic. Tell the world your community has grand ambitions, and inspire people to join. Just don't sell your prospective contributors a story that doesn't exist. Don't tell them you have achieved things you haven't. Don't tell them that you have more people involved than you do.

Community is all about transparency and openness. In such a frank and unclouded environment, such exaggerations and fibs will be outed sooner than you think. When such nonsense is revealed, it causes a lack of faith in you and your community. Remember, *trust is a nonnegotiable requirement of community leaders*. Don't risk it.

DON'T BUILD YOURSELF UP BY TEARING OTHERS DOWN

Buzz is a great thing. It's all about the glass being half full. It is about celebrating opportunity, and identifying what collaboration can achieve when we all share the same passion.

Unfortunately, some communities and leaders feel that spreading negative energy is an acceptable way of furthering their own communities. Second to fibbing about your community, disparaging another community for your own benefit is about the worst thing you can do. It makes you and your community appear petty and indignant. Unfortunately, particularly in online communities, this happens more often than not.

As members of your community get more and more well known (and this could include you), a sense of representation often sets in. People begin to feel that they need to not only defend the community at all costs, but to do it publicly. We then see disagreements and spats played out in embarrassing detail, making the participants look like overopinionated teenagers.

Setting Up Your Base

Many see buzz as a one-way street, analogous to television. They see it as a broadcasted message that people consume. They also see our primary goal as producing a message to be consumed. This is really only half of the goal.

When building buzz, we need to understand what we want. Our buzz is about our community and what our community achieves. When people experience our carefully crafted messages, what do we want them to do? What is *their* next step? What is *our* next step?

A website is *essential* to achieve this. Whether your community is focused around technology, knitting, animal welfare, supporting the poor, or otherwise, a website is a critical resource that you should build and maintain. The ubiquity of the Internet and the low cost of equipment to view it have made an online presence the storefront for your community. If someone hears about your community, the first step he will take is to search for it.

Whether someone hits your website because he stumbled upon it or whether he read about it as part of your buzz, now is the time to grab his attention. You need to enthrall him and pump him up, and while he is engaged, get them up the ramp to be a contributor as easily as possible. Unfortunately, many communities don't do this very well.

Some still see the Web as a largely static medium. You put up a web page and once it is up, it never changes. These folks see websites merely as electronic information leaflets. My brother Martin used to be like this.

Martin is heavily involved in conservation, and he runs an organization in Northern England called Rotters (<http://www.rotters.org/>). He and his family worked hard to secure land and set up facilities to perform community composting, train ex-prisoners and unemployed members of the local area, and to run events. One such event is his woodland festival: a cornucopia of rural attractions, including metalwork, chainsaw tree sculpture, performance art, and live music.

Martin is an outdoorsy guy. He spends all day every day outside with his work. He is the kind of guy who just enjoys being knee-deep in mud and soaked to the skin on a cold English day. Yes, he is mad. Martin was never much of a computer guy. His organization's website was a fairly boring-looking page with information that was once current but never got updated.

But computers became more intrinsic to his life and his organization. He started keeping accounts there, printing fliers and other publicity material, and using eBay to buy and sell equipment. Before long, he and his team rebuilt the website into a vibrant resource about the organization. It had pictures, stories, news, and more. Since the website has been refreshed, Rotters has received more publicity, members, and interest.

Your website is your community's Embassy of Buzz. Make it rock.

Aims

Building a great website may seem a daunting process. The foundation, though, is simple. Your website should seek to satisfy the following aim:

Provide a current resource that answers questions and maintains a relationship with the reader.

Write this down and stick it on your wall. Tattoo your children with it if you need to. Let's break this down into key areas. Bear these aspects in mind when considering how to build your site:

Great overview

If I visit your website, I want to be able to get an overview of the community, its goals, and how to get involved, all within the space of a few minutes. This information should be up-front, easy to access, and easy to read, and should have a simple web address that you can point people to (e.g., www.myproject.com/overview).

Great documentation

Your website should provide documentation and guidance for all aspects of your community. This will always be an ongoing process, but you should consider which aspects of your community are most important and need to be documented. As an example, the process that new members follow to join your community should be a priority.

Great communication

You should ensure that it is really easy for people to get in touch with you with questions, suggestions, and feedback. I would recommend that as a minimum you have an email address that is easy to access. This is a notable problem with many blogs. Many blogging tools (such as WordPress at the time of this writing) do not provide an easy-to-access contact link, and this can make it impossible to get in touch with the author of the blog.

Building a website is a little like painting a picture or writing a song: it is never finished. Creative types always want to add a last fleck of paint or final flourish on the guitar. With a seemingly endless stream of possibility with your website, you need to prioritize. To do this, write down a list of everything you want to achieve, and then order it by what you consider most critical. Naturally, elements such as the contributor ramp should be near the top of the list.

Staying Current

The most important element in any website is content, and the most important golden rule with content is that it needs to be *current* and *accurate*. Of course, the two terms are somewhat intermingled: content about your community that is on your website from 1998 is probably not going to be all that accurate any more.

Creating current content is a two-part process. First, your community processes and methodologies need to be up-to-date. There is nothing more frustrating than joining a community, following the published guidelines to get involved, and then being told that the guidelines are out of date and that you need to adjust your work. That is a surefire way of annoying new contributors.

The second area, and the area that fits in neatly with *buzz*, is news and updates. Your site needs to be a window to exciting content about the community, and this window needs to be updated regularly. If you want people to keep coming back to your website, you need to give them a reason. That reason is fresh content.

You should build a strategy for publishing news. This in itself may seem like a simple task. I know what you are thinking: “Jono, I will just regularly post something new to the website. Simple!” Not quite.

Many communities struggle with regular updates. People simply get busy. People get distracted. Other priorities in the community take over. If all of your news gets posted by one person, and that person decides to spend evenings redecorating a house instead, you lose. When people get busy, they tend to drop nonessential and nonfun tasks. News updates are often one of these first casualties.

So divide this work between a number of contributors. Find three or four people who are happy to update the news site, and talk to each other about what news you are going to post.

Building Conversation

Posting news is one way to keep a website current. There is, however, another approach to keep your site alive: *conversation*. The Internet has become an incredible place to have conversations. Countless forums, websites, and blogs have provided a medium in which anything and everything can be discussed. You should ensure your website is in on the action.

Conversation and commenting facilities are an incredible way of making visitors feel that they have input in to the community. These facilities are typically fairly simple: a blog or news entry is posted, and readers can leave a comment underneath the content to share their thoughts. This has two benefits:

Regular content

Conversation is content that other people provide to your website. Your visitors will come back to participate in the conversation, and your website will always be fresh and full of new and interesting things to read.

Engagement

Allowing any visitor to the website to post an opinion or comment is an incredible statement about engagement. Providing this facility says to all of your visitors that your site is open and always welcoming comments.

These two features in themselves are justification enough for having these facilities on your website, but there is one even more important reason: *community*.

If you allow people to participate on your website, building a reputation tied to their name, community will thrive on your website. Community growth is our end game, and growth happens around conversations.

I have experienced this in a number of places, with the most personal being my own website at <http://www.jonobacon.org/>. For a number of years I have had comment facilities on my site, and over the years I have developed a regular readership that comes back to contribute to the topics of the posts. If this is possible on an individual's website, just imagine what is possible on a community's website!

KEEPING THE CONVERSATION FLOWING

Providing commenting facilities is only the beginning of growing community on your website. To do this you will need to be more proactive than normal in encouraging conversation. Here are some quick tips:

- **Be responsive.** View every comment to your website as seeking a response. Respond and ask questions to keep the comments coming in.
- **Write news that generates discussion.** In your news items, you should explicitly ask for feedback and responses.
- **Be thankful.** Regularly thank your contributors for great comments and feedback.
- **Reward the regulars.** You should think carefully about how to reward your regular contributors. One approach I have taken with my own blog is to give the top three contributors a gift, such as a DVD or book. This shows everyone that you really care about their participation.

At the beginning of growing your community, you will need to be proactive, but as conversation within the community grows, you can take a step back.

Getting Online

Fortunately, you don't need to have super-technical skills to get a great website with conversation facilities up and running. There are a number of preexisting tools that will do the job perfectly. The easiest method of getting going is to use a *blogging engine*.

A blogging engine is a software tool that lives on a website and provides a simple interface for you to publish an article. You can use an editor to format the article in different ways (make parts bold, italic, underlined, different headings, links, images, etc.), and you can also split the posts into different categories. These systems do not require any programming abilities.

There are many free online services for setting up blogs, and with them you can get up and running quickly. They allow you to publish your blog entries and ensure they are indexed by popular search engines, such as Google. Unfortunately, many of these services don't allow you to have your blog at your own web address.

If you would prefer the blog to be on the same domain as your website, you can install your own blogging engine. There are many free and commercial blogging engines available, but I would personally recommend WordPress (<http://www.wordpress.org/>). WordPress offers an excellent, easy-to-use, and powerful framework for publishing content easily. To install your own WordPress website, you will need to have your own server, which you can purchase from a hosting company. They will provide a place to install the software, and you can set it up. Many hosting companies also offer preinstalled WordPress, Drupal, and other systems.

ROLL YOUR OWN VERSUS PREEXISTING ENGINE

When the need for a new website arises, some people prefer to install an engine such as WordPress, and some prefer to write a site from scratch.

I highly recommend that, where possible, you use a preexisting engine. Years ago I wrote my own website engine, and it just wasn't as good as something such as WordPress. It's fairly simple to see why: a project such as WordPress has hundreds of developers creating new features and fixing problems. This nets more functionality and a more stable experience in general.

If you really feel the need to have your own custom website, ensure that you have the resources to maintain it for a long time. When you invest in that website today, you will need to be able to still invest in its maintenance in a year, in two years, or more. Think carefully about this decision: if you go with a preexisting engine such as WordPress, that maintenance is done for you.

Syndication

In the last five years there has been a change in how people access content on the Internet. Traditionally, the only way you could get updates from a website was to repeatedly visit the website to check for new content or features. This meant having a number of bookmarks in your web browser and cycling through them hunting for updates. Fortunately, a solution to this problem has been developed, known as *syndicated feeds*. Once the forte of the techie, they are now popular across the Internet.

When you use a blogging website engine (such as WordPress), each time you add a new entry, a special feed will be updated with the content. This feed sits on the same website. Your readers can then use a piece of software called a feed reader to subscribe to the feed.

As an example, on my website at <http://www.jonobacon.org/>, I have a feed available at <http://www.jonobacon.org/feed/>. This is a Really Simple Syndication (RSS) feed.

NOTE

A good introduction to these feeds is on the BBC website at <http://news.bbc.co.uk/1/hi/help/3223484.stm>.

With most websites providing these feeds, you can subscribe to a number of your favorite websites in your feed reader. Each time you load the reader, it will check each of the feeds for updates and indicate if there is new content. This is not only a hugely efficient way of reading lots of websites, but the feeds typically don't include all of the website imagery and design. This means you just get the content and don't have to download all of the other fluff.

You should ensure your website provides these feeds. Most engines (such as WordPress, Drupal, Joomla!, and others) provide support for these feeds by default, making it really easy to include this functionality on your website.

Syndication feeds also allow people to take your content and merge it into their own website. This has been happening more and more in recent years. Tools can easily embed feeds into the sidebar of a website, for example.

There is also a new kind of website appearing called a *planet*. These websites take a number of syndication feeds about related subjects and show the posts in the correct order. This produces a rolling collection of interesting content: readers simply visit the planet and then read a large number of related websites that they were probably unaware of. A great example of this is Planet Ubuntu (<http://planet.ubuntu.com>).

You should seek to get your feed syndicated to these other sites. It can drive some incredible traffic to your site and get your words out further afield.

SEARCH ENGINE OPTIMIZATION

A popular buzzword in the online world is *search engine optimization* (SEO). It is the science of how to ensure that your website appears at the top of search engines. It is a large and complex science, but I want to give you just a few quick tips that will get you started:

- Using preexisting engines can make this easier. When you use a blogging engine, the creators of the engine have likely already thought about optimizing for SEO. This will automatically get you higher rankings.
- Make sure titles and headings (which factor heavily in searches) are meaningful and contain the words people will look for. It's OK to have a cutesy title like "Why I'm tired this morning," but also include some meaningful indicator of the content, such as "...because I finished the Bike for MS research fundraiser."
- When posting images inside your news/blog posts, ensure that you specify some text to describe the image. This is done in the alt attribute of the image tag. For example:

```

```

- Comment and conversation facilities will increase your SEO by bringing regular traffic to your website and encouraging links.

The key to SEO is having great content that attracts regular traffic. Focus your efforts on getting more people to your website, and your SEO rating will flourish.

Microblogging

In recent years the [Twitter](#) phenomenon has taken over the Internet, creating what many call *microblogs*. While other alternatives to Twitter exist, such as the open source [identi.ca](#) and Facebook's status facility, Twitter continues to rule the roost.

For those of you who have no idea what on earth I am talking about, let me explain using Twitter as the primary example.

When you sign up to Twitter, you get a user account that you can point people to. As an example, mine is <http://www.twitter.com/jonobacon>. I can then type in a short message of no more than 140 letters, and it appears on that page. These messages are quick bursts of what is in that person's mind at the time, and typically contains what they are doing at the moment, interesting thoughts or links on the Internet, random musings, and other content. As an example, here are my last five messages at the time of writing:

Up where I was born in North Yorkshire seeing family, having a wonderful time. Heading out tonight for a bit of a gathering. Fun. :-)

!ubuntu !ubuntulocoteams !ubuntudevelopers Folks, remember to send me over the photos from your release parties - <http://tinyurl.com/d2uh3s>

At the office. Tired after a busy week and the release party last night. Rolling sleeves up to attack the inbox. This is going to get ugly.

Ubuntu 9.04 is out! Thanks for everyone's hard work on a fantastic release !ubuntu !ubuntudevelopers !ubuntulocoteams :-)

Btw folks, point your beady ones at <http://www.jonobacon.org/2009/04/22/ayatana/> - new Design focused desktop love in the form of Ayatana.

NOTE

Some of these messages contain some words beginning with an exclamation point (e.g., !ubuntulocoteams). This is a feature in the identi.ca service (an open source equivalent to Twitter) that sends the message to a group that users can subscribe to. This gets your message out to more people.

Using Twitter, you can then subscribe to someone's messages, or they can subscribe to yours. This is called *following*. This provides a means to see a chronological list of thoughts from people you are interested in.

Many of you will be wondering why on earth anyone would care about this. When I first heard about Twitter, I was no different. In fact, I was bullied into Twitter when a few friends of mine registered an account under my name and started posting joke messages. When they finally surrendered the account to me, I decided to give it a whirl and have been hooked ever since. Microblogging offers many benefits:

Hugely viral

Microblogging messages seem to get everywhere. People read your messages and decide to subscribe, others point to your messages, they appear on search engines, and many microbloggers have a widget that shows their latest messages on their website.

Lots of content

Microblogging requires far less time to engage in than wider structured articles, such as conventional blog entries. As such, you generally see far more content.

Current

Many people watch their microblogging feeds throughout the day. This currency has been demonstrated in many cases. As an example, a friend of mine went to San Jose and tweeted asking for a great seafood restaurant and had multiple recommendations within minutes.

Great messaging medium

When people start subscribing to your messages, it provides an excellent channel to send content to.

The last item is the most interesting part of microblogging. If you make use of microblogging effectively, you have the opportunity to build an audience to regularly send content to. Many assume that Twitter is the only option here, but there are others you should sign up for, too. The three I primarily use are Twitter, identi.ca, and Facebook. These three resources have become a compelling audience for my content. As an example, at the time of writing I have 2,000+ Twitter subscribers, 800+ identi.ca subscribers, and 1,400+ Facebook friends. This is significant group of people who can read my content whenever I send it out.

To make this easier, you can use tools to cross-post content to each of these resources. I use a tool called Gwibber (<https://edge.launchpad.net/gwibber>) in which my message is posted to both Twitter and identi.ca, and I have installed a Facebook application that posts my Twitter messages as status updates in Facebook. Finally, I have included Twitter messages on my blog so that my audience there can read them.

Whether you have a single Twitter account or the chained-together system that I just described, there are some hints and tips that you should bear in mind when using microblogging to build buzz:

Be spontaneous

Much of the reason why microblogging has taken off is that it is perfectly acceptable to post a quick one-liner with something that is in your head. Not everything needs to be carefully considered.

Be yourself

For most people I know who are fans of microblogging, they love how it offers an insight into the lives of the microblogger. As such, feel free to mix together multiple topics. It's perfectly fine to wake up in the morning and send a message about your awesome cup of coffee, then to mention a cool project in your community, share a great link to something you read on someone's blog, complain about the terrible service at lunch, and then point your community to a website where you want to gather their feedback.

Don't txt

Many newcomers to microblogging hear that it is comparable to SMS text messaging and then write in txt speak (e.g., "Gr8 time at comedy show. Much lols. Thx Mary :-)"). Feel free to use full, normal words that the rest of us understand.

Always link

If you are talking about your community, link to something where the reader can find out more. Microblogging is a great springboard to your community resources, so feel free to share them.

Share achievements

When something cool in your community happens, microblog it. Encourage others to do the same. This not only spreads the word, but also acts as a rather nice public pat on the back to the folks who did the cool thing.

Gather feedback

Microblogging is a fantastic method of gathering feedback. This is something I used while writing *The Art of Community*. I used it to ask my subscribers for their thoughts and opinions on given subjects that I was writing about, and I got some fantastic responses. Due to the medium being limited to 140 characters per response, ensure that you ask for very specific feedback so people can fit it in.

Share events

If you have an event scheduled, such as an online meeting, microblog when the meeting is announced and also when it is an hour away. Many people will read the message and join. I discovered the power of this when I did my first videocast. I switched the camera on and microblogged it, and within minutes 24 people were watching.

As you can see, there are many advantages to microblogging, and although it seems redundant at first, give it a whirl. I am willing to bet you find it useful in your community.

The Buzz Cycle

So far in this chapter we have explored many of the wider considerations for building buzz. Before we move on to look at some specific examples, we need to learn the final piece of buzz-theory: the *buzz cycle*.

Whenever you build buzz, you execute on a set of procedures. When combined, this set of procedures ensures that you plan effectively, get as much anticipation for your announcement, and learn from the experience. These steps help frame the best practice involved in buzz making, and they will help you to better plan and structure how you get people excited about your community.

The four stages are:

Planning

Sitting down and building a recipe for what you want to achieve, how you can achieve it, and what is involved.

Buildup

Instead of going straight to the main course, why not start with an appetizer? Build up some excitement and mystery before the main event kicks in.

Announce

The core of our buzz, this is where we kick it out there.

Review

A postmortem of what we did, and an assessment of what worked, what didn't, and how we can improve next time.

Many newcomers to the buzz-building business jump straight to the announcement, with a marginal level of planning. I would strongly recommend against this. Buzz is an art form that

can net incredible results for your community when executed correctly, but it can also cause lasting damage if you get it wrong. Planning and feedback will keep you with the former. To explain how each of these stages are important, I am going to use the 5-A-Day example that we talked about back in [Chapter 4](#) that illustrates the buzz cycle well.

5-A-Day was a project that I conceived while watching a program about healthy eating. In many countries it is recommended that you should eat five portions of fruit or vegetables as part of a healthy diet. It makes healthy eating an easy-to-remember metric that people can factor into their routine, which is a compelling concept.

Around that time, we were very conscious of how we handled our bug list. As Ubuntu grew as a project, the number of users grew; as such, so did the number of reported bugs. Inspired by five portions of fruit and vegetables a day, we formed the 5-A-Day initiative to encourage our community to triage or comment on five bugs a day. The project started and made some incredible progress. Now let's look at the different buzz stages with this example as an illustration.

Planning

The reason buzz requires planning is that, to excite people, you need to know your goals, what tools and resources you need, and how to roll out your plan. You want to squeeze as much juice out of your efforts as possible and get as much focus and attention on your community as possible. You want maximum return for your investment in time.

First, it is time to sit down and consider the different attributes and elements in your buzz initiative. Here are some questions that you should have answers for:

- What outcome do you want to achieve?
- What medium(s) are you going to use to achieve it?
- What preparation work needs to occur before you can begin the buildup phase?
- What other people are involved in the buzz and what are their tasks?
- What is the timeline for the entire buzz cycle?

The answers to these questions will give you a firm idea of your goals and how you can achieve them. For plans that involve only you, an awareness of the answers to these questions is enough. If you are going to be working with other people, however, you should document the answers. This will ensure everyone is on the same page.

In the case of 5-A-Day, I was working with my team, Daniel Holbach and Jorge Castro. The preparation work involved the development of some technical facilities and tools, some documentation, and a timeline. We had a number of conference calls to build the plan, ensure the requirements were in place, and to specify deadlines for each of the buzz cycle phases.

DEADLINES KEEP YOU ALIVE

Many people hate deadlines. They commit us to specifics. For many, and particularly those who enjoy the free-form nature of community, deadlines are unwelcome.

Stick with them, though. Deadlines are critical to achieve goals. In this chapter our goal is effective buzz. When you have multiple people involved in a buzz-building exercise, you need to ensure everyone delivers their contribution to the project on time.

When you apply deadlines, ensure they are documented somewhere. For my team, we plan the deadlines up front and put them on our shared calendar. This is a useful means of reminding you when deadlines are near. The key is ensuring deadlines are in a place where you will look. If they are buried away in a file or notebook, they are of no use to anyone.

Buildup

The next step is when things start to get exciting. This phase brings to mind the often-trumpeted statement “some things are better left to the imagination.” It’s true.

In this phase we want to tease people with a taster of what is to come. We want to pique their curiosity, tempt their senses, and get people chattering about what we are up to. When done right, this phase can deliver some riveting and memorable buzz, before you even announce what you are doing.

I also used this technique to announce that I was working on *The Art of Community* (<http://www.jonobacon.org/2009/01/13/announcement/>). A few days before I announced the project and the website, I took a screenshot of the website and motion-blurred it. I deliberately blurred it so that you could not see what was on the site, but you could just make out the word “Community.” Underneath the screenshot I simply wrote “*Wednesday 14th January 2009 @ jonobacon.org.*” A flurry of over 35 comments then appeared, each musing on what the project could be. Many even tried to unblur the screenshot to see what was there. An hour before I posted the main announcement, a reader called Kyran managed to unblur and provided a link to the new website.

On the 5-A-Day campaign, too, we had an interesting idea. Over the week building up to the announcement, Daniel; Daniel’s girlfriend, Mimi; Jorge; and I each posted photos to our blogs that had us showing symbols with the number 5 in them.

My first blog post included the photo in [Figure 6-1](#).



FIGURE 6-1. A photo designed to build anticipation in the 5-A-Day buzz campaign

(Although I was clearly trying to look cool in the photo, the world and his dog seemed to be mostly amused at the fact I was watching *Along Came Polly* on my TV in the background. Buzz can sometimes backfire.)

Underneath the photo, I also pulled some text from Wikipedia about the number 5 and put it underneath the photograph:

5 (five) is a number, numeral, and glyph. It is the natural number following 4 and preceding 6.

Five is between 4 and 6 and is the third prime number, after 2 and 3, and before 7. Because it can be written as $2^{(2^1)+1}$, five is classified as a Fermat prime. 5 is the third Sophie Germain prime, the first safe prime, and the third Mersenne prime exponent. Five is the first Wilson prime and the third factorial prime, also an alternating factorial. It is an Eisenstein prime with no imaginary part and real part of the form $3n - 1$. It is also the only number that is part of more than one pair of twin primes.

Five is conjectured to be the only odd untouchable number.

When viewed together, particularly on Planet Ubuntu, the blog posts were clearly connected. This started a flurry of discussion about what we could be up to.

NOTE

It should be noted that buildup should only be used on genuinely interesting initiatives. Don't bother using buildup on things that will fail to excite people. As an example, buildup would be great for a new project or initiative, but awful for a change in policy in how your community is governed.

Announce

At this point in the cycle, there should be some rampant speculation regarding the hints you have been dropping in the buildup. You should be seeing suggestions from the sublime to the ridiculous. Don't go too far with the buildup, though. Allow just a few weeks before you reveal your mystery with an announcement.

When announcing you need to ensure you answer all of the most immediate questions the speculators have. If after all the buildup you don't come through with a smörgåsbord of answers, it will simply cause frustration. You want those riddled with curiosity to be delighted to have their curiosity quenched when they hear the news.

The first step when announcing is to point someone somewhere to read, hear, or watch your announcement. You should have a single location to direct people to. For most communities, this is a website. Your goal now is to make the page easy to read.

Most announcements that communities tend to make are posted on their website, but there is an important consideration to bear in mind with web announcements: computer screens are hard to read. Jakob Nielsen, one of the world's highest regarded usability gurus, wrote about the impact of screen text on readers (<http://www.useit.com/alertbox/whyscanning.html>):

Reading from computer screens is tiring for the eyes and about 25 percent slower than reading from paper. No wonder people attempt to minimize the number of words they read. To the extent this reason explains users' behavior, they should read more when we get high-resolution, high-scanrate monitors in five years since lab studies have shown such screens to have the same readability as paper.

With reading on screens known to be more tiring, this behavior naturally translates to people wanting to read less and scan more. As such, we need to deliver our announcement quickly and effectively.

It's important that we get our announcement in perspective: it is going to be one of hundreds of things that the person will read on the Internet that day. We need to stand out. We need to grab the reader's attention and deliver our content.

Nielsen's solution to this problem is simple: write half as much. In his excellent "Writing for the Web" article (<http://www.useit.com/alertbox/9703b.html>), Nielsen recommends three guidelines that can help:

- Be *succinct*: write no more than 50% of the text you would have used in a hardcopy publication.
- Write for *scannability*: don't require users to read long, continuous blocks of text.
- Use *hypertext to split up* long information into multiple pages.

We are trying to avoid swathes of text. We need to architect our announcement so our readers can skip over parts and get straight to the meat.

Let's look at an example. Imagine we are writing an announcement to solicit papers for a conference on renewable energy. Let's start with a high suck factor and write an announcement we can tear apart afterward:

Call for Papers Open

Cranfield Green Alliance is a renewable energy conference that takes place in Cranfield, Bedfordshire. The conference is located at Cranfield University and runs from 10–12 November 2009. The conference covers a range of topics including renewable energy, alternative lifestyles, green cooking, ecological trends, and more. We are now opening up our call for papers and encourage a variety of environmental professionals to submit presentations in their area of expertise. Papers on a range of subjects are welcome and we would encourage you all to submit something soon. The conference attracts a wide range of attendees and exhibitions, and we welcome your involvement in this important event. Your contributions as a visitor or a speaker will be valuable. To submit your paper you should email papers@cranfieldgreenalliance.co.uk no later than 1st October 2009. We look forward to your submissions!

Friends, what we have just experienced is unremarkable, flat, and about as exciting as a paint-drying competition. I am sorry I subjected you to that paragraph: I realize you will never get those minutes back. Consider it a sacrifice for your community.

OK, let's apply some of the guidance we have discussed so far. Let's make the language exciting and inspirational, break up the paragraph so it is easier to read and scan, and make it succinct and clear. Tighten your seat belts. Here we go:

Cranfield Green Alliance Call For Papers Open!

10–12 November 2009: Cranfield University, Cranfield, Bedfordshire, England.
Leading the way to define a new future fueled by renewable energy.

Including exciting and industry-relevant topics such as renewable energy, alternative lifestyles, green cooking, ecological trends, and more. Leaders of the field bring a great opportunity to learn from the finest minds in the industry.

PLAY YOUR PART IN THE REVOLUTION

Do you want to get your voice heard? Do you want to help inspire and encourage a new generation of renewable energy? We thought so. It's time to submit a paper...

Submit a paper on any relevant green topic and deliver it to an audience of over 500 attendees.

HOW TO SUBMIT: Send papers to papers@cranfieldgreenalliance.co.uk no later than October 1, 2009!!

Good luck!

In this example we performed a number of steps to brighten our announcement. This included:

- Separating out the key parts of the message into separate headings and paragraphs.
- Converting the language to be more rabble-rousing and inspiring.
- Engaging in a rhetorical dialog with the reader by asking questions and clearly showing that the answer was to submit a paper, which is the very aim of the announcement.

Your announcement page should pass the elevator test: it should get the reader up to speed with what you are announcing within a minute.

Let's get back to our 5-A-Day example. When we were constructing the 5-A-Day announcement, we produced a page that identified the primary concept of 5-A-Day, how people could get involved, and what they needed to do. Each of the different pieces of information had individual headings, and emphasis was used extensively. View the page at <https://wiki.ubuntu.com/5-A-Day> to see the principles in this section in action—and with a successful outcome.

Review

Postmortems are hugely valuable in any kind of work, and if you don't perform them, you *never learn how to improve*. Whether you are evaluating how well you handled a discussion, cooked a meal, taught your kids how to play football, or built community buzz, a review can uncover great opportunities for improvement:

Efficiency

When you review your work, it gives you an opportunity to identify areas that are inefficient and redundant. You can use these as a basis for improvement.

Feedback

Gathering feedback from the people who consumed your buzz is a great way to see what they felt worked and what didn't. This is a great opportunity to get feedback on your writing, structure, and the other concepts throughout this chapter.

Ideas

When any kind of postmortem of an approach occurs, it almost always generates new ideas. These will help future buzz cycles to be more effective.

In the review phase, revisit the questions you asked in the planning phase and compare the plan with what happened:

- Did you stick to your aim and communicate it well to others?
- Did you identify the right outcomes to achieve?
- Were your chosen mediums the most suitable?
- Did you prepare effectively?
- Did you communicate well to others involved in the buzz about what needed to be done?
- Was your timeline for the buzz cycle accurate? Did you hit your deadlines?

To make this process effective, you should gather feedback from members of your community. Seek to gather responses from those who will provide you with constructive advice of what worked and what didn't. Remember, much of the goal here is to identify flaws in your approach. Flaws are nothing to be embarrassed about: they are opportunities to do even better next time.

Later in the book, we are going to talk about measuring community, and we will be able to use many of the techniques there to help us review our work.

Buzz Targets

Buzz needs a target, and that target is the topic you are focusing on. Each time you steal away someone's attention, she needs to know that it was worth it. You want to ensure that when someone looks in your direction, she feels it was worthwhile.

To do this, you need to decide what you want to promote. Of course, buzz is an ongoing process: you will need to bring attention and focus to your community many, many times. Each of these times you need to ensure there is a purpose. Whether the purpose is announcing the community, attracting new members, or anything else, you should ensure that your goals and intentions are clear.

Two kinds of buzz campaigns are useful in virtually all communities:

- The initial announcement.
- Ongoing efforts to attract members.

We are going to explore both of these, looking at the four elements of the buzz cycle.

Announcing Your Community

The very first time you need buzz is when you announce your community. The goal is to get the message out among people who can contribute to your community, pique people's interest, and get them to learn more.

Your announcement should appear fresh and exciting, and an effective buildup phase is particularly important. Earlier I gave examples of the approach to announcing *The Art of Community* and 5-A-Day; you should consider similar approaches.

Multimedia can make an announcement more exciting and memorable. Lawrence Lessig launched his Change Congress (<http://change-congress.org/>) campaign on his blog (http://www.lessig.org/blog/2008/03/change_congress_launched.html) by recording a short online presentation in which he narrated the goals of the project. I used a similar technique when I announced my Severed Fifth (<http://www.severedfifth.com/>) project. I recorded an announcement (<http://www.severedfifth.com/news/2008/06/severed-fifth-launched/>) and put it online on the day of release. These approaches really help captivate the viewer.

The desired outcome with this kind of buzz is to have people visit your website and to spread awareness of your community.

Applying the buzz cycle

Preparation

Ensure you have your website in place, and that all of the key information about your community and how to get involved is available. You should also ensure syndication feeds are available. Decide where it's important to get mentioned (websites, magazines, personal blogs by leaders in the field, and so forth). You can often source a list of places by asking your community and identifying related websites and magazines.

Buildup

If you have a preexisting blog or other site where you can post content, you could post "Coming Soon..." messages. If you are setting up a local community, you could put up fliers with the date of the announcement and a web address.

Announcement

On the date of the announcement, you should publicize in all the communication channels that make sense. You should provide a short blurb that inspires people to learn about your community and encourage them to visit your website.

Review

You can see where your announcement spread to and whether you were publicized in all the places you hoped. There's also qualitative feedback: did comments and questions show that you described your project clearly? Did the types of people you want respond?

Attracting Contributors

Contributors are at the forefront of what makes a great community. They are not only on the front line furthering your community in the direction of its goals, but they are also your representatives and spokespeople.

Although buzz campaigns can be started to attract contributors, this activity should be seen as an always-present and ongoing promotional effort. Your goal here is to constantly communicate the positive message of your community, its achievements, and how people can get involved.

The greatest communicators of this message are your existing members: you want to turn their satisfaction into active promotion for your community. To achieve this, your members need to feel proud to be in your community. They should feel a drive and passion for your goals and objectives, and they should feel that they want to spread the word so others can enjoy the community too. A positive community will always generate a positive message and be a magnet for new contributors.

The first step in achieving this is to build a sense of enjoyment, ease of contribution, and pride in your members. You build this by combining the elements discussed in this book: simple processes, effective governance, transparency, and so on. When you get these core attributes

right, your members will thrive on the opportunities and direction that your community offers them. You now need to encourage them to share their happiness and drive with others.

Their own resources and social network are an excellent communication channel for this outreach. Your job is to identify methods via which you can help them use these resources to spread the word about (a) the good work your community performs, and (b) why they enjoy being part of it.

For the former, give them buttons for their websites and blogs. Give them posters to print out and put in libraries, in stores, and on lampposts. Provide them with email signatures that they can use. Encourage them to set up Facebook/MySpace pages and more. Each of these resources should direct people back to the community's website.

To encourage your members to share their joy of being a part of the community, the key is that the communication focuses on the personal story: you need to encourage your members to share what they specifically enjoy about the project. In doing this you resort to the essence of community that we discussed back at the beginning of the book: *stories*.

Stories are a fantastic viral marketing asset. A great story is never told once; it is shared again and again. If your community members share great stories about their involvement in the community, the stories will travel far and wide and encourage new and unknown people to dip their feet into your waters.

You should talk about the importance of sharing stories with your members. Help them to understand that on any given day they could talk to someone online, in a coffee shop, or on a train or plane and potentially inspire someone to join the community. This can provide your members with a powerful sense of opportunity for bringing people in and will get them involved. You should now augment this discussion with some specific recommendations of viral approaches of getting the word out there:

Blogging

Blog entries get read, linked, and passed on across the Internet. They are easy to create, accessible to all, and are permanently archived in search engines and often crop up in random searches. Blog entries are also very gratifying for the author, particularly if the readers leave comments.

Microblogging

Earlier we discussed tools such as Twitter, identi.ca, and Facebook as excellent methods of sharing experiences: encourage your members to use these facilities as they do their work.

Word of mouth

Encourage your members to strike up conversations about your community in every possible scenario. Glorify the most insane and ridiculous cases in which a story is told and the recipient joins the community. As an example, one time I met a guy on the London Underground and told him about Ubuntu. He visited the website and eventually joined

and participated in the community. This was incredibly satisfying. Share these experiences, and encourage and celebrate them.

Interviews

Some of your community members may have the opportunity to be interviewed on websites, podcasts, videocasts, or in magazines. These are harder to come by, but encourage your members to ask these publications if they can be interviewed. If you don't ask, you don't get!

Conference presentations

If you have members who are keen to speak at conferences, encourage them to submit papers. If you have some experience of this process, you should offer them help and advice on putting together a submission.

Meetings/events/open days

You should encourage your members to organize meetings and events in which they can tell their story about the community. When I first got involved in open source, I organized presentations and open events at my university to help others understand how fun and satisfying our community is. All it needed was a room and a projector, and planting the idea in the minds of your members is sure to inspire some to organize an event.

An important element in building buzz to attract contributors is to showcase great work. I used many of these techniques when I started Severed Fifth and provided a range of website buttons and Severed Fifth posters (many of which were produced by the community). To generate buzz, I organized a campaign for fans to put the posters and stickers up in their local area. As part of the campaign, I encouraged typical destinations for the posters such as music shops, notice boards, and lampposts, but also showcased some of the wackier places. I saw examples of Severed Fifth stickers and posters in fish and chip shops, on the London Underground, in railway stations, toilet stalls, concert venues, buses, and even stuck to someone who was sleeping. As I heard these stories, I blogged them and encouraged fans to send me photos that I could put on the blog. The viral nature of the campaign encouraged more people to participate.

This viral marketing approach to building buzz has become the new way of doing business on the Internet. The idea is simple: you build buzz and encourage the consumers of your buzz to also build their buzz on the same topic. With this approach, when you unleash something on this network of viral volunteers, it spreads like wildfire. The key here is having this network available, and building that network can require a tremendous amount of energy in helping people to feel engaged, but when they do it will pay dividends in buzz. The key is in making people feel a sense of empowerment and responsibility to spread the message.

An interesting project that really set the standard for this kind of outreach was the Mozilla Firefox promotional campaign, Spread Firefox (<http://www.spreadfirefox.com/>).

Back in November 2004, the SilverOrange Canadian web firm was commissioned to build Mozilla's website. As part of their work they produced an evangelist application on their intranet to manage the structure and content of the site.

Blake Ross (one of the forefathers of Firefox) conceived the idea that Mozilla should encourage and inspire the global Firefox community to lead the marketing for the launch of the popular browser. One of the people involved in this work was Chris Messina. At the time, Chris was a Firefox community member, keen to see the project get better recognition and more widespread focus. Eventually he would go on to lead the Spread Firefox community marketing project in raising over \$220,000 in micro-donations to launch Firefox to a worldwide audience with an ad in the *New York Times*. Chris remembers the formation of the project well:

Originally there were probably about 30 of us in this private intranet, but maybe only 10 of us participated in any regular capacity. For me, this kind of work was all new to me—both open source and this kind of semi-anonymous Internet collaboration. It's not like I'd met anyone on the project personally—in fact, I only happened to find out about it because Steven Garrity had blogged that Mozilla was looking for volunteer designers.

After hearing about the project, Chris joined and applied his passion for Firefox to the campaign. At the heart of buzz is the ability to think outside the box to spread the work, and Chris remembers the approaches they used intimately:

I think there were a number of important elements of this, and that was that we made it fun to get involved. There was both a spirit of camaraderie and of shared purpose (fighting Microsoft), and with that in mind, people came up with some pretty clever ideas in the forums, contributing concepts, strategies, designs... telling the story of how Firefox made a difference to them. We worked hard to promote these efforts through things like the leaderboard (which measured the week-to-week growth in downloads from different affiliate links) and had, I believe, weekly contests or initiatives. Probably the most effective tool was the cumulative download counter... every time we hit a new milestone I would design new artwork to commemorate our success—with each design getting more and more insane.

The efforts of the Spread Firefox team were exceptional: Firefox 3 was downloaded over 28 million times in 24 hours when it was released. The project has gone on to secure a global user base and a reputation for quality, and a thriving and active community that surrounds it.

NOTE

Part of the responsibility of finding members is also going to involve finding leaders. We will discuss this [Chapter 8](#).

Applying the buzz cycle

Preparation

You should fully research which media you want to spread your buzz to. Your aim here is to identify the kind of personality that will be interested in joining your community, and to target the media that they read.

Buildup

I would not recommend any buildup to this target. You want to get straight out there and grab contributors.

Announcement

The announcement should take place in a variety of media. Your aim here is to share and inspire people in the achievements and accessibility of your community. Sell them on the evidence: show them third-party statements and material that firmly demonstrates that your community is a fun and rewarding place to be.

Review

Naturally, one measure of success is how many new people sign up or start helping out on committees. You can also try to see how many existing members helped the buzz with their personal statements, and why they were or were not comfortable doing so.

Building Alliances

Good communication resources and contacts are critical for promotional ideas and concepts to flow from you to the outside world. If you are going to build some great buzz, you need to know how to communicate effectively in different channels. This requires two skills, which must be mastered separately for each channel:

- Find the opportunity to make use of that channel. As an example, if you want to be featured in a particular magazine, you need to create an opportunity in which you can get your content there. This almost always involves building contacts. You should get to know the editors of the magazine and build a relationship that could allow you to feature some content in their channel.
- Ensure that your buzz in that channel is appropriate. The norms of communication between different mediums vary, but the differences can often be subtle and unwritten. If you act outside the expected boundaries (particularly in volunteer channels), it can impact negatively on your community.

NOTE

Also don't forget the old saying of "it's not what you know, it's who you know." If you have contacts who can help get your message out there or who can put you in touch with those who can help, go ahead and ask.

Buzz is designed to be consumed by lots of people. You want as much focus and attention on your community as possible. The more relevant eyeballs, the better.

As part of your planning stage for a buzz campaign, you should weigh the amount of effort involved with the number of relevant eyeballs. You want to ensure that your time and effort preparing materials and content is worthwhile and that a reasonable number of people see your work.

Much of this boils down to readership, and readership varies tremendously. Sometimes you can ask for this information, such as with magazines, but sometimes it is more of a guess. There will be some resources that you will assume have a large audience (such as a very popular website) and some with less (such as a blog).

An important consideration in this area is how the growth of the Internet has changed audience figures. It used to be general wisdom that paper publications were always the source of high audience figures. This is often no longer the case, as many websites—even a number of blogs—have hundreds of thousands of regular visitors.

The Professional Press

The professional press is a large and extensive channel. It encompasses magazines, websites, journals, videos, multimedia content, and more. Each of these publications has professional paid staff who have a responsibility to publish quality content.

The professional press has three primary concerns at the forefront of its mind:

Quality content

First and foremost, the professional press wants to produce leading content. It wants well-produced content that is of interest to its audience. Great content drives an increased...

Readership/audience

Professional publications rely on readership numbers. It is these numbers that largely justify the continuation of the publication. Having high audience numbers depends on getting the previous item in place: quality content.

Advertising opportunities

Most publications make a significant chunk of their revenue from advertising, and advertising does have an impact on content. Although many publications would deny it, advertising deals are often agreed based upon relationships between the publication and the company. These relationships need to be maintained to continue to bring in revenue. In many cases the content in a publication may be heavily critical of a company, product, or initiative. Although this *should* never matter, for many publications it does, and the producer of the content is either advised to change the content or focus on other topics.

You should factor these attributes into your plan for building buzz. You want to target the most appropriate publications that are relevant to your community. You will need to provide them with quality content that is of interest to their audience and consider any potential advertising conflict.

You will need to build a relationship with the publication. With these publications largely staffed by paid personnel, it is entirely reasonable to formally contact them via email or phone and ask them if you can contribute some content. A great first step would be to ask if they could feature your community in their news section. In some cases you may have the chance to build some relationships that you can return to when opportunity strikes later.

The first time I experienced this was years back at the start of my career. It was my first time at the Linux Expo in London. I was there running an exhibition stand for the first time for the KDE project. While there, I went to an after-show party, and the editors from *Linux Format* magazine were there. I got to chatting to them, had more than a few drinks, and a little while later asked them if they would consider publishing something from me. Nick Veitch, editor at the time, responded with, “Sure, write something, but if it is rubbish we won’t publish it.” I wrote my article, it got published, and so started my journalism career.

Linux Format opened many doors for me, but most importantly, it gave me a platform to talk about things that I considered interesting. It opened up a set of opportunities that have since helped with building buzz and promotion in the open source projects that I have been involved in.

Though it’s been some years since my days writing for *Linux Format*, I got in touch with then staff writer and now editor-in-chief Paul Hudson to gather some insight from the perspective of an editor to share with you all. Paul is a firm believer of the have-a-go approach to getting content in:

Both of us got into the world of free software journalism by saying, “hey, why don’t I write for you?” and I think that same situation occurs a lot—people don’t realise how much they can contribute until they just ask.

I think people imagine some sort of incredible vetting process must take place in order to write for magazines—as if only people in smoking jackets with PhDs from the school of ignorant snobbery are able to get stuck into writing, but that’s simply not true. Well, not *always* true, at least! Technical magazines and websites are crying out for people to get involved and just share what’s cool and what’s new in their world.

Paul regularly handles a slew of wannabe writers and passionate community members keen to get their projects featured in the magazine. With this in mind, he offers some useful guidance for improving the likelihood of getting coverage in magazines:

Don’t use email. We get stacks of emails, and most of them remain unread. The reason for this is that PR agencies blast us with all sorts of emails about things whether they are relevant to the magazine or not, so inevitably some important emails get lost in the mess.

Instead, call first, ask to speak to the news editor or someone else on the team, and just have a chat to them. They want good contacts as much as you do, so if you're someone who represents a project that's on their radar, they would love to be in touch with you. They are also much more likely to read your emails if you've already made contact by phone.

When you write release announcements, make it really clear what's new. This is something the GNOME project, as one such example, does well (<http://library.gnome.org/misc/release-notes/2.24/#rnusers>). They list the new features with pictures, so that someone can decide at a glance whether it's worth looking into.

If you are a software project, provide at least one screenshot that shows off the best feature you've got to offer. Remember, these guys are looking for "wow" things to print, and if you can send them a shot of your software looking awesome, they are much more inclined to run it as a news story.

Remember that even in technical magazines, some people are still journalists first and geeks second. Put your documentation online and link to all the technical information you like, but when you're trying to get a journalist interested in what you have to say, it's much more important to say "MyProject 2.0 uses 25% less RAM than MyProject 1.0" than to say "The switch to the xyz toolkit blah blah blah please send me straight to your Trash folder." Sure, drop in all the technical information you want later on, but you need to win them over in the first two sentences by focusing on what really kicks ass in your software.

If you're not producing software, getting into magazines is slightly trickier, because magazines rarely want to print a story if it's similar to something else they ran recently. So if your user group wants to get featured, you need to step outside the installfest (unless it's *big*) and do something pretty darn special. Whatever you do, take a photo and make it available under a Creative Commons license that allows commercial use.

The rules change with nontechnical magazines because once you enter the mainstream, you need to focus more on people. The *New York Times* won't find the Gecko web rendering engine interesting, but it will find Spread Firefox interesting because grassroots marketing really is changing the browser landscape.

While Paul offers some useful advice on the best-practice methods of getting content in the hands of editors, he is keen to emphasize that many communities simply don't get out and try, and this makes for a huge opportunity for printed nirvana:

Let me try to make this a bit clearer with a specific example from *Linux Format*. We run a page of LUG information every month, and we have to email people to try to get content to fill those pages, despite printing an open plea every issue asking people to get in touch.

So it's not that community members are struggling to get their information in—it's more that many of them just aren't trying. Perhaps they think we're not interested. Perhaps they think we won't print it. But as they so rarely try, most of them will never know.

Maybe they're just targeting magazines that are just a little bit out of their reach, but that's another schoolboy error—Editor X is much more likely to print an article about your community if Editors Y and Z already have. So start small; find a magazine that fits your niche closely and get yourself covered in there. Then use that to help get coverage in other places, building it up bit by bit.

The professional press can seem a bit unnerving. Professional journalists often feel like a live-by-the-seat-of-your-pants collection of hard-working, focused, and unrelenting writers. Don't let this worry you. Journalists are good people and they get asked for content opportunities all the time. Just go out there and ask.

When I started doing this, I would ask everyone. I would email 10 or 15 magazines to see if I could contribute content. I would not spam them: each email would be focused on that specific publication, and each would be relevant to my topic.

I would recommend that you email over a list of topics that you can write about and ask if you could write something about those topics. Alternatively, write an article and submit it. The benefit of the latter is that the journo has direct access to content, which is often an attractive proposition. Just go out there and ask; there really is no harm.

The Amateur Press

In the last five years, the amateur press world has exploded. The Internet has provided an incredible medium in which anyone can write about anything and have the chance to grow an audience. Technology and open access to information have provided an incredible opportunity to be heard, and many have built new reputations out of these opportunities. Consequently, millions of blogs and thousands of podcasts have sprung up around the world.

The amateur press is a world largely fueled by volunteers. The authors write their words not to claim a paycheck, but to share their ideas, perspectives, and opinions. Although populated by amateur scribes, this does not necessarily equate to a lack of quality. Some of the greatest work I have ever read has shown up on a blog. This could be the musings of Lessig on the copyright wars (<http://www.lessig.org/blog/>) or the deeply amusing yet incredibly well-written and inspired political blather of Flyingrodent (<http://flyingrodent.blogspot.com/>). Both are inspired works, yet very different in content and presentation.

The timeline of the amateur press revolution largely matched that of the professional press revolution many years earlier. The publishing world exploded onto the scene and thousands of books, newspapers, and journals sprung up. Each of these publications had its own perspective on its respective topic, and it became very difficult for readers to identify where the real quality was. The solution to this was the launch of other publications that read, reviewed, and collated this content (a great example being *The Week* at <http://www.theweek.com/>).

We have started to see much of this in recent years with blogs and podcasts. Websites such as Technorati (<http://technorati.com/>) have been sifting through the blogosphere and identifying the most popular and interesting blogs. Well-respected members of given communities will also often provide their “blogroll,” which lists the blogs that they enjoy reading. These resources all provide an excellent opportunity to identify which blogs you should be focusing on.

The amateur press is hugely important for buzz. The professional press is far more complicated and restricted in terms of getting heard, whereas often a few emails exchanged with the amateur press will net great results.

It’s not surprising why:

- The amateur press has a shared appreciation for volunteer community. They understand your reasons and intentions and will often want to promote them.
- There is no advertising conflict in most of the amateur press: they can write about whatever they want.
- There is typically no limitation on content. On a blog you can write as many posts as you like. This opens up more opportunities for you to get in on the content.

The most significant mediums in the amateur press arena are blogs, podcasts, and videos. Let’s take a quick look at all three and explore their cultures.

Blogs

Weblogs (typically known as blogs) started out life as online diaries. In them people would share what they were doing, what they were thinking, and what interested them. When blogging started, there were few blogs, and most were devoted to deeply technical topics.

Alan Cox was one of the earliest bloggers that I am aware of. Living in Swansea in Wales, Cox developed his celebrity among early Linux fans due to his work on the Linux kernel. Cox worked on incredibly low-level deep and dirty programming. It was about as unrelentingly hardcore as you could get.

When I first started reading his diary, I was fascinated. This was not the work of Alan Cox communicated through a journalist’s eyes. What I was reading were the direct thoughts of the man himself. Without wishing to sound like an overenthusiastic psychology major, I felt like I was actually closer to the person I was reading about. It gave a direct line to his world, and it pretty much rocked mine.

Since then blogging has expanded somewhat. In addition to blogs being used as personal diaries, many are now referred to as *personal publishing systems*. Many people, myself included, instead use blogs as a means of writing articles that are of interest to them. I use my blog to write about community, music, technology, usability, and more. I also use it as a medium to express achievements, goals, and more.

It is entirely conceivable that both your existing community members and the people you want to have as friends have blogs. With this in mind, blogging should be a critical component in your buzz-building.

The first task is knowing which blogs to build buzz with. Look for relevant blogs and strike up a relationship with the authors. Explain what your community is doing, and what your goals are. Try to get the author on board with your mission. You can then ask whether the author would be interested in sharing your work on his blog. If you have your own blog, you could offer to provide a link to his blog in exchange.

Blog wars. Although blogging has had a hugely positive impact on how people can articulate and share opinions and perspectives, there has been a dark side. Blogging has also become a medium in which much overzealous opinion can sometimes be expressed a little *too* quickly. Unfortunately, I have a rather embarrassing example of someone who fell into this trap: yours truly.

First, a bit of background. There used to be a company called Lindows that made a version of Linux that shared many visual and operational similarities to Windows. Microsoft frowned at the name “Lindows,” and a fight started to change the name. Lindows initially resisted, but after mounting pressure, changed their name to Linspire.

Now to the issue. Let me take the liberty to explain in the words of the article itself:

Recently a chap named Andrew Betts decided to take the non-free elements out of Linspire and release the free parts as another Linspire-derived distribution called Freespire. This act of rereleasing distributions or code is certainly nothing new and is fully within the ethos of open source. In fact, many of the distributions we use today were derived from existing tools. Unfortunately, Linspire saw this as a problem and asked for the Freespire name to be changed.

Reading through the notice of the change, the language and flow of the words screams marketing to me. I am certainly not insinuating that Betts has been forced into writing the page, or that the Linspire marketing drones have written it and appended his name, but it certainly doesn't sound quite right to me. I would have expected something along the lines of “Freespire has been changed to Squiggle to avoid confusion with the Linspire product”, but this is not the case. Instead we are treated to choice marketing cuts such as “To help alleviate any confusion, I contacted Linspire and they made an extremely generous offer to us all”. Wow. What is this one-chance-in-a-lifetime-not-sold-in-stores offer? Luckily, he continues, ‘they want everyone who has been following my project to experience ‘the real’ Linspire, FOR FREE!!!’. Now, pray tell, how do we get this “real” version of the software “FOR FREE!!!”?

“For a limited time, they are making available a coupon code called ‘FREESPIRE’ that will give you a free digital copy of Linspire! Please visit <http://linspire.com/freespire> for details”. Oh...thanks.

I gave Linspire a pretty full-throated kick in the wedding vegetables in my blog entry. I told the story, objected to what I considered hypocrisy given their own battle with similar-sounding trademarks, and vented. I wish *Guitar Hero* had existed back then: it would have been a better use of my time.

I was wrong. My article was never going to achieve anything. Shortly after the article was published, then-CEO Kevin Carmony emailed me. He was not a happy bunny. His objection, and it was valid, was that I flew off the handle without checking in with him first. My blog entry was my first reaction. The happy conclusion to this story is that I apologized to Kevin, admitted to being a bit of an arse, and we have remained friends. In fact, a little while later I joined the Linspire Advisory Board shortly before I joined Canonical to work on Ubuntu. It's funny how things work out.

PRACTICE WHAT YOU PREACH

In this chapter we have discussed the important attributes in setting up a website and blog for your project and also how to build buzz using other people's blogs.

Importantly, *you personally* should have a blog. Use it as an opportunity to discuss your own personal interests and also to talk about your community.

Podcasts

Podcasts are audio shows that are distributed on the Internet. They typically have between one and four presenters, and they are often based around fairly specific topics. Many listeners use a special piece of software called a podcatcher to subscribe to a podcast so that when new episodes of a podcast are released, they are automatically downloaded to a media player such as an iPod. This is a fantastic way to keep listeners updated with new content.

A significant reason behind the success of podcasts is that they deliver interesting specialist content to the listener to fill those dull minutes traveling to work. Many podcasts include interviews, reviews, features, debates, and other content. They vary hugely in both audio and content quality, and some podcasts have netted thousands of listeners.

As I mentioned earlier in this book, I cofounded a podcast with some friends called LugRadio. The show was very specifically focused on open source and digital rights. It took a lighthearted and irreverent approach to the content, and we deliberately focused on making the content social, fun, and amusing. Each show presented a range of topics for discussion, and each of us would weigh in and share our thoughts, often resulting in raucous debate and discussion.

Podcasts are always looking for pointers to interesting content and announcements. You should email the presenters and explain what you are working on, and see if they would be interested in featuring your community on the show. If you manage to get a spot on a popular

podcast, it could bring a wealth of new blood to your community. Although you may feel a little funny about emailing the presenters out of the blue, go ahead! If you don't ask, you don't get.

When pitching to a podcast, the most important tip is that your tone should match that of the podcast. When we were doing LugRadio, we would often get offers for interviews and features, but often the tone would be right out of a Marketing 101 textbook. This not only demonstrated that the person making the offer had not listened to the show, but it was a red flag for boring, emotionless content that had no place on LugRadio. On the other hand, we also got offers of content that was fun, loose, and insightful, and these were snapped up instantly.

If you get accepted for an interview or to have your community featured, listen to a number of episodes of the podcast to get a feel for the tone. Use it as a guide, but don't be afraid to share your own personality: you have the opportunity to inspire people to join your community, so just be yourself within the context of the podcast.

Finally, always ensure you have a web address to point the listeners to. This will provide an option to feed them more information, and the link can be listed in the podcast's show notes. Ensure the website that the link points to is packed with content that's ready when the episode of the podcast is published.

NOTE

As a gesture to the makers of the podcast, it is highly recommended that you spread the word about the podcast episode that your community is featured in. You could do this on your website, in your community's communication channels, and on blogs. This will help build a strong relationship with the podcast, leaving the door open for future content and interviews.

Videos

Online video has become increasingly popular as the Internet has become faster and more accessible. Although a hefty Internet connection is required to suck said videos down onto your computer for viewing, the sheer popularity of services such as YouTube (<http://www.youtube.com>) and blip.tv (<http://blip.tv/>) has demonstrated that many do indulge in such audiovisual delights on the Internet.

While some of us may reminisce about the dark days of dial-up Internet access, it is important to remember that many parts of the world still rely on slower dial-up connections. For these folks, videos are simply not an option. As such, before you get too excited to step into the shoes of Steven Spielberg, you should consider how accessible videos are for your community. As an example, if you are reaching out to a community in a remote part of Africa, you may want to rely on another lower-bandwidth medium. In general, my recommendation is to make use of video, but not as a primary medium. Instead, use it to complement your other, more widely accessible resources.

By far the most popular video service at the time of writing is YouTube. The idea is simple: anyone can upload a video and anyone with a web browser equipped with Macromedia Flash can view it. YouTube opened the doors for anyone with a webcam or a cheap video camera to be able to create and publish online video. This has resulted in thousands of hours of freely accessible video hitting the Internet.

This is only part of the value of YouTube, though. Videos on YouTube are hugely discoverable: it is possible to upload a video and have thousands of people stumble across it. This happens because each video on YouTube also displays a list of videos that are related to the one being viewed. This feature alone hugely increases the likelihood of people finding your videos. To do this you need to ensure that you name and add keywords to describe the content of your video in a way that enhances the chances of a certain demographic of user being able to find it. As an example, if you are part of a mapping community, you might want to tag your video with the words “map,” “geography,” “geo,” “location,” and any specific regions that were featured in the video. It is stunning how many people will find your videos, and this is further bolstered by word of mouth and the simplicity of embedding videos in web pages.

Another hugely useful feature of YouTube are *channels*. These are home pages on YouTube that contain videos from a certain provider (such as an artist, actor, or your community). There are different types of channels on YouTube designed for different types of provider that have additional facilities such as custom logos, blog entries, and tour dates. A huge benefit of a channel is that people can subscribe to it and will be notified when you add a new video. This is an excellent way to keep people hooked into your videos.

YouTube channels are something we have used extensively in the Ubuntu community. As part of our ongoing efforts to educate and train developers in how to contribute to Ubuntu, we created the Ubuntu Developer Channel on YouTube at <http://www.youtube.com/ubuntudevelopers>. On the site, we uploaded tuition videos, developer interviews, and more. At each Ubuntu Developer Summit, we would interview attendees to get updates about their work on the next release and perform question and answer sessions with key community members. These videos were hugely successful, and many of them gathered thousands of views within weeks of going online. The channel has over 1,800 subscribers at the time of writing. YouTube is an excellent resource for delivering education and best practice, and I highly recommend you make use of it if you have the resources and time.

Another interesting option for video is live streaming. This is where you produce a live video-cast that people can view as it is being recorded at a scheduled time. Traditionally, live streaming has always been off-limits for most of us: the bandwidth requirements are so epic that it makes it too costly and impractical.

Fortunately there is another option in the form of Ustream (<http://www.ustream.tv>). The concept is neat: the video you record on your computer with your lower-bandwidth Internet connection is streamed to the Ustream server, and then your viewers connect there with

Ustream's oodles of bandwidth to show your video. This means that your viewers don't hammer your own Internet connection, and it puts streaming in the hands of us all.

Ustream not only provides a simple means of streaming video, but also includes other features, such as a live chat channel for the show, and recording, tagging, and syndication facilities. The live chat channel is particularly interesting: it provides an opportunity for viewers to interact with the presenter as the broadcast is happening. This means that a viewer could tune in and comment on the content, and the presenter can read the comment and repeat it in the broadcast.

This is something I first tried around the time I was wrapping this book. While experimenting with Ustream, I tested it by broadcasting live from my living room and posting the link to the videocast on Twitter and identi.ca. Within minutes I had 24 people viewing my entirely ad hoc and off-the-cuff broadcast. With my interest piqued, I decided to start performing a regular show called *At Home with Jono Bacon* (<http://www.ustream.tv/channel/at-home-with-jono-bacon>).

Whether you make use of prerecorded or live video, there are some nuggets of best practice that are useful to keep your viewers engaged in your content:

- Do your best to keep production values high. As an example, if you are recording the video with your laptop's webcam, consider buying an external microphone. Many of the built-in mics in laptops sound awful and distort easily. Ensure that the location the camera points at looks clean, uncluttered, and professional, and wear clothes that don't distract the viewer.
- Before you produce your video, make some notes about what you will discuss. The easiest way of doing this is to make a series of bullet points with the topics you want to feature. If you are nervous, you may want to write a script, but I would highly recommend that you don't: unscripted content that is well delivered is far more natural and engaging.
- If possible, have more than one presenter. Multiple presenters always make for more interesting shows because there is an opportunity to bounce off each other with conversation, spark up debate, or play specific roles (e.g., the teacher and the learner).
- When creating an educational program (such as a tuition video), consider embedding in the video the focal point of the tuition (e.g., the computer screen if a programming video) or slides. There are many free tools that can capture computer screen content to video to help with this, such as Screencast-o-matic (<http://www.screencast-o-matic.com/>), Wink (<http://www.debugmode.com/wink/>), and recordMyDesktop (<http://recordmydesktop.sourceforge.net/>).
- YouTube and Ustream allow you to put notes next to your video. This is an excellent place to list the topics you are covering in the video, provide links to websites, and credit those involved in the content and creation of the video.

- Consider the licensing of your content before you release it. I would always recommend that you license your video under a Creative Commons license (more information on this is at <http://www.creativecommons.org/>). You should also consider the license of third-party content. As an example, if you want to use the latest U2 tune in your video, you might not be able to legally use it, or if you can, you may need to cough up some royalties. Be very careful here: although it is tempting to just go ahead and use the song, many online video producers have been busted for copyright infringement. I would always recommend that you play it safe and only use properly licensed content for your needs.
- Finally, you should be aware that at the time of writing the Macromedia Flash plug-in that many video websites use (including YouTube and Ustream) is closed source. Some proponents of software freedom and open source may refuse to view those videos for this reason. If this is likely to be problematic, it is recommended you also provide access to your videos in an entirely free format, such as Ogg Theora (<http://theora.org/>).

We only have space here to delve into a few tasty morsels of best practice, so if you are interested in making videos it is recommended that you buy a dedicated book on the topic. This will help you get up and running quickly.

Summary

Buzz is a complex and wide-ranging topic and one that can be exercised in many different ways. Fundamentally, the art of building buzz is an individual striving for it. Some will keep the game simple and use email and blogs to build excitement, and some will organize more grandiose campaigns and events, some even cutting crop circles in the ground in the shape of a logo to get the word out. The extravagance of the buzz campaign is often directly linked to the extravagance of the personality behind it, but this is not a closed-door club in which only those born with a sense of adventure can build great buzz. Anyone can develop this sense for thinking outside the box.

At the heart of these approaches is thinking of ways in which you can attract eyeballs in your direction and then deliver a message when they have you in their sights. It is important to get the balance right here: buzz without substance is *hype*, and you should avoid that like the plague. Always ensure that the message that you send is just as compelling as the buzz approach that you used.

Finally, one last note about thinking outside the box: it is a risky business. You are going to make mistakes and end up frustrating some folks. When you do screw up, see it as a learning process. Pick yourself up and review what you did objectively. All mistakes are acceptable under the premise that you learn from them. As you continue to build more and more buzz, you will make fewer mistakes and create more successes as you bring people to your community.

Measuring Community

“Learning without thought is labor lost.”

—Confucius

GREAT COMMUNITY LEADERSHIP REQUIRES accepting the volatility of community. Volunteer communities are a bubbling pot of varying personalities, commitments, skills, and experiences. It is this reason why I often refer to the work of community leaders as “herding cats.” But as leaders we are here not only to lead and inspire our cats, but also to learn to see the patterns in the chaos that is community.

There are many patterns out there: common personality traits, techniques for getting people excited about something, methods of handling conflict, common opinions, and more. It is these patterns, correlations, and structures that help us to not only better manage our own communities, but to share our experiences with others. When I see a pattern, I want to share it so you can use it in your own communities. Unfortunately, the randomness of the chaos can sometimes hide these patterns from common view, and many perceive “community measurement” as something of an oxymoron.

Earlier in this book we contrasted logical vessels such as programming languages and more randomized entities such as community:

[Programming languages] live and breathe in a world where the answer to a question is *yes* or *no*. There is no *maybe*. In a world where *maybe* does not exist, you can plan ahead for an answer. With community, the importance and diversity of the question are equally essential.

You could be forgiven for thinking that community is unpredictable and immeasurable, but let's not overemphasize the challenges of "maybe." Community may not be as straightforward as "yes" and "no," but that doesn't mean that we can't see the patterns, learn them, and share them.

Our strategic plan has helped us lay out a blueprint for what our community can achieve. But a plan is just that: a plan. It is a statement of work that we *intend* to do. It is not enough to simply provide your community with a strategic plan, even if it was collaboratively produced. Your community is going to need help, assistance, and some gentle nudging to achieve these goals.

Great community leadership requires regular and consistent feedback. When we originally produced our strategy, we were conscious of creating a feedback loop to gather input from our community. We now need to enforce the same feedback loop when it comes to the execution of the items in that plan. To really know we are achieving our goals, we need to be able to *measure* our community effectively.

Community Self-Reflection

People can be broadly divided into three groups that define their approach to tasks.

The first group believes that they know best. They go after their goals with little or no input from anyone else. They have a clear idea in their minds of what needs to be done, and they do it. They don't solicit feedback, because they don't need feedback: they know best. They are confident, if a little cocky at times.

The second group wouldn't know a decision if it hit them in the face. They need extensive help and guidance to not only flesh out how to achieve their goals, but also need coaching in the individual steps involved. If they had their way, they would ask you to do it for them. In the absence of delegation they instead want you to advise, make decisions, and generally think for them.

I am a fan of neither cocky nor procrastinating people. I am, however, a *huge* fan of the third category of people: those who are confident in their approach, but reinforce and improve said approach with feedback, mentoring, and guidance. Interestingly, many of the members of the first group, shun this approach as a perceived admission of imperfection. Well, I hate to break this to you, folks, but none of us are perfect. The American singer and actress Eartha Kitt described it perfectly: "I am learning all the time. The tombstone will be my diploma."

The greatest leaders are always willing to listen and learn, and the most inspiring people I have worked with have all taken this approach. This is what I personally strive for, and I encourage you to do the same.

As we discussed earlier, community is very much a soft science. It is an art form that is improved and extended by sharing stories and experiences. It is these tales that extend our knowledge

and offer us insight. The greatest gift you can offer to a community is the willingness to listen. When leaders listen, their community talks and everyone feels engaged.

The Foundations of Feedback

There are lots of easy ways to measure communities—the number of members on a forum, the number of contributions to a shared project, and so forth—but it’s not so easy to find *meaningful* measurements. We want our measurements to feed into our interpretations of what we’re doing and to trigger changes that can further improve our work.

Unfortunately, many community leaders obsess a little too much with the act of *gathering information* as opposed to gathering *meaningful* information. The goal here is not to construct an enormous vacuum cleaner to suck every tiny detail of your community into a graph. The goal is instead to identify what we don’t know about our community and to use measurements as a means to understand those things better.

Measurements without meaning are simply annoying. Randomly sucking in statistics is intensely time consuming—not only for you, but also for the people who provide you with the input. Most of us reading this book will be building volunteer communities in which time is a precious substance. Don’t waste it.

Each time you engage with your contributors to gather feedback, there is an unwritten yet implicit social contract: as a result of the feedback they expect change—hopefully positive change. When positive change does not happen, frustration sets in. If your measurements have purpose and you are willing to make change based on those measurements, your community will be satisfied.

Defining Purpose

Earlier in this book we constructed our strategic plan, which included a vocabulary identifying the key features of our strategy. Let’s have a quick recap:

- We first created a *mission statement* that outlined the broad aims of the community.
- Based on this statement we produced a set of high-level *objectives*. These are the major achievements that together form our mission statement.
- For each objective we have a set of *goals*. Each goal is a near-term outcome that we want to achieve. When we complete all the goals in an objective, we can consider the objective achieved.
- For each goal we have a set of *actions*. When we complete all of the actions in a goal, we can consider the goal achieved.

As you can see, the different parts of the strategy are nested inside each other. They look something like this:

Mission Statement

Objective

Goal

Actions

Actions

Actions

Goal

Actions

Actions

Objective

Goal

Actions

Actions

Our goals are the target of our feedback. They are what we want to measure. They are the purpose and the reason for our work throughout this chapter.

Inside these goals we are going to build an extra feature into this hierarchy: *hooks* and *data*. These features will help us gather important feedback on how well we are achieving the goal.

With a clear set of goals, each containing meaningful measurement facilities, we will be able to take a snapshot of the internals of our community that shows us how our strategic plan is proceeding. If the measurements show a lack of progress, we may have to alter actions, goals, or even objectives.

Hooks 'n' Data

So far we've discussed the importance of gathering feedback and measurements from your community, and that the focal point is the goals that we decided on in our strategic plan. The next step is to build into each goal a feedback loop that can deliver information about our progress on the goal.

This feedback loop is composed of two components—*hooks* and *data*:

Hooks

A hook is a medium or resource in which we can slurp out useful information about our goal. As an example, if our goal was to reduce crime in a neighborhood, a hook could be local crime reports from the police. The reason I call them hooks is that they are the protruding access points in which we can display interesting information.

Data

If a hook is the medium that provides useful information, data is the information itself. Using our previous example of a goal to reduce crime in a neighborhood, the hook (local crime reports) could provide data such as “10 crimes this month.” The data is composed of two attributes, the data itself and the measurement unit. Again, the kind of unit can be used to feed a display (e.g., numerical units are great for graphs).

To help understand this further, let’s look at an example. In the Ubuntu community, my team has worked to help increase the number of people who become new developers. In our strategic plan we created an objective to increase the number of community developers and fleshed it out with goals for improving developer documentation, awareness, and education. Each goal had the expected set of actions. For us to effectively track progress on the objective, we needed data about developer growth.

Fortunately, we have access to a system called Launchpad (<http://www.launchpad.net>), which is where all Ubuntu developers do their work. This system was an enormous hook that we could use to extract data. To do this we gathered a range of types of data:

- The current number of developers (e.g., 50 developers).
- How long new contributions from prospective developers took to be mentored by existing developers (e.g., 1.4 weeks).
- How many of these new contributions are outstanding for mentoring (e.g., 23 developers).

Launchpad had all of this information available. Using some computer programs created by Daniel Holbach, we could extract the data. This allowed us to track not only the current number of developers but also how quickly progress was being made: we knew that if the number of developers was regularly growing, we were making progress.

We could also use this data to assess the primary tool that new developers use to participate in Ubuntu: the queue of new contributions to be mentored. When a new developer wants to contribute, she adds her contribution to this queue. Our existing developers then review the item, provide feedback, and if it is suitable, commit it.

By having data on the average time something sits on that queue as well as the number of outstanding items, we could (a) set reasonable expectations, and (b) ensure that that facility was working as well as possible.

In this example, Launchpad was a hook. Using it involved some specific knowledge of how to physically grab the data we needed from it. This required specialist knowledge: a script was written in Python that used the Launchpad API to gather the data, and then it was formatted in HTML to be viewed.

Launchpad was an obvious hook, but not the only one. Although Launchpad could provide excellent numbers, it could not give us personal perspectives and opinions. What were the thoughts, praise, concerns, and other views about our developer processes and how well they worked? More specifically, how easy was it to get approved as an Ubuntu developer?

To gather this feedback, our hook was a developer survey designed for prospective and new developers. We could direct this survey to another hook: the list of the most recently approved developers and their contact details. This group of people would be an excellent source of feedback, as they had just been through the developer approval process and it would be fresh in their minds.

With so many hooks available to communities, I obviously cannot cover the specific details of how to use them. This would turn *The Art of Community* into *War and Peace*—complete with tragic outcome (at least for the author). Fortunately, the specifics are not of interest, as all hooks can be broadly divided into three categories:

Statistics and automated data

Hooks in this category primarily deal with *numbers*, and numbers can be automatically manipulated into statistics.

Surveys and structured feedback

These hooks primarily deal with *words and sentences* and methods of gathering them.

Observational tests

These hooks are *visual observations* that can provide insight into how people use things.

Let's take a walk through the neighborhood of each of these hooks and learn a little more about them.

Statistics and Automated Data

People have a love/hate relationship with statistics. Gregg Easterbrook in *The New Republic* said, "Torture numbers, and they'll confess to anything." Despite the cynicism that surrounds statistics, they turn up insistently on television, in newspapers, on websites, and even in general pub and restaurant chitchat. The problem with the general presentation of statistics is that the numbers are often used to make the point itself instead of being an indicator of a wider conclusion.

Statistics are merely indicators. They are the metaphorical equivalent to the numbers and gauges on the dashboard of a car: no single reading can advise on the health of the car. The gauges, along with the sound of the car itself, the handling, look and feel, and smell of burning rubber all combine to give you an indication that your beloved motor may be under the weather.

Despite the butchered reputation of statistics, they *can* offer us valuable insight into the status quo of our community. Statistics can provide hard evidence of how aspects of your community are functioning.

Many hooks can deliver numerical data. A few examples:

- Forums and mailing lists can deliver the number of posts and number of members.
- Your website can deliver the number of visitors and downloads.

- Your meeting notes can deliver the number of participants and number of topics discussed.
- Your development tools can deliver the number of lines of code written, number of commits made to the source repository, and number of developers.
- Your wiki can deliver the number of users and number of pages.

For us to get the most out of statistics, we need to understand the mechanics of our community and which hooks can deliver data from those mechanics. We will discuss how to find hooks from these mechanics later in this chapter.

The risks of interpretation

Although statistics can provide compelling documentation of the current status quo of your community, they require skill to be interpreted properly. A great example of this is forum posts. Many online communities use discussion forums, the online message boards in which you can post messages to a common topic (known in forums parlance as a *thread*). Within most forums there is one statistic that everyone seems to have something of a love affair with: the total number of posts made by each user.

It's easy to see how people draw this conclusion. If you have three users, one with 2 posts, one with 200 posts and one with 2,000 posts, it's tempting to believe that the user with 2,000 posts has more insight, experience, and wisdom. Many forums leap aboard this perspective and provide labels based upon the number of posts. As an example, a forum could have these labels:

- 0–100 posts: New to the Forum
- 101–500 posts: On the Road to Greatness
- 501–1,050 posts: Regular Hero
- 1,501–3,000 posts: Dependable Legend
- 3,001+ posts: Expert Ninja

As an example, if I had 493 posts, this would give me the “On the Road to Greatness” label, but if I had 2,101 posts, I would have the “Dependable Legend” label. These labels and the number of posts statistic is great for pumping up the members, but it offers little insight in terms of quality.

Quantity is rarely an indicator of quality; if it were, spammers would be the definition of email quality. When you are gathering statistics, you will be regularly faced with a quantity versus quality issue, but always bear in mind that quality is determined by the specifics of an individual contribution as opposed to the amalgamated set of contributions. What quantity really teaches us is experience. No one can deny that someone with 1,000 forum posts has keen experience of the forum, but it doesn't necessarily reflect on the quality of his opinion and insight.

Plugging your stats into graphs

Stats with no presentation are merely a list of numbers. When articulated effectively though, statistics can exhibit the meaning that we strive for. This is where graphs come into play.

Graphs are an excellent method of displaying lots of numerical information and avoiding boring the pants off either (a) yourself, or (b) other people. Let's look at an example. Earlier we talked about a project to increase the number of community developers in Ubuntu, and one piece of data we gathered was the current number of community developers who had been approved. This is of course a useful piece of information, and as the number climbs it helps indicate that we are achieving our goals. What that single number does not teach us, though, is how *quickly* we are achieving our goal.

Imagine that we had 50 developers right now and we wanted to increase that figure by 20% a year. This would mean we would need to find five developers in the next six months. This works out at approximately one developer per month. If we want to encourage this consistency of growth, we need not only to look at the number of current developers once, but also track it over time so we can see if we are on track to achieve our 20% target. Using this example in the Ubuntu world, we could use Launchpad to take a regular snapshot of the number of current developers, plot it on a graph, and draw a line between the dots. This could give us a growth curve of new developers joining the project.

Another handy benefit of graphs is to show the impact of specific campaigns on your community. On my team at Canonical we have a graph that shows the current number of bugs in Ubuntu. On the graph is a line that shows the current number of bugs for each week. As you can imagine, the line that connects these numbers shows a general curve of our bug performance. This line is generally fairly consistent.

Each cycle, we have a special event called the Ubuntu Global Bug Jam (<http://wiki.ubuntu.com/UbuntuGlobalJam>) in which our community comes together to work on bugs. Our local user groups organize bug-squashing parties, and there are online events and other activities that are all based around fixing bugs. Interestingly, each time we do the event, we see a drop in the number of bugs on our graph for the days that the Global Bug Jam happens. This is an excellent method of assessing the impact of the event on our bug numbers.

TECHNICAL TIP

You may be wondering how you can gather data from various hooks and display them in a graph automatically. I just wanted to share a few tips. If this seems like rocket science to you, I recommend that you seek advice from someone who is familiar with these technologies.

Gathering data from hooks is hugely dependent on the hook. Fortunately, many online services offer an application programming interface (API) that can be used by a program to gather the data. This will require knowledge of programming. Many programming languages, such as Python and Perl, make it simple to get data through the API.

Another approach with hooks is to screen scrape. This is the act of downloading a web page and figuring out what text on the page has the data. This is useful if an API is not available.

For graphing, there are many tools available that can ease graphing if the data is available. This includes Cricket (<http://cricket.sourceforge.net/>), and of course you could load data into a spreadsheet with a comma-separated value (CSV) file if required.

Surveys and Structured Feedback

Surveys are an excellent method of taking the pulse of your community. For us, they are simple to set up, and for our audience, they are simple to use. I have used surveys extensively in my career, and each time they have provided me and my teams with excellent feedback. Over the next few pages I want to share some of the experience I have picked up in delivering effective surveys.

The first step is to determine the purpose of the survey. What do you want to achieve with it? What do you need to know? Every survey needs to have a purpose, and it is this purpose that will help you craft a useful set of questions that should generate an even more useful set of data.

NOTE

You should avoid surveys just for the purpose of creating a survey. Only ever create a survey if there is a question in your head that is unanswered. Surveys are tools to help you understand your community better: use them only when there is a purpose. Examples of this could include understanding the perception of a specific process, identifying common patterns in behavior in communication channels, and learning which resources are used more than others.

Again, your goals from your strategic plan are a key source of purpose for your surveys. As an example, if your goal is “increase the number of contributors in the community,” you should break down the workflow of how people join your community, and produce a set of questions that test each step in this workflow. You can use the feedback from the answers to gauge whether your workflow is effective and use the data as a basis for improvements.

Choosing questions

When deciding on questions, you should be conscious of one simple fact: everyone hates filling in surveys. When someone has considered participating in your survey, you need to be able to gather that person’s feedback as quickly and easily as possible. This should take no longer than five minutes. As such, I recommend you use no more than 10 questions. This will give the respondent an average of 30 seconds to answer each question.

The vast majority of surveys have questions with multiple-choice ratings for satisfaction. Most of you will be familiar with these: we are provided with a satisfaction scale between 1 (awful) and 5 (excellent). You are then expected to select the appropriate satisfaction grade for that question. Surveys like this are simple and effective.

THE VARIANCE OF THE VOTE

Ratings are a funny beast, and everyone interprets them differently.

A great example of this is the employee performance reviews that so many of us are familiar with. In one organization I have worked at, the scale ranged from 1 (unacceptable) to 5 (outstanding). I did a small straw poll of how different people interpreted the grading system, and the views varied tremendously:

- Some felt that if 1 is unacceptable and 5 is outstanding, then 3 would be considered acceptable, and if staff completed their work as contractually expected, a 3 would be a reasonable score.
- Some others felt that meeting contractually agreed upon standards would merit a 5 on the scale, and that 3 would indicate significant, if tolerable, lapses.
- Interestingly, some people informed me that they would never provide a 5, as they felt there was *always* room for improvement.

When people fill in your survey, you will get an equally varied set of expectations around the ratings. You should factor this variation of responses into your assessment of the results. One way to do this is to add up the responses from each person and increase or reduce them proportionally so each person's total adds up to the same points. But this may not be valid if someone legitimately had a wonderful or horrific experience across the board.

When writing your questions, you need to ensure that they are simple, short, and specific enough that your audience will not have any uncertainty about what you are asking. When people are confronted with unclear questions in surveys, they tend to simply give up or pick a random answer. Obviously both of these are less-than-stellar outcomes. Let's look at an example of a bad question:

Do you like our community?

Wow, how incredibly unspecific. Which aspect of the community are we asking about? What exactly does "like" mean? Here is an example of a much better question:

Did you receive enough help and assistance from the mailing list to help you join the community successfully?

This is more detailed, easier to understand, and therefore easier to answer. It's no coincidence that the results are more immediately applicable to making useful changes in the community.

Using the previous example of a survey to track progress on the goal of increasing the number of contributors, here are some additional example questions:

- How clear was the New Contributor Process to you?
- How suitable do you feel the requirements are to join the community?
- How useful was the available documentation for joining the community?
- How efficiently do you feel your application was tended to?

Each of these asks a specific question about your community and the different processes involved.

Showing off your survey reports

Earlier, when we talked about statistics, we also explored the benefits of using graphs for plotting numerical feedback. We could feed the data directly into the graph, and the findings are automatically generated. This makes the entire process of gathering statistics easy: we can automate the collection of the data from the hook (such as regularly sucking out the data) and then the presentation of the data (regularly generating the graph).

Unfortunately, this is impossible when dealing with feedback provided in words, sentences, and paragraphs. A person has to read and assess the findings and then present them in a report. It is this report that we can present to our community as a source for improving how we work.

Readers have priorities when picking up your report. No one wants to read through reams and reams of text to find a conclusion: they want to read the conclusion up front and optionally read the details later. I recommend that you structure your survey findings reports as follows:

1. Present a broad conclusion, a paragraph that outlines the primary revelation that we can take away from the entire survey. For example, this could be “developer growth is slower than expected and needs to improve.” It is this broad conclusion that will inspire people to read the survey. Do bear one important thing in mind, though: don't turn the conclusion into an inaccurate, feisty headline just for the purposes of encouraging people to read the survey. That will just annoy your readers and could lead to inaccurate buzz that spirals out of your control, both within and outside your community.
2. Document the primary findings as a series of bullet points. These findings don't necessarily need to be the findings for each question, but instead the primary lessons to be learned from the entire survey. It is these findings that your community will take as the meat of the survey. They should be clear, accurate, and concise.
3. You should present a list of recommended actions that will improve on each of the findings. Each of these actions should have a clear correlation with the findings that your survey presented. The reader should be able to clearly identify how an action will improve the current situation. One caveat, though: not all reports can present action items.

Sometimes a factual finding does not automatically suggest an action item; it may take negotiation and discussion for leaders to figure out the right action.

4. Finally, in the interest of completeness, you should present the entire set of data that you received in the survey. This is often useful as an addendum or appendix to the preceding information. This is a particularly useful place to present nonmultiple-choice answers (written responses).

When you have completed your survey and documented these results, you should ensure they are available to the rest of your community. Sharing these results with the community is (a) a valuable engagement in transparency, (b) a way of sharing the current status quo of the community with everyone, and (c) an opportunity to encourage others to fix the problems or seek the opportunities that the survey uncovers.

To do this, you should put the report on your website. Ensure you clearly label the date on which the results were taken. This will make it clear to your readers that the results were a snapshot of that point in the history of your community. If you don't put a date, your community will assume the results are from today.

When you put the results online, you should notify your community through whatever communication channels are in place, such as mailing lists, online chat channels, forums, websites, and more.

DOCUMENTED RESULTS ARE FOREVER

Before we move on, I just want to ensure we are on the same page (pun intended) about documenting your results.

When you put the results of your survey online, you should never go back and change them. Even if you work hard to improve the community, the results should be seen as a snapshot of your community.

You should ensure that you include with the results the date that they were taken so this is clear.

Observational Tests

When trying to measure the effectiveness of a process, an observational test can be one of the most valuable approaches. This is where you simply sit down and watch someone interact with something and make notes on what the person does. Often this can uncover nuances that can be improved or refined.

This is something that my team at Canonical has engaged in a number of times. As part of our work in refining how the community can connect bugs in Ubuntu to bugs that live upstream, I wanted to get a firm idea of the mechanics of how a user links one bug to another. I was

specifically keen to learn if there were any quirks in the process that we could ease. If we could flatten the process out a little, we could make it easier for the community to participate.

To do this, we sat down and watched a contributor working with bugs. We noted how he interacted with the bug tracker, what content he added, where he made mistakes, and other elements. This data gave us a solid idea of areas of redundancy in how he interacted with a community facility.

What Jorge on my team did here was user-based testing, more commonly known as *usability testing*. This is a user-centered design method that helps evaluate software by having real people use it and provide feedback. By simply sitting a few people in front of your software and having them try it out, usability testing can provide valuable feedback for a design before too much is invested in coding a bad solution.

Usability testing is important for two reasons. The most obvious is that it gets us feedback from a lot of real users, all doing the same thing. Even though we aren't necessarily looking for statistical significance, recognizing usage patterns can help the designer or developer begin thinking about how to solve the problem in a more usable way.

The second reason is that usability testing, when done early in the development cycle, can save a lot of community resources. Catching usability problems in the design phase can save development time normally lost to rewriting a bad component. Catching usability problems early in a release cycle can preempt bug submissions and save time triaging. This is on top of the added benefit that many users may never experience such usability issues, because they are caught and fixed so early.

Open source is a naturally user-centered community. We rely on user feedback to help test software and influence future development directions. A weakness of traditional usability testing is that it takes a lot of time to plan and conduct a formal laboratory test. With the highly iterative and aggressive release cycles some open source projects follow, it is sometimes difficult to provide a timely report on usability testing results. Some examples of projects that overcame problems in timing and cost appear in the accompanying sidebar ([“Examples of Low-Budget, Rigorous Usability Tests”](#)) by Celeste Lyn Paul, a senior interaction architect at User-Centered Design, Inc. She helps make software easier to use by understanding the user's work processes and designing interactive systems to fit their needs. She is also involved in open source software and leads the KDE Usability Project, mentors for the OpenUsability Season of Usability program, and serves on the Kubuntu Council.

EXAMPLES OF LOW-BUDGET, RIGOROUS USABILITY TESTS

There are some ways you can make usability testing work in the open source community. Throughout my career in open source, I have run a number of usability tests, and not all have been the conventional laboratory-based testing you often think of when you hear “usability test.” These three

examples help describe the different ways usability testing can be conducted and how it can fit into the open source community.

My first example is the usability testing of the Kubuntu version of Ubiquity, the Ubuntu installer. This usability test was organized as a graduate class activity at the University of Baltimore. I worked with the students to design a research plan, recruit participants, run the test, and analyze the results. Finally, all of the project reports were collated into a single report, which was presented to the Ubuntu community. The timing of the test was aligned with a recent release and development summit, and so even though the logistics of the usability test spanned several weeks, the results provided to the Ubuntu community were timely and relevant.

Although this is the more rare case of how to organize open source usability testing, involving university students in open source usability testing provides three key benefits. The open source project benefits from a more formal usability test, which is otherwise difficult to obtain; the university students get experience testing a real product, which looks good on a curriculum vitae; and the university students get exposure to open source, which could potentially lead to interest in further contribution in the future.

My second example involves guerilla-style usability testing over IRC. I was working with Konstantinos Smanis on the design and development of KGRUBEditor. Unlike most software, which usually are in the maintenance phase, we had the opportunity to design the application from scratch. While we were designing certain interactive components, we were unsure which of the two design options was the most intuitive. Konstantinos coded and packaged dummy prototypes of the two interactive methods while I recruited and interviewed several people on IRC, guiding them through the test scenario and recording their actions and feedback. The results we gathered from the impromptu testing helped us make a decision about which design to use.

The IRC testing provided a quick and dirty way of testing interface design ideas in an interactive prototype. However, this method was limited in the type of testing we could do and the amount of feedback we could collect. Remote usability testing provides the benefit of anytime, anywhere, anyone at the cost of high-bandwidth communication with the participant and control over the testing environment.

My final example is the case of usability testing with the DC Ubuntu Local Community (LoCo). I developed a short usability testing plan that had participants complete a small task that would take approximately 15 minutes to complete. LoCo members brought a friend or family member to the LoCo's Ubuntu lab at a local library. Before the testing sessions, I worked with the LoCo members and gave them some tips on how to take their guest through the test scenario. Then, each LoCo member led their guest through the scenario while I took notes about what the participant said and did. Afterward, the LoCo members discussed what they saw in testing, and with assistance, came up with a few key problems they found in the software.

The LoCo-based usability test was a great way to involve nontechnical members of the Ubuntu community and provide them an avenue to directly contribute. The drawback to this method is that it takes a lot of planning and coordination: I had to develop a testing plan that was short but provided

enough task to get useful data, find a place to test (we were lucky enough to already have an Ubuntu lab), and get enough LoCo members involved to make testing worthwhile.

—Celeste Lyn Paul
Senior Interaction Architect
User-Centered Design, Inc.

Although Celeste was largely testing end-user software, the approach that she took was very community-focused. The heart of her approach involved community collaboration, not only to highlight problems in the interface but also to identify better ways of approaching the same task.

These same tests should be made against your own community facilities. Consider some of the following topics for these kinds of observational tests:

- Ask a member to find something on your website.
- Ask a prospective contributor to join the community and find the resources they need.
- Ask a member to find a piece of information, such as a bug, message on a mailing list, or another resource.
- Ask a member to escalate an issue to a governance council.

Each of these different tasks will be interpreted and executed in different ways. By sitting down and watching your community performing these tasks, you will invariably find areas of improvement.

Measuring Mechanics

The lifeblood of communities, and particularly collaborative ones, is communication. It is the flow of conversation that builds healthy communities, but these conversations can and do stretch well beyond mere words and sentences. All communities have collaborative mechanics that define how people do things together. An example of this in software development communities is bugs. Bugs are the defects, problems, and other it-really-shouldn't-work-that-way annoyances that tend to infiltrate the software development process.

Every mechanic (method of collaborating) in your community is like a conveyor belt. There is a set of steps and elements that comprise the conversation. When we understand these steps in the conversation, we can often identify hooks that we can use to get data. With this data we can then make improvements to optimize the flow of conversation.

Let's look at our example of bugs to illustrate this.

Every bug has a lifeline, and that lifeline is broadly divided into three areas: *reporting*, *triaging*, and *fixing*. Each of these three areas has a series of steps involved. Let's look at reporting as an example. These are the steps:

1. The user experiences a problem with a piece of software.
2. The user visits a bug tracker in her web browser to report that problem.
3. The user enters a number of pieces of information: a summary, description, name of the software product, and other criteria.
4. When the bug is filed, the user can subscribe to the bug report and be notified of the changes to the bug.

Now let's look at each step again, see which hooks are available and what data we could pull out:

1. There are no hooks in this step.
2. When the user visits the bug tracker in her web browser, the bug tracker could provide data about the number of visitors, what browsers they are using, which operating systems they are on, and other web statistics.
3. We could query the bug tracker for anything that is present in a bug report: how many bugs are in the tracker, how many bugs are in each product, how many bugs are new, etc.
4. We could gather statistics about the number of subscribers for each and which bugs have the most subscribers.

So there's a huge range of possible hooks in just the bug-reporting part of the bug conveyor belt. Let's now follow the example through with the remaining two areas and their steps and hooks:

The following are the triaging steps:

1. A triager looks at a bug and changes the bug status.
2. The triager may need to ask for additional information about the bug.
3. Other triagers add their comments and additional information to help identify the cause of the bug.

Triaging hooks:

1. We could use the bug tracker to tell us how many bugs fall into each type of status. This could give us an excellent idea of not only how many bugs need fixing, but also, when we plot these figures on a graph, how quickly bugs are being fixed.
2. Here we can see how often triagers need to ask for further details. We could also perform a search of what kind of information is typically missing from bug reports so we can improve our bug reporting documentation.
3. The bug tracker can tell us many things here: how many typical responses are needed to fix a bug, which people are involved in the bug, and which organizations they are from (often shown in the email address, e.g., billg@microsoft.com).

Fixing steps:

1. A community member chooses a bug report in the system and fixes it. This involves changing and testing the code and generating a patch.
2. If the contributor has direct access to the source repository, he commits the patch. Otherwise, the patch is attached to the bug report.
3. The status of the bug is set to FIXED.

Fixing hooks:

1. There are no hooks in this step.
2. A useful data point is to count the number of patches either committed or attached to bug reports. Having the delta between these two figures is also useful: if you have many more attached patches, there may be a problem with how easily contributors can get commit access to the source repository.
3. When the status is changed, we can again assess the number changes and plot them on a timeline to identify the rate of bug fixes that are occurring.

In your community, you should sit down and break down the conveyor belt of each of the mechanics that forms your community. These could be bugs, patches, document collaboration, or otherwise. When you break down the process and identify the steps in the process and the hooks, this helps you take a peek inside your community.

Gathering General Perceptions

Psychologically speaking, perception is the process in which awareness is generated as a result of sensory information. When you walk into a room and your nose tells you something, your ears tell you something else, and your eyes tell still more, your brain puts the evidence together to produce a perception.

Perception occurs in community, too, but instead of physical senses providing the evidence, the day-to-day happenings of the community provide the input. When this evidence is gathered together, it can have a dramatic impact on how engaged and enabled people feel in that community.

Even in the closed and frightening world of a prison community, with its constant threat of random violence and tyranny, there are shared perceptions, interestingly between staff and prisoners. Professor Alison Liebling, a world expert on prisons, discovered common cause between staff and prisoners in her *Measuring the Quality of Prison Life* study, which took place between 2000 and 2001. Liebling invited staff and prisoners to reflect on their best rather than worst experiences and identified broad agreement between staff and prisoners on “what matters” in prison life. She discovered that “staff and prisoners produced the same set of dimensions, suggesting a moral consensus or shared vision of social order and how it might be achieved.” Her work provided a model that described and monitored that which previously

appeared impossible to measure: “respect, humanity, support, relationships, trust, and fairness,” which had remained hidden under the traditional radar of government accountability.

Perception plays a role in many communities, particularly those online. Some years back I was playing with a piece of software (that shall remain nameless). I spent quite some time setting it up and was more than aware of some of the quirks that were involved in its installation. In the interest of being a good citizen, I thought it could be useful to keep a notepad and scribble down some of the quirks, what I expected, and how the software did and did not meet my expectations. I thought that this would provide some useful real-world feedback about a genuine user installing and using the software.

I carefully gathered my notes and when I was done I wrote an email to the software community’s mailing list with my notes. I strived to be as constructive and proactive in my comments as possible: my aim here was not to annoy or insult, but to share and suggest.

And thus the onslaught began....

Email after email of short-tempered, antagonistic, and impatient responses came flowing in my general direction. It seemed that I struck a nerve. I was criticized for providing feedback on the most recent stable release and not the unreleased development code in the repository(!), many of my proposed solutions were shot down because they would “make the software too easy” (like that is a bad thing!), and the tone was generally defensive.

Strangely, I was not perturbed, and I still took an interest in the software and community, but as I dug deeper I found more issues. The developer source repository was very restrictive; the comments in bug reports were equally defensive and antagonistic; the website provided limited (and overtly terse information),;and the documentation had statements such as “if you don’t understand this, maybe you should go somewhere else.” Well, I did.

When each of these pieces of evidence combined in my brain, I developed a somewhat negative perception of the community. I felt it was rude, restrictive, cliquy, and unable to handle reasonably communicated constructive criticism.

It was perception that drove me to this conclusion, and it was perception that caused me to focus on another community in which my contributions would be more welcome and my life there would be generally happier.

Throughout the entire experience there was no explicit statement that the community was “rude, restrictive, cliquy, and unable to handle reasonably communicated constructive criticism.” This was never written, spoken of, or otherwise shared.

Measuring perception involves two focus points. On one hand you want to understand the perception of the people inside your community, but you also want to explore the perception of your community from the outside. This is particularly important for attracting new contributors.

To measure both kinds of perception, our hooks are people, and we need to have a series of conversations with different people inside and outside our projects to really understand how they feel. As an example, imagine you are a small software project and you have a development team, a documentation team, and a user community. You should spend some time having a social chitchat with a few members in each of those teams. This will help paint a picture for you.

Some of the most valuable feedback about perception can happen with so-called “corridor conversations.” These are informal, social, ad hoc conversations that often happen in bars, restaurants, and the corridors of conferences. These conversations typically have no agenda, there are no meeting notes, and they are not recorded. The informal nature of the conversation helps the community member to relax and share her thoughts with you.

Perception of you

Another important measurement criterion is the perception of you as a person. As a leader you are there to work with and represent your community. Your community will have a perception of you that will be shared among its members. You want to understand that perception and ensure it fairly reflects your efforts.

Perception of community leaders is complex, particularly when a leader works for a company to lead the community. As an example, as part of my current role at Canonical as the Ubuntu community manager I work extensively with our community in public, running public projects. There are, however, some internal activities that I focus on. I help the wider company work with the community. I work on Canonical projects that are currently under a Non-Disclosure Agreement (NDA). There is also the work I do with my own team, such as building strategy, reviewing objectives, conducting performance reviews, making weekly calls, and more. Many of these internal activities are never seen by the wider community, and as such the community may not be privy to the genuine work that helps the community but is not publicized.

Gathering feedback about your performance is hard work. It is difficult to gather constructive, honest, and frank feedback, because most people find it impossible to deliver that content to someone directly. Even if you are entirely open to feedback, you need to ensure that the people who are speaking to you feel there will be no repercussions if they offer criticism. You need to work hard to foster an atmosphere of “I welcome your thoughts on how I can improve.”

Due to the difficulty of gathering frank feedback, you may want to rely on email to gather it. When we have physical conversations or even discussions on the phone, body language, vocal tone, and enunciation make those conversations feel much more personal. The visceral connection may make it intimidating for your respondent to provide frank and honest feedback (particularly if that involves criticism). Email removes these attributes in the conversation, and this can make gathering this feedback easier.

TRANSPARENCY IN PERSONAL FEEDBACK

In the continuing interest of building transparency, an excellent method is to be entirely public in letting your community share their feedback about you.

As an example, you could write a blog entry asking for feedback and encouraging people to leave comments on the entry, and allow anonymous comments.

This is a tremendously open gesture toward your community. It could also be viewed as a tremendously risky gesture. There is a reasonable likelihood that someone could share some negative thoughts about you there, and others may agree. (But that's also feedback you need to collect!)

Anonymity and Privacy

The act of measuring community is an exercise in gathering information about other people and drawing conclusions. Some of this data will be generic to the community as a whole (such as statistics about mailing lists or forums) and some will be specific to an individual (such as a response to a survey). When data can be directly linked to an individual, the subject of anonymity and privacy steps into view. Although at first these two topics may seem like they could potentially cause problems if you put your foot down wrong, their guidelines are simple. Let's talk about them both now.

Anonymity

Anonymity is a valuable tool when gathering feedback, particularly around contentious topics. If you are gathering feedback about a particular governance body in your community, many people may feel uncomfortable with associating their views with their identity. For this reason, anonymous feedback can be a useful option.

There is a dark side to anonymity, though. The Internet has long proved that when identity can be hidden or obscured, all manner of whack jobs and nutcases want to be your friends or, rather, your enemies. In the world of the Internet, the quote from the 1988 movie *The Dead Pool* really resonates: "Opinions are like assholes. Everybody's got one and everyone thinks everyone else's stinks." If you open an avenue for people to share their views online...expect anything and everything.

Anonymity creates two major risks. First, when identity is hidden, some jump at the opportunity to be rude, arrogant, and sometimes outright offensive. These people are often referred to as *trolls* in online communities. The second, subtler problem is that anonymity will sometimes cause people to overstate their concern with an issue: they will often dial up the

annoyance factor to 11. This means that you get a misrepresented perspective, and this can skew your aim of getting genuinely representative views from the community.

With this we have a difficult balance: there is value in anonymous feedback, but there is a significant risk of trolling and overstated unrepresentative perspectives. How do we find a balance? A simple solution is to welcome anonymous data, but be cognizant of what it could represent. Therefore, when conducting your research, you should encourage a combination of feedback with identity attached in addition to anonymous feedback.

Consider the example of running a survey to assess the quality of experience of a contributor joining your community. I would recommend that you have two identical surveys: one is directed to the 10 most recent people who have joined your community, and the other open to anyone. When evaluating the results, treat the survey that you directed to particular people as the most valuable input, but still consider highly the results of your other survey. Combining the results of both surveys is likely to produce a balanced perspective.

Before we move on, I just want to dispel the myth of anonymity on the Internet. This all boils down to a simple rule:

No one is anonymous on the Internet.

No one.

(Yes, that includes you super-elite hacker types, too.)

In the online world it's tempting to believe that you are anonymous, but so-called "anonymity" is merely a carefully constructed set of abstractions that ultimately puts most people off trying to discover your identity. These barriers always have a trail, though, and if someone tries hard enough, he could break down the barriers of anonymity.

The value and risk of anonymity is hugely dependent on what kind of community you are building. If you are building a small local knitting community to meet, share patterns, and enjoy each other's company, it is unlikely that anyone is going to work too hard to break anonymity. If you are involved in a technical community based around security and hacking, some will see your anonymity as a challenge. In more technical communities, as well as communities dealing with sensitive issues in politics or health, you may have a harder time soliciting anonymous feedback for fear of others finding out.

Privacy

The middle ground between anonymity and full public disclosure is feedback provided with an identity under the proviso that it is kept private. Maintaining this level of privacy is an important consideration when handling anyone's information, and particularly important when handling sensitive information around conflict.

Privacy is sacred. It should never be comprised, and when you engage with someone who shares private information, you become responsible for that information. As such, you should ensure that you have a suitable means of securing that privacy. This does not need to include

super-technical encrypted emails and retina scans to get into your laptop, but simple processes that will ensure that your respondents have confidence in you keeping their information to yourself.

The most fundamental underlying step here is that people trust you. It doesn't matter what procedures you put in place to stop leaks; if people don't trust you as a warm body, they will not entrust their thoughts to you. As we discussed earlier, you should always build a sense of trust and confidence in your community. You should then build on this trust with methods of gathering feedback that are secure.

As an example, I was once performing an assessment of a governance body and wanted to gather feedback from each of the members on the body. I was keen for this feedback to be brutally blunt and honest, and to do this I made a few conscious decisions:

- To ensure privacy, I did not use a public resource such as a wiki or forum to ask for the feedback, but instead requested it to my private email address. This meant that all submissions came directly to me.
- I was explicit in the email that the information provided should be frank and honest and that the answers would be subject to absolute privacy. I made it clear that the feedback would not be shared either publicly, with the other members, or with other third parties.
- I sent the email to private email addresses, not an email address associated with the commercial sponsor. This would remove any conspiracy-theory worries of someone snooping on email on a mail server (which would be inconceivable in a practical sense, but I just wanted to calm any possible worries).
- I sent the questions individually to each member, as opposed to using either CC or BCC to send them. This ensured there would be no accidental Reply-All gaffe in which one member's feedback would go out to the other members.

Each of these steps was subtle but important: they helped to secure confidence in the respondents so they could provide me with the honesty of feedback that I required. It worked, and I got some excellent feedback in that assessment.

The last bullet on the previous list was all about reducing the possibility of a gaffe. These accidents and mistakes have happened to us all: accidental emails, phone calls, and messages sent to the wrong people with sometimes embarrassing consequences. These gaffes are bad enough in nonsensitive situations, but when we are dealing with private data, they can be very serious. As such, you need to ensure you are aware of possible gaffes and try at all costs to avoid them.

As a starting point, here is a gallery of what not to do:

Email

In an email discussion about private topics, always check who is receiving the email. This is particularly risky these days with auto-completion. Believe me, I speak from experience....

Mailing lists

Always double-check that when having a private conversation, someone hasn't included a mailing list address. This happens a lot more often than you would imagine.

Blogging

When you hear that exciting piece of news that someone tells you, ask first if you can blog it. Many have fallen foul of blogging private information. That never ends well.

Phone calls

When discussing private topics, check who is around you. There may well be members of your community you don't know who are listening to every word.

Online chat networks

Before you talk to "jon_c" online about some private topics, just double-check it is not "jon_o" or "ron_c". That could get you in quite a pickle.

With a careful consideration of the expectations and risks surrounding privacy, it is likely that you can gather your feedback with no ill consequences. The key thing here is to *think before you do*. Checking that list of email addresses once more may take 10 seconds, but could prevent years of potential embarrassment.

Moving On

In this chapter we have explored many of the methods in which we can open up our community to take a peek inside. We have discussed the opportunities and pitfalls associated with measuring community, and this chapter should have provided you with a firm foundation in which to gather data that can help you optimize your community and make it more efficient, pleasurable, and productive.

Now we are going to move on to discuss one of the most important elements of community, particularly in large and growing communities: governance.

Governance

“Which is the best government? That which teaches us to govern ourselves.”

—Johann Wolfgang von Goethe

MIKE BASINGER IS A NICE GUY. Some would say a little too nice for his own good: he is one of those people who are impossible to dislike, no matter how much you try. Quiet, conscientious, considerate, and understated, Mike is the epitome of open source community. Few would imagine that he helps to govern the worldwide Ubuntu community at the highest level. At the same time, many of the people who know that wouldn't realize that Mike has never worked for Canonical Ltd., Ubuntu's commercial sponsor; he has always remained a volunteer.

Back in 2005, Mike joined the Ubuntu community by throwing himself into the bustling Ubuntu Forums soon after switching to Ubuntu. The forums were a good fit for him. Mike has a passion for helping people with technical problems. He loves the thrill of the chase: hunting down those pesky issues and helping to fix them for appreciative community members. As a good community citizen, Mike was on the front line helping new Ubuntu users get acclimated to their new waters.

The forums became stunningly popular. Thousands and thousands of excitable community members joined, and each day pages of conversation on all manner of topics would flow throughout the site. Interestingly, unlike most other Linux distribution forums, the Ubuntu Forums were not created by the commercial sponsor (in this case Canonical Ltd.). Instead, the

Ubuntu Forums were created by Ryan Troy, a passionate user who wanted to build an environment in which any Ubuntu user could ask for help and talk to other users.

As the forums grew, their popularity generated a few bumps in the road, and they needed more and more organizational focus. New moderators needed to be appointed, frustrating conflict issues were getting more common due to the size and diversity of the forums membership, administrators needed to distribute maintenance matters, and there was increasing demand for the forums to integrate with other parts of the Ubuntu community. The forums needed governing, and who should be one of the people to step up to the plate...? One Mike Basinger.

Over the following months, the Forums Council was designed, documented, and ultimately approved by the main Ubuntu Community Council. Alongside his peers on the Forums Council, Mike took the lead to govern. He and they did an excellent job, and Mike continued to contribute throughout the community.

In March 2007, the highest governing body in the Ubuntu community, the Community Council, was preparing to elect a new board of governing members. Clearly impressed with Mike's contributions to the Forums Council and elsewhere, Mark Shuttleworth (the leader of the Ubuntu project) asked Mike if he would consider a nomination to the Community Council. Somewhat stunned yet flattered, Mike considered Mark's suggestion and then agreed to be put forth as a nomination.

Just before the Ubuntu Developer Summit in Seville in southern Spain, Mark announced the nominees for the Community Council. The four nominees were put before a Yes/No vote by all approved Ubuntu Members, and Mike was elected to the Community Council in May 2007. It was a good day for Ubuntu, and it was a good day for Mike.

From his beginnings as a volunteer contributor like any other, Mike had risen to one of the most powerful roles in the Ubuntu community. This is entirely due to his hard and careful work and his commitment and devotion to the spirit and community of Ubuntu. When I asked him what kept him excited about Ubuntu throughout this entire period, he shared his clear passion for Ubuntu and the opportunities it has opened up for him:

What excites me about the community governance is the sense that Ubuntu is a community of thousands of people from every country, race, sex, and religion who have got together and said "we want computing to be this way." Linux and Open Source have enabled this as opposed to what Microsoft or Apple tell you. It is the sense that our community's governance is open and anyone who wants to contribute can and has a say in the direction of Ubuntu. It is that the community's main focus is to help each other, be that is to write code, create documentation, or answer questions from our users.

I am intensely proud of Mike's experience. Stories like this illustrate the power and opportunity of community to build methods of collaboration and governance that are truly equal rewarding and further enabling the hardworking. Stories like this make community leaders feel all warm and fuzzy. And stories like this, along with the warm and fuzzy feeling they engender, make

us community leaders unusually generous in a bar. If a community manager offers to buy you a pint some day, it's likely that something such as this has just happened....

Accountability

A significant contributing factor in Mike's success is that he feels a genuine and very real sense of accountability and responsibility for his work and Ubuntu as a whole.

In volunteer environments, participation is entirely optional. Some of your volunteers will spend every day entirely plugged into your community, and some will contribute whenever they please. Unfortunately, the latter form of drive-by contributors can't support many of the responsibilities and leadership components in a community. If you want to release a software product and commit publicly to a release date, you want to ensure that whatever happens, your volunteers feel a very personal sense of commitment to achieving that goal. Likewise, if your community appoints members to lead and govern a community, they must feel and exhibit a strong sense of responsibility and commitment. In short, you want people to feel accountable for their actions: you want them to *make time* when the community is under the pump to achieve something.

Accountability is a valuable asset. It is as close to a declaration of commitment that you will get in a community. Some of your members will feel this sense of accountability and some won't. Those who do are a valuable resource; you should entrust them with responsibility, but don't take advantage of their sense of accountability.

As community leaders, we should be laden with a sense of accountability. We are trusted to guide and inspire the community forward, and we should feel that responsibility in a very real way. With that sense of accountability is a responsibility to listen and learn. Our communities not only seek leadership from us, but also substance when it comes to governance. A great community not only makes it simple and enjoyable for a member to contribute to its tasks, but to contribute to this governance. This is the goal of this chapter.

Governance Does Not Suck

Of all the topics that fall under the umbrella of community leadership, governance is one of the most important yet most misunderstood. In the traditional sense, we are all intimately familiar with governance: it is exemplified by the government of your country. Your government is the representative body that manages the nation's resources and deals with its problems, opportunities, and current affairs. For most of us, our experience of government is the legislation, processes, and laws that define our daily lives, and those who perform this governance are the suited and booted politicians that populate our newspapers, television news shows, and radio broadcasts. We are all familiar with our governments, and we all have an opinion about them.

Unfortunately, opinion often points south, and governments get something of a bad reputation. It is easy to see why: the media is littered with stories of incompetence, sleaze, and self-interest, largely defended by toadying policymakers who refuse to provide a straight answer to a straight question. When queried by journalists, TV anchors, and researchers, it is not uncommon for the subjects of this governance (the citizens) to view the whole shebang with an air of suspicion: a government seen as fundamentally disconnected from the people. In a manner of speaking, it can be easy to draw the conclusion that all governments essentially suck.

Unfortunately, many citizens are so frustrated and disappointed by the incompetence in their government that they see the very concept of government as fatally flawed, as a system that fails to engage with its people with a raft of promises, few of which are faithfully kept.

Of course, this view is nonsense. The concept of selecting a few to effectively govern the many works, and we should not allow the incompetence of specific individuals to tarnish the potential of the system. In many cases we never get to hear about the amazing work of government, just the scandals and failed initiatives.

For years many governments have successfully delivered radical change and improvements to their people. Consider the rebuilding of Europe after World Wars I and II, expanding the right to vote, equal access to public accommodations, reducing disease, reducing workplace discrimination, legislation around safe food and drinking water, the nationwide construction of highways, financial security in retirement, scientific funding and technical research, reducing hunger and improving nutrition, space exploration, and more. It is government that helped forward these worthy achievements. When the system works, beautiful things can happen.

Governance and Community

In the same way that the government of a country is tasked with improving infrastructure, living conditions, and the welfare of its nation, the governance body of a community is similarly tasked with the welfare of those it governs. Instead of ensuring clean drinking water, community governance ensures transparency in our deadlines. Instead of readily available health care, we strive for robust resources to do our work. Instead of building that freeway, we build effective and open communication processes. The issues may differ, but the primary function of representing and maintaining the interests of those you govern remains.

Of course, governance is an epic and often academic topic. Thousands of books, research papers, and understudies have sought to understand, rationalize, and quantify what makes great governance. As a result, much of the available teaching is outrageously complex and written for those who have devoted their lives to understanding the subject.

I have *not* committed my life to governance, I have never served a professorship in governance, and I don't plan on doing so. I am a community manager, not a governor, and I would rather spend my time helping my community be effective than reading a library full of theory on governance.

Like many of you, what I seek to understand is how I can ensure my community will be governed well. I want to cut to the chase: what are the important elements to focus on in building a smooth and efficient governance structure?

In this chapter, we are going to uncover these elements. We are going to explore the aims and intentions of great governance, look at different approaches, and then discuss how to produce your own governance bodies. To seal the deal I am going to provide an extensive case study of how the Ubuntu governance structure works: this can be an excellent basis in which to roll out your own approach.

Let's roll....

The Case for Governance

Although I may appear to be a lighthearted guy on the outside, inside burns a cynical Englishman. Like many others, I myself have a blacklist of irritants in life. Hovering near the bottom of the list are bad customer salespeople, above that are those objects that attract the full velocity of my little toe when I walk in my house without shoes, then sycophantic reality TV imbeciles, and, bobbing around the top of the list, people who jump too quickly to governance.

Governance is not a rite of passage for community. It is not an expected norm, and its absence is not something that is perceived as immaturity. Before we begin exploring the nuances of governance, we need to determine if you need governance in the first place.

At the end of the day, only your community should judge whether governance is required and what form it will take. There are, however, some indicators that suggest that a governing body could be useful to have in place:

Size of the community

One of the first indicators that governance may be required is that your community grows extensively. If you have a community of around 10 people, governance is probably not required. If you have a community of 100 contributors (not users/consumers/onlookers) or more, it becomes a more pressing consideration. If you exceed the 1,000 mark, you should strongly consider governance, and possibly muscle relaxants, too.

Increasing conflict

Conflict resolution is a primary responsibility in governance. You should be careful about what you consider real conflict, though. People disagreeing on a few things is not conflict. People having full arguments in which multiple people are involved and factions develop is a far more serious issue. When this happens, sometimes the different sides hit an impasse

and can't move forward. Governance can really help here, under the proviso that the community respects the conclusions of the governance body.

Extensive resources

If you find that your community requires significant resources and some of these resources are donated, you may require a governing body to oversee the stewardship of these resources. An example of this could include a software project such as the Debian (<http://www.debian.org>) project, which requires extensive resources such as servers, hosting, build farms, and more.

Commercial interests

When there are commercial interests in a community, a governance body can be useful to ensure that the community is "kept honest." The governance body should be tasked with the responsibility of always maintaining and defending the primary values of the community and standing up against any improper requests that may result from commercial sponsors.

If some of these elements apply to your community, it would be worth considering governance in more detail. Let's now expand on this introduction and explore some of the responsibilities that governing bodies can provide, and consider how this could apply to your own community.

To ensure we focus our minds on these important topics, I am going to revisit the approach from earlier in the book when we built a Community TODO List to remember which items were important in growing strong community. Our list will look a little like this.

Governance TODO List

- *Item.*
 - *Item.*
 -
-

Follow the Leader

The primary responsibility of a governance body is to lead. It is there to initiate and engage in a conversation about topics that affect the community as a whole and represent the best interests of that community. A governing body seeks to understand and make decisions that are representative of the community, its goals, and its culture.

For many communities, leadership is broken into a few different threads that pull in the same direction to form a diversely detailed governing body. Just as a conventional government will have leaders and departments that focus on specific areas (e.g., Department of Health, Department of Employment), many communities divide their leadership up, too.

For most communities, this leadership is broken up as:

General governance

Decisions that need to be made around general topics that apply to the community as a whole. This could include things such as how people join the community, resources and infrastructure, community-wide policies and procedures, governance changes, etc.

Direction

Decisions about the goals, ambitions, and focus of the community. As an example, with a software project, this kind of leadership would decide on which features to aim for in the next release. In a local civil rights group, this could be how the group is planning on raising awareness and which campaigns will be organized.

Specialist governance

This really applies only to larger communities. Specialized governance may be required in specific areas of expertise. As an example, in a software community the developer community may require its own governance, and so may the discussion forums.

For many communities, each type of leadership falls together inside a single governance body. This is particularly common in smaller communities.

For example, many user groups have a known collection of leaders who advise and govern around all manner of topics, including how people can join the group, how the group should focus their efforts, which campaigns should be worked on, how money and assets are handled, and more. It is often the same small set of people who advise on these issues, and the expertise and more general focus of a user group makes this kind of simple approach a perfect solution.

For larger or more specialized communities, these separate leadership roles are often divided into different governance bodies. As an example, in the Ubuntu community we have the following governance bodies:

- Community Council→General Governance.
- Technical Board→Direction (technical direction and processes).
- Team Councils→Specialist Governance (e.g., the Forums Council governs the Ubuntu Forums).

If this approach piques your interest, you should be tickled pink to learn that I will be providing a detailed summary of how Ubuntu is governed later in this chapter. We will look at the structure of its governance, crack it open, and see how it works. Let's now add our first item to our Governance TODO List.

Governance TODO List

- Ensure your community is governed in terms of General Governance, Direction, and (if applicable) Specialist Governance.
-

Engage the People

Governments are fundamentally representatives of the people, and for this representation to be fair and accurate, the government needs to engage with the people. A government that lowers itself into a silo and rarely interacts with its people is doomed to a future riddled with problems. If a government fails to communicate with its people, the people will not only lose faith in those who govern, but also in the confidence of being governed in the first place.

George Burns, the famous American comedian, replete with arched eyebrow and cigar smoke punctuation, had a lot to say about government. In a 1979 issue of *Life* magazine, Burns shared a nugget of insight that resonated with many people:

Too bad that all the people who know how to run the country are busy driving taxicabs and cutting hair.

He isn't wrong. When I started my career in community management, I spent a lot of time traveling. At OpenAdvantage I would hurl myself across Europe from conference to conference, and my move to Canonical expanded my travel to the worldwide stage. As I moved from plane to plane, and ate miniature bag after miniature bag of salted peanuts, I went to meet and greet the various members of my community. While en route, I had my own opportunity to meet a battalion of taxi drivers, each with his own carefully considered manifesto of what his government had screwed up and the rather obvious solution to the problem. I heard views on Chinese politics in San Francisco, opinions of U.S. trade treaties in Prague, and just about everyone had a view on George W. Bush. I suspect that that man doesn't get as many Christmas cards as he used to....

Virtually all of these taxi drivers had one thing in common: they all felt that their voice was rarely heard by their governments. Their right to vote was always cherished, but it was seen as a binary decision around favor or rejection. These always-entertaining cabbies weren't asking for much, they just wanted to be able to have a conversation with their governments, and so they should.

Great communities always have a close connection between the governing bodies and the members of the community. This relationship requires more than a communication channel with the governance body; that part is simple. Real engagement is when government and community enter into a two-way conversation. Gone should be the days in which the

government dictates to the people. Today governance should focus its heart on engaging in conversation with its members. Whether you call it shooting the breeze, having a good ol' chin-wag, or anything else, you need to be having it with the people who govern you.

It is likely you are going to be governing others, and as such we need to ensure that we are cognizant of engaging in conversation with our communities. Let's put this on our list.

Governance TODO List

- Ensure your community is governed in terms of General Governance, Direction, and (if applicable) Specialist Governance.
 - Build communication channels between the governance body and those whom they govern.
 - Foster a culture in which the members of the community can engage in conversation, debate, and discussion with their governing bodies.
-

Aspire to Inspire

Every community looks to its governing members for direction and advice, and leadership helps to ensure the community is on the right path and feeling productive and nimble. A very close cousin of leadership is inspiration. Your members will also look to you to inspire, motivate, and enthuse them. If you make the hairs on the backs of their necks stand on end when you lay out your vision and what you want accomplished, your community will succeed.

Inspiration is an important responsibility for leaders. Earlier in this book we spent some time discussing how to write inspiring words for your community. Unfortunately, some community leaders who work to build governance bodies seem to forget that governance is seen as leadership and leaders are expected to inspire. Don't fall into the trap of assuming that governance is merely about decision-making. There is no reason why you can't constrict it in this way, but you will be missing out on a wealth of opportunities to excite and energize your community.

You can see the divergence in this approach in conventional government. Compare and contrast how some presidential figures have approached inspiration. A recent example of an inspirational orator is Barack Obama. Irrespective of where you stand on his politics, Obama has inspired a significant chunk of the electorate behind a rhetoric of pumped-up, energized, forward-looking narrative, and the promise of a bright future. I am sure you can think of many presidents who merely took office and started legislating.

Obama primarily inspired people with the promise of a brighter future, and you should do the same for your own community. However, there is another important and more focused responsibility that governance bodies should shoot for: inspiring your members based on the values of your communities.

Governance is almost entirely based around values. When a government appears open, transparent, and honest, it generates trust, respect, and a faith in its leaders. When a government throws values out of the window and replaces them with self-interest and sleaze, your community may as well pack up its bags and go home.

You need to not only understand your values, but celebrate them. It is these values that will continue to make your community feel open and engaging. When you learn to inspire based upon values and the promise of the future, it will stand you in good stead and stand your community in even better stead.

Let's ensure we make a note of this important topic on our TODO list.

Governance TODO List

- Ensure your community is governed in terms of General Governance, Direction, and (if applicable) Specialist Governance.
 - Build communication channels between the governance body and those whom they govern.
 - Foster a culture in which the members of the community can engage in conversation, debate, and discussion with their governing bodies.
 - Seek to inspire, motivate, and enthuse your community based on future opportunities and the honesty and openness of your governance.
-

To Bring Peace

A final topic that is an essential function of governance is the ability to bring peace to your community. We all look to our leaders to resolve and calm conflict, and your community will be no different.

Every community faces conflict. Communities attract different personalities, goals, approaches, attitudes, ambitions, and opinions, and some of them are going to rub up the wrong way. In the worst of these situations, conflict can cause deadlocks, and the community will look to its governors to unblock it. We need to expect conflict, acknowledge it, and react to it elegantly.

Conflict resolution is a large and complex topic, and with this in mind I have devoted an entire chapter to it later in the book ([Chapter 9](#)). As such, we will revisit this topic in that chapter. For now, though, you should simply ensure that inside the box in your mind that says “Governance” is a smaller box with “Conflict Resolution” written on the front. Let’s also add it to our TODO list.

Governance TODO List

- Ensure your community is governed in terms of General Governance, Direction, and (if applicable) Specialist Governance.
 - Build communication channels between the governance body and those whom they govern.
 - Foster a culture in which the members of the community can engage in conversation, debate, and discussion with their governing bodies.
 - Seek to inspire, motivate, and enthuse your community based on future opportunities and the honesty and openness of your governance.
 - Provide a clear, objective, and mature approach to solving conflict and contentious issues and for providing a decision when faced with deadlocks.
-

Learning from the Leaders

With a firm understanding of the needs and aspiration behind governance, let’s now take a few minutes to look at some of the existing approaches to governance that some communities have taken.

Of course, there are thousands of communities around the world, with varying governance approaches. Fortunately there are patterns in the chaos, and there are three prevalent themes in many communities. These are:

Dictatorial charismatic leadership

Governance and decision-making that is largely driven and controlled by a single person.

Enlightened dictatorship

Governance that effectively has no formal leader but in which leadership is determined by reputation in the community.

Delegated governance

Governance that is delegated to a series of smaller units that all fit together to form a single governing body.

Let's take a quick spin through these different themes and see how they apply to our own communities. This can then provide us with an idea of how we want to structure our own governance.

Dictatorial Charismatic Leadership

People hate the word *dictator*. It typically gets something of a bad reputation in polite conversation. Unfortunately, history has taught us that many of the most famous “dictators” were mass-murdering psychopaths who foisted their attitudes onto their people. I am sure that it was for this reason that some of you may have been little surprised to read that there is a type of acceptable governance that is driven by dictators. Fortunately, mass murder is not a common practice in these communities.... Dictator-led communities work just like they say on the tin: they are communities in which a single person calls the shots. This person will often set direction and focus, approve what is acceptable in the community, and in technical communities will be the arbiter of what gets included in the project. If you still can't stomach the word *dictator*, substitute *charismatic leader* in your mind each time I use it.

Now, I know what you are thinking: this doesn't sound at all very community spirited. A community in which one person acts as the funnel through which everyone else must flow? Surely that can't actually work! You would be surprised.

There are many dictator-led communities that are popular and generate very large communities. Two very prominent technical examples of this include Linux and Python. Within these communities exist two very visible leaders: Linus Torvalds and Guido van Rossum, respectively. Linus and Guido are the people who have traditionally decided on direction, set focus, and accepted or rejected contributions.

In the free software world, one of the most notable cases of dictatorship was the choice of the third version of the GNU General Public License, perhaps the software license in most widespread use by free software projects (including Linux). Years of discussion went into this license, including intense meetings and negotiations with representatives of companies and software projects of all sizes. Yet in the end, someone had to make a decision, and that person was the illustrious president of the Free Software Foundation, Richard Stallman.

Although the dictators in these communities are typically the original founders of the community, this does not mean that they don't lean on the community for help and assistance in judging contributions to the project. Typically these leaders will handpick trusted and reliable members to lend a hand. In these communities there is often no open governance, no elections, and no community-discussed focus and direction.

Despite these communities' restrictive nature, time and time again contributors join up and enjoy their involvement. Despite this, however, I would personally recommend against a dictator-led community for a few reasons:

Lack of transparency

Earlier I went into detail about why openness and transparency are important in volunteer communities. Dictatorial communities are something of an antithesis to this approach, and their leaders always face the risk of not being representative of the views of the wider community.

Bus factor

Communities that have a single, strongly focused leader face considerable risk if that leader gets hit by a bus. Other, more openly governed communities are often able to transition to other leaders more efficiently.

Direction

Communities with a single leader who decides what direction the community takes can have difficulty expanding their focus. As an example, if your community is building a website, the leader may stick with outdated ideas of how it should be structured and behave after the rest of the world has moved to new technologies and architectures. There is a very real risk of "it's my ball and we are playing my game" with this approach.

Of course, if you are the founder of your community, only you can decide whether the dictatorial approach is suitable. In almost all scenarios I would recommend against it, and if you are keen to have a strong level of control, at least delegate some control to councils (as we will discuss in the section "[Delegated Governance](#)" on page 225).

Technical communities who have successfully implemented of the dictatorial model often refers to their primary leader as a *benevolent dictator*. This term is inspired by historical leaders such as Mihailo Obrenović III, Prince of Serbia; Maria Theresa of Austria; and Frederick II of Prussia, who led and governed their people as dictators but applied rationality to their approaches.

In historical terms, this rationality was manifested as religious toleration, freedom of speech and the press, and the right to hold private property. In the technical space, benevolent dictators apply the same sense of rationality to toleration, and freedom of speech and press, but also often inspire open contributions, delegated responsibilities, and more.

Enlightened Dictatorship

In contrast to dictatorship and benevolent dictators, a popular form of community that has grown in both online and offline communities is *enlightened dictatorship*. With this approach the concept is simple: there is essentially no leader.

Confused? Don't be.

Not all collaborative communities need leaders; they just need a general ad hoc agreement of what is not acceptable. When your community knows what is “not cool,” it means that all other contributions are, by definition, “cool.”

While this may sound like an environment driven by chaos and mismatched focus, many communities have been productive with this approach. Although there is no formal leader, the sense of leadership naturally grows out of reputations that are developed and matured within the community.

At its heart, this is a pure form of meritocracy: when people do great work, they become thought leaders. Although some communities may enable people to climb the ladder based upon meritocracy, in an enlightened dictatorship there simply is no ladder.

An interesting example of this approach is the KDE project. Founded by Matthias Ettrich, KDE set out to build an easy-to-use desktop environment, and it has become popular among Linux and other Free Software enthusiasts.

One such enthusiast was myself. Back in 2001 I joined the project, attracted not only by its purpose and direction, but also by the apparent openness in the community. To participate back then involved learning a fairly stiff set of programming tools that was commanded by the Aladdin's Cave of complexity that is the C++ programming language.

My C++ skills were...frankly...pants. I tried my best to learn. I bought books, I read websites, took courses, and even watched tuition videos hosted by a man with a kipper tie. Despite my bumbling attempts at C++, one magical opportunity existed: I knew that if I could gain those skills, I was welcome in that community. Of course, these days a more diverse menu of opportunities is available for those who want to participate in KDE, but even in the dark age of when I was involved, openness was always present.

Although KDE has hundreds of contributors from around the world, there is no formalized leader that exists beyond the limited set of developers who look after the different chunks of code (called modules) and the release manager. It is the many developers involved in KDE who are the arbiters of which bugs get fixed and what features are added. This makes for an interesting dynamic when interacting within the group and how the group is perceived by the outside community. This dynamic has created something of a mantra of “those who code get their way,” a position that some cherish as a pure approach to open source community and some believe makes a community less approachable. Whichever the position you take, the KDE project has made great strides in productivity.

One of the most evident places this mantra is exercised is within the KHTML portion of the project, which produces technology for displaying web pages. As WebKit (an alternative technology originally based on KHTML) has gained traction within the embedded and desktop

markets, many people from outside the community have questioned why KHTML is still maintained at all and not been abandoned in favor of WebKit.

Ian Reinhart Geiser, a longtime KDE contributor and member of the KHTML project, explains:

Technical arguments aside, it is the choice of the maintainers of KHTML to keep maintaining their code base and continuing its life. There is no active movement against WebKit and in fact there is a smaller group of developers who are working on a KDE version of WebKit. The will of the KHTML developers is what decides the technical direction of progress in the KDE project.

It is the presence of enlightened dictatorship that Geiser arguably believes has enabled the KHTML developers to push forward with their own approaches that they feel happy with. Although it may not make sense to some, it exemplifies the freedom in this project for raw technical contributions to define direction—a freedom that has helped the project maintain its flexibility and its momentum.

Delegated Governance

Our final approach to governance is the one I find most appropriate, open, transparent, and conducive to robust community. In essence, it puts governance in the hands of an openly nominated and elected group of people who have respected and recognized expertise.

There are many powerful examples of this approach to governance in place, and it is one that we have used extensively in the Ubuntu project. We are not alone, though; each of the major Linux distributions takes the same approach, including Debian, Fedora, and OpenSuSE.

In delegated governance, the founders of the project nominate a diverse body of leaders to represent the best interests of the community. This governing body has a mandate and a set of responsibilities, along with a transparent procedure for electing or otherwise replacing members, and these members are typically chosen for their well-respected contributions to the community. Although the approach is very open and transparent, it does have one distinctive risk: *complexity*.

Building an open yet responsible governance body is not a particularly simple task. You need to know what you want to achieve, how to structure the governance, what the requirements of your members are, and how to ensure that your community is fully supportive of the authority put in the hands of the resulting Community Council.

That can appear complicated to put in place, and can also seem overly complex to your community. As such, we need to work hard not only to craft an effective governance body, but to communicate to the whole community the way governance works and get them on board. Again, we can do this. We just need to pay keen attention to detail and ensure we tick all the right boxes.

THE UGLY B-WORD: STILL UGLY

Here I need to throw out another quick reminder of the importance of avoiding *bureaucracy*.

Always remember that most people view governance as neither cool nor interesting. Our goal is to keep the amount of governance infrastructure and documentation to an absolute minimum while still maintaining the quality and objectivity that we strive for.

Setting Up a Community Council

If you follow my advice and choose delegated governance—or at least some elements of it—you need to build a governance body. For many communities this means forming a *council*. A council is a group of selected members in a community who govern collectively. Issues and topics can be presented to this council, and they will come to a united opinion that is typically concluded in a vote. As an example, if the council has seven members, they may allocate 20 minutes to discussing a proposal and let it go forward if four members support it. (In practice, if your council is deciding a lot of issues on the basis of a bare majority, the community is not well governed and there is likely to be destructive conflict—but we’re just covering the basic principle here.)

Many communities form a *Community Council*, a single council that governs the entire community. For most communities this is a great solution, but larger communities benefit from delegating some authority from the Community Council to additional subcouncils. For now, though, we are going to explore what is involved in setting up a single Community Council to govern your community.

COUNCILS VERSUS TEAMS

This chapter discusses long-term forms of governance, which are basically invested in *councils* or *boards*. This is different from groups that form more organically and naturally to handle particular objectives or goals in your strategic plan. The less formal groups are usually called *teams* or *committees*. I talked about teams in [Chapter 2](#), and they are certainly the basis for effective community work. But teams or committees focus on getting some concrete task done, not on governance, so I don’t cover them in this chapter.

Certainly, a council can form a team to handle a specific task related to governance, such as writing one of the documents I describe in this chapter or nominating members for councils.

Designing a Council

Our first task is to understand what you want to achieve with your Community Council and document the full extent of its responsibilities. Governance bodies are fundamentally institutions with an authority the community agrees on, so we need to understand where that authority begins and ends. This agreement is also important for your community: they will want to have confidence in what the Community Council can help with, and also will want to understand what areas are beyond their control. Obviously, the limits are even more important for subcouncils so they don't make decisions at cross-purposes.

Designing and documenting your council is important for three reasons:

- Explicitly discussing the council's responsibilities will help you as a community leader understand the full rationale behind the council and how it should work.
- Clear expectations also help everyone who serves on the council know how to handle their responsibilities and will make them more willing to serve.
- A well-designed and -documented council will protect your community from accusations of corruption. If the remit and extent of the council are well known and the members are known to act within those expectations, accusations of favoritism or misuse of authority will be rare.

The building blocks of a council are your decisions regarding:

- Responsibilities.
- Structure.
- Membership.
- Communication.

The next few sections of this chapter look at the choices you have for each of these building blocks, and some of the criteria that help you make your choice. Along the way, I'll show you how to document your decision—a crucial aspect of governance. Like anything else you want your members to learn about, a council requires documentation.

To make this as simple to understand as possible, it's easiest to describe an example fictional council. So, friends, for the next few pages we are now all members of an exciting new open source project that has produced a Frequently Asked Questions content management system called Tobe (named after "to be or not to be, that is the question"). Actually, trivia fans, this was an old project I worked on many moons ago when I was a web developer, so it seems like an apt choice for an example.

NOTE

The Tobe project is no longer running, so don't get too excited if you are looking for something similar. I still think an FAQ management system would be an incredible resource for many communities, though, so if you produce one, do let me know!

Our fictional Tobe system is a complete web application for maintaining FAQs. It is written using the PHP language and MySQL database. Because it's fictional, let's say it has an excellent user interface, wonderfully written code, and legions of fans around the world (including Johnny Depp and Nicole Kidman). As such, Tobe has a large and bustling contributor community, so large that we are feeling the need for a Community Council to help guide the project forward and ensure the community is always open and accessible.

Responsibilities

The primary purpose of a council is to provide fair governance and feedback on an agreed set of responsibilities in the community. What you choose as these responsibilities will vary: it is highly dependent on your community. But your community needs to be united in agreement about what these responsibilities are. We need the community to be fully aware of what purpose the council serves.

Choosing this set of responsibilities may be more complicated than you imagine. What we are shooting for here is a council that members need to consult relatively rarely and only when there's no way around it. What we don't want to do is to build an environment in which someone can't scratch her leg without consulting the council. If we assign too much responsibility to the council, it will annoy the community and make it feel restrictive, and I can guarantee that the council will also become an impediment to getting things done. For communities, I heartily endorse the old aphorism, "That government is best which governs least."

On the other hand, we don't want to provide the council with too little responsibility: it does need to have the power to resolve disputes and set direction firmly when the community needs that guidance. In a nutshell, we need to strike a balance.

The common responsibilities that you may want to consider for your council are:

Membership approval/rejection

A council is often a useful body for approving or rejecting members who want to join your project. You need to decide what kind of members these should be—general members, developers, other contributors, etc. We will discuss this in more detail shortly.

Conflict resolution

Many councils act as an objective third party in dealing with conflict. In this case, members of the community will schedule a conflict discussion with the council, and the council members decide on an appropriate outcome. This has occurred in the Ubuntu community many times, and the Community Council typically has been successful in bringing closure to conflict issues.

Project values

Many councils act as an arbiter over the core values of a community. As an example, in the Ubuntu community there is a set of core values around freedom that the project always seeks to maintain. If anyone feels these core values should be adjusted, or if they have not

been met, the member is advised to raise this with the Community Council. Note that upholding core values is key to conflict resolution, but it's not quite the same thing, because communities also sometimes drift away from their values unintentionally and just need a reminder.

Community process changes

Councils can drive changes to processes in your community and get those changes accepted. If you have a Community Council that is known to fairly represent the community and they approve a process change, your community will feel like the process is now considered "official." As an example, in the Ubuntu community we changed the process in which developers applied to be a member of the project, and the Community Council needed to approve it. When it was approved, the community unequivocally accepted it.

Ordaining governance bodies

If your community is large and requires multiple governance councils, your primary Community Council should be the body to make a proposed subcouncil official. As an example, Ubuntu's Community Council approved and officially nominated members for subcouncils such as the Forums Council and IRC Council.

Direction

Councils often determine, or at least weigh in on, the long-term direction of a project. As an example, if you are a software project, your council may decide on the feature goals for a given release targeting a given release date. This is a controversial topic in governance, and many communities vary in whether they allow a governing body to dictate features or their timing. In the Ubuntu community, a separate body called the Technical Board acts as an arbiter over debates concerning feature direction.

While you are considering which responsibilities your council should take on, ask yourself how much impact these topics are going to have on the day-to-day work in your community. As we said earlier, you want to achieve a state in which your community has to interact with a governing body only on limited occasions.

Let's look at a few examples of what not to do:

- As I mentioned, leaders may decide to grant councils the right to set feature direction when there's no need for it. If you have excellent contributors who are not on the council, the right features may simply arise from their work and discussions.
- The council does not act as a bottleneck to choose people for small, straightforward roles such as mailing list moderator. Making the moderators go before the council to approve a new moderator is a waste of everybody's time, as long as the moderators are happy with their own decision.
- The council should not have to approve the formation of a new team. Let people set up their own teams as they see a need. This will result in more teams and more diversity in your project. But the council could set up a special category of *approved teams* who have

additional responsibility and need to be vetted by the council. An example of a useful role for an approved team is the Approved LoCo Teams (local user groups) in the Ubuntu community. While anyone can set up a local Ubuntu LoCo team, those teams who have demonstrated consistent good work can be approved and will have better access to merchandise, content, and resources. This is a useful way of reducing waste of resources because approved LoCo teams are generally a lot more responsible.

After deciding on these responsibilities, you should put together a document that outlines each responsibility in detail. This can be the basis for forming the council.

Let's apply this to our Community Council for the Tobe project example. I think this would be a suitable selection of privileges.

TOBE COMMUNITY COUNCIL RESPONSIBILITIES

The Tobe Community Council (TCC) will govern the Tobe project by exercising the following responsibilities:

- **Developer Approval**—the TCC will judge applications for project developer positions. A prospective developer should present a wiki page that outlines his work in the Tobe project and lists at least two endorsements from current developers. The TCC will vote on the application.
- **Process Changes**—any significant changes to community-wide processes should be approved by the TCC.
- **Governance Changes**—any changes to community governance (including the formation of new governance bodies) should be approved by the TCC.
- **Conflict Resolution**—if a member of the community has a conflict with other members that is hampering work and cannot be resolved, this issue can be raised to the TCC. The notice can either be given publicly in a meeting or be privately posted to the TCC mailing list. Any case put before the TCC should have as much externally referenced evidence as possible to demonstrate the conflict.

(You may notice that I have not included “Direction” in these responsibilities. I don't believe that governed feature direction is necessary for a small community such as Tobe, because it could inhibit innovation.)

Structure

With our responsibilities in place, we now need to plan our council's structure. While a council may seem a fairly straightforward structure—a group of cooperating members—we need to consider some important details.

The first decision is how many members should be on the council. For most of the councils I have been involved in, a good number is five to seven. This provides a good range of opinion on topics while building in enough redundancy to handle absences (such as when members are on vacation).

Base the number of council members on the number of good candidates you actually have available. Don't pick seven members for the sake of a large size if you only have two or three existing contributors who would make suitable council members. We will discuss how to assess quality members later in this chapter.

The next decision is whether one of your council members has a tie-breaking privilege. If you have a meeting attended by six council members and three vote on each side of an issue, you reach a deadlock that cannot move forward. So seriously consider appointing a member with the privilege to cast a tie-breaking vote in a deadlock situation. In the Ubuntu community, Mark Shuttleworth has this privilege. The Fedora and OpenSuSE Linux distributions have similar roles in place.

If you decide to include a tie-breaking role, ensure that the person in this role is the very model of objectivity and responsibility. The privilege confers great power, and you must ensure it is always exerted with the best interests of the community at heart. We will discuss membership requirements in the next section, and your tie-breaker should excel in demonstrating the attributes in that section.

NOTE

Of course, you may consider yourself for this privilege. But before you do, look at yourself in the mirror and ask whether you deserve it. Although by reading this book you evidently have the interests of community at heart, can you really offer an entirely objective opinion?

The next consideration when deciding on the membership structure of the council is how long each member serves. There are of course varying opinions on this. Some believe that one year is a suitable term, particularly given the fast-changing nature of community. Others believe two years is more appropriate because it provides a better sense of stability and focus for the council, and it lets your community get used to the same names and expectations. Some people, after serving on councils, say, "It took me a year to figure out how the council works."

I believe that term length is largely dependent on (a) the type of council you are building, and (b) the maturity of the membership you have available. For general Community Councils that oversee process changes, conflicts, and other similar elements from our list of responsibilities a few pages back, it is reasonable to have a two-year term. This has been used in many existing councils and has worked out well. For technical and direction-focused councils, I feel a one-year term is often more appropriate. A shorter term always ensures that your council has fresh perspective, and if the current council is serving well, simply allow the members to continue their term.

Term length is related to the decision of whether existing council members can (a) serve repeat terms, and (b) be on multiple councils simultaneously in your community.

I am personally against term limits. If you have excellent governing community members, you should allow them to continue doing great work. However, you should also have a fair and representative system in place to ensure that your community does not develop an “old boys club” with ineffective governing members who may not be especially forward-thinking. In other words, excellence should not be capped by term length, but you should ensure your community can kick out ineffective members.

ALWAYS MAINTAIN QUALITY

Remember that if you have problems with an “old boys club” nesting in your council, your governance structure *can* be changed to prevent it.

In some communities, if the selection process for governing members repeatedly nets ineffective officers, the general feeling can be “well, that’s the way the community works and we just have to deal with it.”

Don’t just deal with it. If your community governance is not working, change it so that it can work. In doing this, it’s important not to single out and excoriate problematic members. A proposed modification to council nomination or election should not be “David Foo is doing a terrible job, so he should be banned from applying to the council in the future.” Instead, you should build a system that weeds out ineffective members quickly and averts their nomination in the first place. You can base the system on experiences with current bad council members, but think in general terms.

Criteria for excluding and removing council members could include a required level of participation, audits of how well the council works, and an agreed manifesto listing what members will work on. Each of these initiatives can help your community judge whether a member is effective. Finally, institute a process to review whether members should still be on the council.

Of course, you need to temper these considerations with moderation: you do not want council members to feel like Big Brother is constantly checking in on them.

Commercial sponsorship

Before we move on, I want to raise a sensitive consideration when it comes to the structure of your council: commercial investment and sponsorship.

Many communities have commercial sponsors and investors. These sponsors often have a staff of developers and contributors who work within the community to make contributions that benefit the investor. Sometimes the community has a single sponsor, who perhaps founded the community in the first place.

Examples of this situation include the major Linux distributions: Ubuntu was founded and is mostly funded by Canonical Ltd., Fedora by Red Hat, and OpenSUSE by Novell. With such significant investment in these communities, what impact should these investors have in the governance of the project? At what point is investment in a community considered a reasonable justification for governance?

Let's take a look at the impact of these sponsors on the governance of those projects:

Ubuntu (sponsored by Canonical Ltd.)

Of the seven places in the Ubuntu Community Council, only one seat is appointed—that of Canonical founder Mark Shuttleworth, who has tie-breaking power. The other six seats are open to anyone, Canonical or otherwise.

Fedora (sponsored by Red Hat)

The Fedora Board (their Community Council) has a Red Hat-appointed chairperson with tie-breaking power and nine seats. Of those nine seats, four are appointed to Red Hat staff and the other five are openly elected. This means that there must be at least half the board working at Red Hat, one of whom has a tie-breaking vote.

OpenSUSE (sponsored by Novell)

The OpenSUSE Board (their Community Council) has a chairperson who works for Novell and who has tie-breaking privileges. There are four other seats, two of which must be occupied by Novell staff.

Although the numbers are different, the Fedora and OpenSUSE approaches produce essentially the same effect: a board in which 50% of the seats are reserved for the sponsor, one of whom is the chairman with a tie-breaking vote. This means that, conceivably, the sponsor could push through any issue they want: in OpenSUSE, the Novell staff could vote their majority, while in Fedora, there would be a deadlock of votes down the middle, allowing the chairman to cast the deciding vote.

The Ubuntu approach is different: Mark Shuttleworth has the only reserved seat. Even with his tie-breaking privilege, it would be difficult to push something through unless it got to a deadlock, and Canonical has no way to engineer a deadlock internally, as all other seats are equally available to non-Canonical contributors. In fact, at the time of writing the majority of members on the Community Council don't work for Canonical.

I believe that for general volunteer communities such as open source projects, sponsorship and investment should never buy you a place in a governance body. I say this not because these commercial sponsors cannot be trusted, but because volunteer associative communities are often based upon the contributions of all members, and these members give of themselves in exchange for the assurance that their hard work will go toward their fellow members and the community's future.

Let us now apply some of this thinking to our Tobe example council.

TOBE COMMUNITY COUNCIL STRUCTURE

The Tobe Community Council (TCC) will have five membership seats, one of which is held by the founder of the project and has a tie-breaker privilege.

Commercial sponsorship does not guarantee representation by employees of the sponsor on the council.

Membership

With a clear idea of your requirements and structure, the next step is to figure out what you are looking for in your council's members.

A council is nothing more than a collection of dependable people with a set of responsibilities. You need to depend on the members of your council to demonstrate maturity, competence, objectivity, and sensitivity. Good council members are there not just for their personal contributions and abilities, but because they can represent the needs of the community.

After I gave a keynote on governance at the SoCal Linux Expo in Los Angeles, a chap came over and asked me what appeared to be a devilishly simple question:

What kind of people should I look for to be on my community's council?

This got me thinking. Can we build a recipe for an excellent governing community member, and list the ingredients to look out for when choosing these members?

Well, kind of. In my experience, community members come in many shapes and sizes. There is absolutely no commonality in terms of gender, race, career choice, technical experience, or age. You simply can't look at someone in the street and determine that she is a great council member. As we discussed earlier in the book, these checkboxes of surface-level diversity items will not help you to build a great council. We instead need to look for deep-level attributes that point to maturity and capability in governance.

Fortunately, there is a common set of traits that you should be looking for. It is these ingredients that will indicate that someone has the chops for governing your community:

A listener

Great governors are always willing to listen. You will rely on your council members to listen to your community and make sensible decisions based upon hearing the full story. This is particularly important in cases of conflict resolution. A useful method of identifying this trait is to listen to a prospective council member in a normal conversation and make a mental note of just how much he listens as opposed to speaks. See how much he contributes to the discussion, see how often he chips in or interrupts, and see how much

he queries the information he hears. This will help you determine whether he is a good listener.

Unbiased and objective

Your council members need to ensure that they don't demonstrate any apparent sense of bias. Of course, everyone is biased in one way or another, but you need to shoot for council members that can do their best to put their biases to one side. A good test of this is if a prospective council member is open to changing her views based upon further information. Another good sign is whether she is able to agree on a topic with someone whom she typically disagrees with. Watch her debate, and observe if she adjusts her argument as the debate progresses: this is a great attribute to have in a governing member.

Detail-oriented

Great governors pay serious attention to detail. They understand that the devil is in the details, so they pick up on the hidden details in a discussion and ensure these details get covered. Watch how they communicate to see how they raise and react to details.

Reliable

One of the most significant considerations with community-based governance is simply having people show up and fulfill their responsibilities on the council. If you have a governing body that has members who should attend meetings, don't be surprised if some members don't show up. You need to find people who are willing to put that PlayStation controller down, willing to get out of bed early if a meeting is scheduled to cater to another part of the world, and willing to *make time* for your community. Of course, life happens and people cannot always make meetings, but you can assess how reliable your prospective members are by seeing how often they attend your community over an extended period of time. Don't assess them for the few months building up to nominations for the council, as they may try harder than normal to demonstrate reliability. Instead, observe them over a wider period of time without them knowing. This will help you get a more objective view on their reliability and attendance.

A fair fighter

This is a rare trait. You really want to look out for people who will fight for the right thing, but put the integrity of the council and the community above the outcome of a fight. The ideal personality here is relaxed and calm, but when required, will put their head above water to do the right thing. The only way you can assess this is to look at their experience since they joined the community and balance the times they have stood up for different things, whether it was justified, and how they handled losing an argument.

With these expectations in mind, you should properly document the expectations of council members so that prospective candidates have a clear idea of what is involved. Let's apply this to our Tobe example.

TOBE COMMUNITY COUNCIL MEMBERSHIP EXPECTATIONS

Each seat on the Tobe Community Council (TCC) is for a period of two (2) years and has the following expectations:

- **Engagement**—each member of the TCC is expected to engage with the community politely and fairly and to refrain from using bad language, offensive statements, or insulting comments.
- **Fairness**—each member of the TCC is expected to consider each case brought to it; listen to all the evidence and perspectives from the people involved; and pass a fair, unbiased, and objective decision based upon the evidence provided, the goals of the Tobe project, and the best interests of the community.
- **Reliability**—the TCC has a defined set of required meetings and responsibilities. Each member is expected to be responsive and receptive to these responsibilities. Each member is also expected to notify the TCC concerning any prolonged periods of absence. If a member of the TCC cannot fulfill his or her responsibilities, lacks the time to participate effectively, or otherwise cannot tend to the requirements of the TCC, he or she is expected to step down gracefully. It is also preferable but not required that the member who has stepped down help his or her replacement transition into the role easily.

This rather juicy statement combined with the *Responsibilities* and *Structure* statements earlier provide a comprehensive set of expectations around the role of a governing member and its scope.

Before we move on, I want to share a few thoughts about your own expectations. If this is your first time working to set up a governance body, it can often be difficult to figure out which people have the right requirements. If you are anything like I was when I first did this, you will be terrified that you have missed an important piece of fine print when documenting the council, and that you are going to unwittingly recruit a destructive idiot.

First, don't worry. This is your first time. Everyone makes mistakes, and you are going to as well. The best medicine for avoiding mistakes is experience, and I would highly recommend you find someone who has worked on governance before and ask that person to take a look at your work and pass comment. Take a quick look online at some of the existing governance bodies in communities that you follow, see who was involved in setting them up, and drop him an email to see whether he would be willing to offer some advice and thoughts.

NOTE

Of course, by saying this I know I am about to get deluged in email! I am certainly willing to help, but when the deluge occurs, I might not be able to help all of you, so don't be offended if I don't have time.

In this section we have focused our efforts primarily on the attributes that we are looking for in our council members, but not how we actually elect them. We will discuss this a little later.

Communication

With a design in place that has firm responsibilities, structure, and membership expectations, we now need to decide how our council members will communicate with each other. Here we need to decide (a) which resources they will use to communicate, and (b) the requirements we make on their time for activities such as meetings.

Let's start with resources. You first need to ensure you have a primary communications channel in place. Earlier in this book we discussed some of the resources that are available, so we won't cover that ground again, but let's instead look at a suitable approach that many communities have used: a single mailing list and online chat channel.

A mailing list is an excellent method of having general discussion inside the council. It is simple and low-bandwidth, and you can ensure that people's access to it is restricted to their term on the council. Mailing lists are also useful for documenting discussion that may need to be revisited later.

I would highly recommend setting up a mailing list, whatever your community. Before you rush out and do this, though, you need to make a few important decisions about its use:

Focus

Mailing lists are excellent for general discussion. Although they can also be used for voting on an issue, I would instead recommend real-time discussion such as a phone call or IRC channel for that. Many communities have found that voting on mailing lists can take weeks to finalize. With real-time discussion, the voting is much quicker.

Membership

Provide access to your mailing list only to members of the council. Make it clear that when they leave the council or their membership expires, their subscriptions to the mailing list will also expire.

Privacy

Mailing list software packages provide the option to have public archives of the discussion. I strongly recommend that you have private, nonpublic archives. While this may surprise you, I recommend this because when public archives are available, your members will not be as blunt or honest in their feedback. When your council deals with a conflict issue or a membership request, you rely on honest opinions from your members about the individual(s) concerned. This can be tricky if the mailing list is public and a council member wants to share some critical views of that person with the rest of the members. If the list archives are private, you will get a better quality of feedback from your members.

On the other hand, some organizations require open meetings by law. In this case, provide two mailing lists: a public one for most community issues and a private one for personnel issues, which includes the conflict and membership situations just mentioned. This is

comparable to physical meetings where councils ask visitors to leave, and go into “executive session” for personnel issues.

The problem with a mailing list (particularly one with limited membership and closed archives) is that it lacks the transparency and access that your community will rightfully expect. So I highly recommend that you augment your mailing list with regular real-time meetings, preferably using IRC or possibly even an open conference call on the phone. These meetings should be publicized as an opportunity for your community to raise topics with the council.

The choice between IRC and the phone depends on the habits of your community members. From a pure features perspective, IRC is better: it is cheaper, easier to log, and accommodates more simultaneous participants than the phone. The problem with IRC is that only technical communities tend to know much about it. If your community has no idea what IRC is, getting them to use it is akin to trying to make a cat bark. As such, the phone could be a better bet.

NOTE

Of course, if your community is small and local, there is no reason that meetings can’t occur face-to-face in a local coffee shop or somewhere similar. Look at your community and make a decision based upon the norms of how it communicates.

Whatever you decide, you should have regular meetings for your council. The purpose of these meetings is to provide an opportunity for your community to engage with the body that governs it. As such, you should make it explicitly clear to your council members that they should attend every meeting.

A final note on communication is to make sure your community can add items to the council meeting agenda in a simple and organized way. Your community needs to feel that they can raise topics on the council. I recommend having a public agenda visible so that (a) people can add items to it, and (b) others can see what items are planned for discussion and join in if it interests them.

A useful way of doing this is to set up a wiki and simply ask people to update the agenda page when they want to discuss a topic.

Let’s now put a policy regarding communications in place for our Tobe Community Council.

TOBE COMMUNITY COUNCIL COMMUNICATIONS

The Tobe Community Council (TCC) has two primary resources for internal communication and interaction with the community:

- **tobe-community-council Mailing List**—this mailing list, which is by invitation only, is available to all council members but to no one else. Council members are removed from the list when

their role on the council ends. It can be used to raise any issue in private within the TCC. Discussions occur only within TCC members and are not archived or shared elsewhere.

- #tobe-meeting—on the first Tuesday of every month at 18:00 UTC, the TCC has a public meeting that the full community is welcome to attend. The agenda, available at <http://www.exampleproject.org/tobe/wiki/MeetingAgenda>, will be discussed in each meeting. If you want to add an agenda item to the meeting, update that page and ensure you attend. All meetings are logged and available at <http://www.exampleproject.org/tobe/meetinglogs>.
-

Codifying Your Council

We are now rocking and a rolling with a good idea of how our council is going to look. Throughout our discussion we have documented our decisions for our Tobe example, and you should do the same for your own community. You can then gather these notes together and combine them all into the same document. This process is known as *codifying* the council.

When you have this document together, make it publicly visible. You can then use it as a starting point to gather feedback from your community about the proposed council. Notify your community and provide a simple means of them providing feedback (such as updating a wiki page).

This feedback provides an excellent opportunity to clarify any elements that are vague or incomplete in the document. It will also help you to engage with the community to ensure that the governance structure is really supported and reflective of the needs of the wider community. If you develop a Community Council plan in secret and in closed quarters, you will find it incredibly difficult to push it through, and rightfully so. This proposed governance body is a social contract about how your community is run, and the community must not only agree to it but also feel as comfortable about it as they do about that lovely old pair of snug shoes that we all have.

To help illustrate council codification, I'll present the codification document we used when forming the Ubuntu Forums Council. This was a council that was put in place in 2006 to help govern the hugely popular Ubuntu forums. Our codification document received extensive review and refinements, and when most people were happy with it, it was approved by the Ubuntu Community Council and put in place. Here it is.

FORUMS COMMUNITY GOVERNANCE CODIFICATION

The forums represent many people's first meeting with Ubuntu and are an important resource for support and social interactions and have become one of the most important subprojects within Ubuntu. They are the single largest GNU/Linux support forums and one of the most important venues

for community support and interaction. Started independently by Ryan Troy two years ago, their rapid success was officially recognized when they were designated as the Official Ubuntu Forums.

This document aims to:

- Increase recognition of contributions in the forums with membership, which is ultimately used to approve Community Council members.
- Provide a clear delegation and codifications of the existing leadership in the forums and plan for handling these decisions in the future.
- Describe clear democratic and meritocratic processes for the appointment of leadership and staff positions in the forums.
- Remove several “single points of failure.”
- Describe methods for both preventing and resolving any future inter-administrator or inter-staff conflicts within the forums.
- Recognize the hard work of the forums staff through recognition as an integral and *integrated* part of the forums community.
- Provide a straightforward process for top forums contributors to be recognized as full members in Ubuntu, with the right to vote on resolutions posed by the Community Council.
- Provide for a reporting process so that news, ideas and work done in the project by Forums users will be communicated to the broader community and appropriately recognized.

Changes to Current Ubuntu Policy

The proposal includes both new policy and the codification of a few existing Ubuntu policies. These should be discussed with the Community Council and the forum staff. After it has been approved by the Community Council we will add it to the community governance page (<http://www.ubuntu.com/community/processes/governance>) in the Ubuntu website.

Note that the document is structured to describe NOT JUST the Forums, but instead all the areas of the project which are large and independent enough to have their own dedicated leadership structures.

Team Councils

For active teams and subprojects with Ubuntu, the Ubuntu Community council delegates many of its responsibilities to “Team Councils.” These councils act as proxies for the Community Council over a particular team or scope of activity within the Ubuntu community. These governance councils are ultimately responsible for the actions and activity within their team or scope and resolves disputes and manage policies and procedures internal to their team and frequently appoint Ubuntu members on behalf of the Community Council.

The Ubuntu Forums Council (FC) is the team governance council for the official Ubuntu forums.

Forums Council Charter

The Forums Council is the group that is ultimately responsible for the governing of the forums and interfacing between the forums and the rest of the Ubuntu community and governance systems. It will:

- Consist of five (5) members. Membership should be public and published.
- Decisions will be made by a majority of voting Forums Council members when at least three and more than half of the total members have voted.
- FC members should be accessible by and responsive to the forums community (i.e., through a dedicated forum).
- Hold “meetings” regularly and visibly. Meetings can either be in IRC in the “ubuntu-meeting” channel or in a special, publicly visible area or subforum.
- Be appointed by the Ubuntu Community Council in consultation with the Forums Council, forums staff, and active contributors to the forums. Nominations would be open and public and would be considered and evaluated by the CC. Each candidate should prepare a wiki page summarizing their nomination and their contributions and including and referencing testimonials (e.g., something similar to what is prepared for Ubuntu membership). The CC commits to evaluating all nominations on the following criteria, listed in order of importance:
 - The nominees[?] (*essential*) active status as an Ubuntu member.
 - The nominees[?] support from *at least* one active forum staff member (*essential*).
 - Opinions and testimonials (positive and negative) from current members of the Forums Council;
 - Opinions and testimonials from current forums staff;
 - Opinions and testimonials from Ubuntu Members, Ubunteros, and other active participants in the forums;
 - Clear evidence of activity within the forums (quality, quantity and duration);
- Serve terms of two (2) years. FC members could serve multiple or repeated terms. Weight will be given to proved contributors and reelection of consistently active members should be both easy and common.
- Be formed, initially, of the current forums administrators (i.e., Ryan Troy [Ubuntu-Geek], John Dong [jdong], and Mike Braniff [KiwiNZ]).
- Have a chairman with a casting vote, appointed by the Community Council, initially to be Ryan Troy.

The FC would have a number of rights and responsibilities, and be ultimately responsible for the smooth operation of the forums. These include:

- Appointing or recalling administrators, moderators and forums staff or determining criteria by which they are appointed.

- Resolving disputes between forums staff and moderators as per the existing dispute resolution system and forums guidelines.
- With advice, feedback, and help from the forums staff, maintaining and enforcing the Forums Guidelines and associated infrastructure (e.g., the resolution center).
- Regularly and when possible (i.e., monthly), sending reports or representatives to CC members to weigh in on issues of membership and to update the council on the FC business.

Staff and Ubuntu Membership

Forums staff will be appointed by the Forums Council. Forums staff are expected to uphold and set an example that is consistent with the Code of Conduct.

Forums staff and participants have the option to become Ubuntu members. Current staff can apply for membership at an Ubuntu CC meeting. Their contributions as staff members and contributors on the forums should provide more than sufficient evidence of a sustained and significant contribution to the Ubuntu community.

Dispute Resolution

The FC will be responsible for maintaining forum guidelines and systems for internal conflict resolution (e.g., the forums resolution center).

Additionally, there should provide a documented method whereby any disagreements or conflicts *between* moderators can request a hearing by the FC.

In extreme situations, users and moderators who feel that they have not been given a fair hearing by the FC can appeal a decision to the CC. The CC considers the FC to be a greater authority on forums matters and in the majority of these cases, the CC will likely refer these issues back to the FC.

Any deadlock within the FC will can be referred to the Community Council for resolution.

The Ubuntu Forums Council has been hugely successful, and the expectations around its governance have never been questioned. The document helped keep everyone on the same page.

Nominating and Electing Council Members

With a firm foundation of how your Community Council will work and one that has been publicly documented and approved by the community, the next step is to populate the council with members. Earlier we discussed the attributes that we are looking for in members (a listener, unbiased, objective, responsive, attentive to detail, etc.), but now let's talk about how we can find and motivate these people.

Forming a new council

When forming a new council, especially the first council in your community, you typically need to grandfather people in. In other words, the first set of members who govern the council are a handpicked group who you are confident will get the council up and running under the agreement you just codified.

Everyone in your community needs to feel that they have the opportunity to be on the council, but you also want to ensure that you maintain a high level of quality for your council members. A common solution to this problem is to have an election in which your community can vote for those who govern them.

Unfortunately, the big misconception in community elections is that elections alone will sort the wheat from the chaff. In other words, many believe that if you have an open election, the community will settle on the highest quality candidates for the council. This is often not the case.

Popularity contests do not form great governance bodies. Someone without the maturity and vision you need may have a lot of friends and influence and snag a place on the council. On the other hand, there may be a quiet and conscientious community member who is perfect for a position in the council but never gets noticed. In fact, the most qualified community members hardly ever put themselves forward for a governing council, because they are too wrapped up doing their work for the community. Fortunately, governments found long ago an effective compromise between pure representative elections and top-down selection: a nomination process.

Pull together the members you know well, who you feel are the current de facto leaders of the community, and hold a meeting to nominate the appropriate people for a position on the council. The people who can participate in this discussion could be a founder, highly visible contributors, an existing council, or others with experience and insight into the community that you trust. This group should nominate a good range of people, and preferably more people than you have places for on the council. As an example, if you have five seats on the council, try to come up with seven or eight nominations. Each of these people should be an excellent candidate who satisfies the criteria we discussed earlier.

Of course, you should always ensure that before potential members are publicly nominated, you contact them first to ensure they actually want to be on the council. Many people want to have nothing to do with governance. As you talk up your goals, some will come around to realize that serving on the council complements their current ways of contributing and flexes their talents, but others really have no stomach for meetings or rule-making, and should not be browbeaten into serving.

Assume that you'll be rejected the first time you approach each potential candidate. It's almost a given. After all:

- Most people are naturally modest and do not want to claim an honor. Even heroes who rescue people from burning buildings always give speeches afterward saying, “I’m not a hero.”
- The people you want are busy and happy contributing to the community. They’re afraid that joining the council will take time away from their regular contributions (and they’re usually right), and they’re also afraid that governance is relatively dull.
- Most people have never served on a board or council and don’t know what it entails. They assume they don’t have the competence to do so.
- All kinds of cruft around the image most people have of governments and leaders get in the way of their seeing the good things councils can do.

Anticipate all these reactions. They are legitimate, but you have counterarguments to offer. Try to meet in person with a candidate to listen to her views and have a candid exchange. If a face-to-face meeting is not possible, try to use a voice call. The general points you can make are:

- Your community has reached a point where the lack of governance (or in the case where a council is creating subcouncils, the relative lack of attention to one area) is preventing the members from doing what they want. The potential candidate is furthering her personal goals as well as the goals of the community by taking on the new tasks of governance.
- Power is not a bad thing. Whatever goals the contributor has, she can push them forward on a much greater scale by representing her needs on the council and making sure the community provides the resources for these tasks.
- Nobody is putting on a jacket with medals and ribbons; serving on a council is not equivalent to becoming a Generalissimo. Other members of the community will still view the council members as ordinary folks whom they can complain to and have a beer with.
- Some of the greatest life lessons come from serving on governing bodies. Everyone who does it says afterward that they’ve learned an immense amount about people; organizations; and the essential ways that the wheels of life turn in terms of timing, finance, and so forth.
- A council term is limited, and before they know it, they will be back in the rank and file doing the work they originally loved—but this time with a far deeper understanding of this work and how to make it successful.
- Everybody feels the same way at first. The very qualities that make them feel inadequate are precisely the ones that will make them good council members.

Most candidates, once you sincerely hear their viewpoints and offer your own, will come around and agree to serve. But as I mentioned, some are truly ill-suited to council work. If they remain convinced that they’ll be bored, frustrated, or unproductive on your council, don’t pressure them—just express appreciation for the work they’re already doing. You can probably

find a team handling some task that they'd like to work on, and over time they may work themselves through the community structure into a governance position. Next, ask candidates to produce pages that act as a platform for their candidacy. This page should essentially persuade anyone reading it that they are perfect for the role. To make this as simple as possible, you should provide them with a template that they can fill out with their information. Here is a suggested template:

```
Candidacy Document for <name>
Date: <date>
INTRODUCTION
<some introductory text explaining who they are, summarizing their experience and what
they can bring to the council>
COMMUNITY EXPERIENCE
Item of experience.
Item of experience.
TESTIMONIALS
<here other members can add their support for the candidate>
<name> <email> <reason for support for a position on the council>
GOALS
<what they'd like to accomplish in their council role>
```

Each of these candidacy templates should be made available online in the same place so that your community members can review it. A wiki is a great solution.

You can then open up a vote. How you conduct this vote depends largely on how your community is structured. As an example, if you have the concept of membership, such as approved members or approved developers, it is recommended to allow only those approved contributors to vote. This will give all legitimate members of the community a voice, while ensuring that the decision is made by those who actually know the community well enough to offer an informed opinion.

If your community is more loosely formed and just consists of anyone who joins a forum or mailing list, it is more suitable to nominate members from core contributors in the community.

Ubuntu Governance Example

So far in this chapter we have discussed the requirements for governance, created a Community Council, explored the requirements for council members, and chosen the council's members.

As we said way back in [Chapter 1](#), stories are the vessels of great best practices, and with governance such an important part of community growth, I want to devote a good chunk of this chapter to an example of an open, transparent, and effective community that I have been involved in putting together: the Ubuntu community. This case study will give you a firm idea of how one large community has approached it, and the lessons learned.

In the Beginning...

In 2004, Mark Shuttleworth, a South African entrepreneur, founded the Ubuntu project. A longtime user, fan, and contributor to free software, he built his digital certificate company, Thawte, on a free software foundation. When he sold his company to Verisign, he made what can only be described as a “rather large bucket of money.”

After spending a year in Russia training for a quick trip into space as the first South African space tourist, Mark started laying plans for a new Linux distribution. At the time, distributions were nothing new: Red Hat, Mandrake, Debian, and others were already producing comprehensive operating systems that technical types such as myself were not only using but trying to convince our reluctant friends and family to use.

Mark’s vision was different, though. First, he wanted to build Ubuntu on the excellent foundation of the existing Debian distribution. To do this he hired some of the cream of the crop in the Debian community, and they set to work to build a powerful and easy-to-use operating system. His vision did not end there, though. Mark not only wanted to promote an open source operating system, he also wanted to have the operating system be openly governed. Mark knew the importance of community, and he knew the importance of transparency in a community.

At the time, Linux distributions were going through something of an evolution. Linux was beginning to enjoy real commercial interest from the widespread IT industry, and some of the traditional Linux distributors, such as Red Hat and SuSE, were seeing serious adoption. Unfortunately, with all of these business interests, power lunches, and monetization, there was a general feeling (at least as I remember it) that community was getting a little lost in the mix. This was not exactly surprising. The traditional IT industry was built on an understanding that programmers are hired to build your software, and that product managers make the decisions required to direct those programmers and deliver your product.

Linux and open source was different: these companies had to understand that volunteers in bedrooms, basements, and universities had as much input in the direction and focus of that software as anyone else.

We now know that if these community members are welcomed and treated in a collaborative manner, a company can net a huge amount of unpaid development. The buzzwords *crowdsourcing* and *peer production* are widespread in the business world. Unfortunately, many of the companies jumping on the open source bandwagon back in 2004 failed to understand and embrace this community. Instead, they merely *tolerated* it as a politically correct nod to the wider open source community.

Ubuntu was different. Unlike other companies who wanted to keep governance control of the community in their own hands, Mark wanted Ubuntu to be a *pure* community. He was keen that there should be a community governing body and that everyone should be able to join that body based on hard work and commitment to the community.

It is this open community governance that attracted me to Canonical and made me want to become the Ubuntu Community Manager. I was attracted by the premise of a Linux distribution that combined a real open community governance infrastructure with the commercial support and funding of a company. If we could strike this balance, the world would be our oyster.

The Structure of the Ubuntu Community

Like many open source communities, the Ubuntu project is *not* a democracy, but a meritocracy. Contributors are judged in the Ubuntu community on the basis of a *significant and sustained contribution*. It is this simple rule of thumb that determines who takes a place in our community as well as our governance positions.

Today, the Ubuntu community is broken into three approximate layers of governance, shown in [Figure 8-1](#).

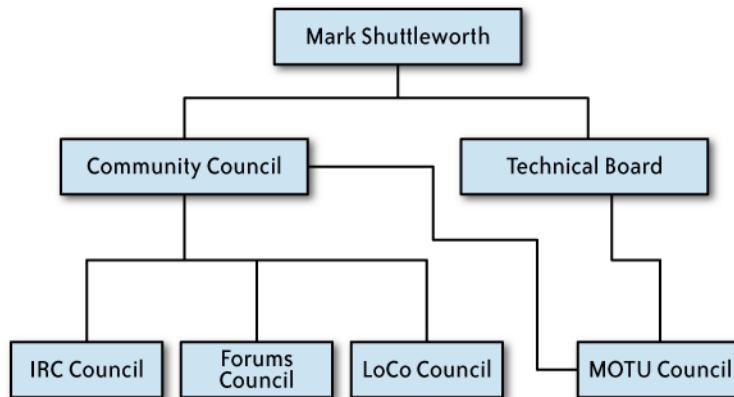


FIGURE 8-1. The governance structure of the Ubuntu community

The community has been divided into four primary governing areas:

Mark Shuttleworth

As the founder and primary sponsor of Ubuntu, Mark is afforded the privilege of a tie-breaking vote and of deciding what his employees at Canonical focus their work on. Although Mark Shuttleworth appears at the top of [Figure 8-1](#), the Community Council and Technical Board do not report to him, this is purely to illustrate his tie-breaking privilege.

Community Council

This is the highest governing council in the community. It makes decisions about how the community is run, changes to processes, and other community-wide issues.

Technical Board

The Technical Board decides on technical process changes and technical policy decisions.

Team Councils

These are specialist councils that govern specific parts of the community.

Let's now take each of these layers for a spin and explore how they fit together in the wider community. I will also kick each of these assessments off with the vital statistics of how that part of the community is structured.

Mark Shuttleworth

Reports to:	No one
Number of members:	1
Responsibilities:	Special role on Community Council and Technical Board

Mark Shuttleworth is the founder and primary sponsor of the Ubuntu project. He has poured millions of dollars of his own personal wealth into the project. In the wider community, Mark has two very distinctive privileges:

- He can assign Canonical employees to work on specific projects, specific feature goals, and specific bugs.
- He also has a tie-breaking vote on the Technical Board and Community Council, should it be required.

Although the first point may be expected, as Mark is the founder and owner of Canonical Ltd, the second point may be a surprise. Tinfoil-hat-wearing conspiracy theorists could see this as a means of Canonical enforcing its will on the community. Certainly from my experience, this is really not the case.

Obviously every community functions best when it can reach broad consensus about a way forward. Collaborative discussion and decision-making is always the desired approach in a volunteer. Unfortunately, it is not uncommon in the open source world for there to be many good arguments, and for arguments to divide communities rather than enrich them. The energy that is focused on these circular debates would always be better spent creating solutions to problems.

If there is a situation in which the community is entirely deadlocked on a decision, with multiple valid arguments for a given position, Mark can use his casting vote to unblock it. Obviously this is not a decision that would be taken lightly, and at the time of writing this casting vote has never been required since the formation of the project in 2004.

Community Council

Reports to:	No one
Number of members:	7
Responsibilities:	General community governance

The Community Council is the primary general governance council in the community. It has been present in the community since the inception of the Ubuntu project and has become a well-respected and instrumental part of the wider Ubuntu community.

The mandate of the Community Council is:

The social structures and community processes of Ubuntu are supervised by the Ubuntu Community Council. It is the Community Council that approves the creation of a new Team or Project, and appointment of team leaders. In addition, the Community Council is the body responsible for the Code of Conduct and tasked with ensuring that maintainers and other community members follow its guidelines.

The way members are chosen is discussed later in the section [“Council or Board Member” on page 258](#).

Its responsibilities are:

- Maintaining the Ubuntu Code of Conduct, which describes the standards of behavior expected of Ubuntu maintainers and other community members.
- Arbitrating disputes under the Ubuntu Code of Conduct. This will happen if a member of the community has asked the Community Council to review the behavior of another member in terms of the Code of Conduct (a document that every Ubuntu member is expected to agree to that outlines basic levels of respectful collaboration).
- Team creation and appointment of team leaders.
- Creation of new structures and processes. A community member who wishes to create a new structure or process submits a proposal to the Community Council for discussion and approval. The Community Council determines lines of reporting and responsibility between different community structures.

The purpose of the Community Council is to provide a group of experienced Ubuntu contributors who have experience working in a range of teams in the community. This group is available to hear any issues, conflicts, or requests that the community would like to raise and request a verdict on. Each council member is expected to vote with a +1 (meaning “yes”) or a -1 (meaning “no”). A majority vote determines the decision of the Community Council.

Importantly, whenever the Community Council sits to decide on an issue, there must be a quorum: that is, a minimum number of members present to form a majority vote for the body.

The Community Council has decided on a range of issues over the years. Some examples of this include:

- The formation of new governance bodies, such as the Forums Council, IRC Council, and MOTU Council (a developer council).
- Approving and rejecting new members. This was later delegated to the Regional Membership Board (more on this later).
- Individual conflicts. Sometimes a member who feels a governance body has been inappropriate in judging a case escalates the issue to the Community Council (more on this later, too).
- The approval of new processes. One example is Monthly Team Reporting for approved teams, which asked teams in the community to provide summarized notes on their work every month.

For each of these issues the format is the same: the issue is raised with the council, it is discussed, and the council members have a vote. The majority vote concludes the decision of the Community Council.

An important consideration has been to make the Community Council as easily accessible and approachable to the community as possible. Any community member is welcome to raise an issue with the Community Council and have it discussed fairly and objectively.

This is how an item is put forward to the council:

- The community member adds the item to the Community Council Agenda, which is done by adding it to the Meeting Agenda page at <https://wiki.ubuntu.com/CommunityCouncilAgenda>. Every Ubuntu community member is freely able to add items to the agenda.
- The Community Council meets twice a month to have an online meeting on IRC. Everyone is welcome to attend, and the meeting (at the time of writing) happens in the #ubuntu-meeting channel on the Freenode (<http://freenode.net/>) IRC network. Meetings happen at 21:00 UTC on the first Tuesday of each month and 11:00 UTC on the third Tuesday of each month. The different times ensure that everywhere in the world has a reasonable chance in their local time zone to attend a meeting at least once a month.
- At each meeting, the list of agenda items on the <https://wiki.ubuntu.com/CommunityCouncilAgenda> page are used as a source of discussion. Each item is discussed and a vote occurs. The majority vote is the concluded opinion of the Community Council.

This process has served the Ubuntu community well. It has provided a simple method in which any Ubuntu community member can raise an issue with the primary governing body of the community.

Technical Board

Reports to:	No one
Number of members:	4
Responsibilities:	Technical and process governance

The Ubuntu Technical Board has a somewhat unique and sometimes misunderstood role in the community. The Technical Board is intended to advise on the technical direction of Ubuntu and usually focuses on wide-reaching critical issues as opposed to the thousands of smaller technical decisions that are made every day in the community.

As with the Community Council, the Technical Board has a published mandate:

The Ubuntu Technical Board is responsible for the technical direction that Ubuntu takes. The Technical Board makes final decisions over package selection, packaging policy, installation system and process, toolchain, kernel, X server, library versions and dependencies, and any other matter which requires technical supervision in Ubuntu.

There is also a set of published responsibilities for the board:

- The Ubuntu Packaging Policy, which describes the standards with which Ubuntu packages must comply. Each release of Ubuntu is associated with a specific version of the Package Policy. Ubuntu community members may propose updates and changes to the Package Policy.

NOTE

Although I try to minimize technical discussions in this book, the concept of a *package* appears often enough to deserve a bit of background. Large projects such as Ubuntu are created by hundred or thousands of teams working on software as different as word processors, graphing tools, games, scientific libraries, and device drivers. Each team produces one or more packages, in which form they distribute their software. One of the critical tasks of the Ubuntu project is to get these packages to work together. Some are more stable and trusted than others, so the packages are organized into different *repositories* to reflect such differences.

- Ubuntu Release Feature Goals, which determine specific features that we aim to include in each release of Ubuntu. These are documented on the wiki pages for each release. Ubuntu community members may propose additional Feature Goals until that release is in Feature Freeze, after which Feature Goals will be deferred to the next release.
- Ubuntu Package Selection, the list of packages that will be installed in a Base or Desktop Ubuntu installation, as well as the list of packages that qualify for full support in the main repository as opposed to the universe repository. (The main repository is more selective

and offers support for its packages.) You may propose packages for inclusion in the Base Install, the Desktop Install, or in the main repository, where they will be immediately available to Ubuntu users.

Some readers may be wondering why the Ubuntu community has a separate Community Council and Technical Board. Surely it would make sense to combine both into a single entity? The reason for two governance bodies was very much a deliberate decision.

The requirements for the governing members in the Community Council and the Technical Board are very different. Community Council members need to have a wide and expansive knowledge of the community, its processes, and its culture. It is entirely reasonable that these members have a more limited technical knowledge of Ubuntu: we don't expect them to be packaging or programming ninjas. Their expertise is based around general community awareness and decision-making as opposed to technical expertise.

The Technical Board, on the other hand, has very demanding technical requirements for its members. Members should not only have a high level of technical expertise, but also should have contextual knowledge of the entire distribution. As an example, we don't expect someone to be an expert in just the desktop, but also the entire underlying system, the toolchain, the bug database, and more. We call these developers *generalists*, and they are hugely valuable members of software-oriented communities such as Ubuntu.

In addition to having a strong technical knowledge, Technical Board members are expected to have a firm understanding of the processes and community infrastructure of the project; keep abreast of the technology and challenges on the horizon outside of Ubuntu; and show strong experience in software development, maintenance, and technical collaboration.

Ubuntu community members can engage with the Technical Board in much the same way as with the Community Council:

- Everyone is welcome to add items to the Technical Board Agenda page at <https://wiki.ubuntu.com/TechnicalBoardAgenda>.
- The board also meets regularly on IRC, and the community is also welcome to attend.
- Decisions are again made with a vote based around a quorum of Technical Board members attending the meeting.

The Technical Board has been hugely successful in providing governance and considered conclusions in some complex technical decisions. Some of these decisions have involved compelling technical decisions, the ethics around software freedom, and other areas.

One such example of a contentious topic was a proposal some time ago to have some advanced desktop effects technology switched on by default in Ubuntu. For this technology to work, it required the computer to have fully working 3D graphics hardware. Unfortunately, back in early 2007 some of the most common graphics hardware would work effectively only if a closed source, proprietary driver was installed. For the effects technology to be switched on, it would

mean that we would need to ship a closed source driver that could be installed if the hardware was present.

This caused enormous controversy inside and outside the Ubuntu community. Many were worried that the core ethic of free software would be compromised if the proposal went forward. For a while the atmosphere got a little tense, and some bloggers assumed the role of prophets of doom, spreading fear of a closed source Ubuntu: “human sacrifice, dogs and cats living together...mass hysteria!” (Sorry, I am an outrageous *Ghostbusters* fan.)

To handle this feisty topic, a joint meeting between the Ubuntu Technical Board and the Ubuntu Community Council was chaired. The issue was discussed, and the combined board united around a common conclusion: Ubuntu would not ship closed source drivers, but infrastructure would be built that would make it devilishly simple to install the drivers if required so that the desktop effects technology could be used. The solution was elegant, combining an unrelenting commitment to providing ordinary computer users with dazzling technology and securing the free software ethics that are at the heart of the project. The community wholeheartedly supported the decision and how it was concluded.

Team councils

Reports to:	Community Council
Number of members:	Varies; typically 5
Responsibilities:	Specialist governance

When the Ubuntu community was born, the Community Council and the Technical Board were the only two governance bodies put in place. They smoothly handled their respective parts of community governance and were productive in that work. As the community grew, though, thousands of contributors joined the project, hundreds of teams were formed, and huge demand was placed on the Community Council and Technical Board. The Community Council in particular began to struggle under the workload, and meetings lasting upward of three hours were routinely organized to handle the sheer volume of requests. Something needed to change.

At the Ubuntu Developer Summit in Mountain View, California, in 2006, it was decided to form a new layer of governance called Team Councils. A limited set of subcouncils would be set up in parts of the project where repeated requests for governance were made. These councils would then govern those areas of the community.

One such example was the Ubuntu LoCo Teams. This worldwide community of 200+ local Ubuntu user groups would get together to advocate Ubuntu, provide support, organize release parties, and more. As this part of the Ubuntu community continued to grow, its governance needs heightened. Decisions about hosting, team naming, documentation, events, team

approvals, and more, were required. So the Community Council decided to form a LoCo Council who would tend to these decisions.

The LoCo Council was officially appointed by the Community Council, and the new LoCo Council members were chosen based upon their years of experience with Ubuntu LoCo Teams. The council went on to successfully govern the community.

Similar councils were set up for other parts of the Ubuntu community, including the IRC Council, Forums Council, and MOTU Council. In each case, the Community Council appoints the members, although for new members of existing councils these appointments are often based upon recommendations from the existing council. We discuss how to set up councils such as these in more detail a little later in the chapter.

Membership

Inside the Ubuntu world it doesn't take long before you see references to "a member of our community" and the prospect of "joining the community." The concept of *membership* is something that varies tremendously across different communities. For some communities, membership is merely taking an interest in what active members are doing, whereas some communities have a more formalized style of membership that requires some kind of approval process.

In the Ubuntu community anyone is welcome to join and participate, and formalized membership is not required. This ensures that the community is open for people to join and dip their toes in. This is great for people new to Ubuntu, but we do have some more formalized membership roles for those who want to provide a more concrete level of contribution or those who want to contribute to critical parts of the system that require approval. These roles are:

Ubuntu Member

A basic level of approved membership that indicates that a contributor has provided a reliable level of contribution. This role doesn't depend on the type of contribution: people can become a member if they do packaging, development, translations, documentation, advocacy, or whatever else is recognized to be of value.

Developer

For those who require upload access to our primary software repositories, this role is required. Candidates must adhere to a more complex set of technical requirements.

Governor

This is a role that applies to anyone on a governance body in the community. This role requires a demonstration of the attributes of a great council member, discussed earlier in this chapter.

Each of these roles has a different set of expectations and requirements. Let's now delve into each of them in more detail.

Ubuntu Member

This is the first level of formalized membership that someone would shoot for upon joining the project. This role essentially guarantees that the contributor has demonstrated effective participation in a large distributed community such as Ubuntu. Each prospective member must sign the Ubuntu Code of Conduct.

To clarify what this role means, a mandate was defined at the inception of the project:

Membership in the Ubuntu community recognizes participants for a variety of contributions, from code to artwork, advocacy, translations and organizational skills. If you are active in the Forums, or submitting icons or sounds or artwork, then you are eligible for Membership, which gives you a say in the governance of the project.

One of the reasons we have the concept of an Ubuntu Member is to help the project identify reliable contributors. Anyone who is approved as a member is likely to understand the basics of what is involved in participating in a distributed project such as Ubuntu. Although this level of membership does not guarantee quality of work, it does provide a fairly reliable baseline for engagement: it shows that the contributor has a history of providing sustained contributions to a community. This in itself is useful for a range of purposes, such as soliciting input, voting, encouraging participation in other parts of the project, and mentoring new community members.

Users apply for membership through the following process:

1. The user must have a profile on Launchpad, where people engage in Ubuntu work such as packaging, translations, bug triage, and more.
2. An application document is created on the Ubuntu Wiki. This application should include some standard items: a summary of contributions to Ubuntu, a link to the user's Launchpad profile, a complete description of contributions to Ubuntu, plans and ideas for Ubuntu in the future, and testimonials of other Ubuntu Members who support the application.
3. The user signs the Ubuntu Code of Conduct. This ensures that the contributor agrees to participate under positive rules of engagement in the community. The vast majority of reasonable and self-aware contributors automatically meet all of the criteria from the Code of Conduct.
4. The contributor then adds her application to the next Regional Membership Board meeting. There are three boards—Americas, Europe, and Asia—and each board votes on applications from their regions. As an example, an applicant in Paris applies to the European Membership Board.

5. The user must attend the meeting (which takes place online) where the board members review the application and ask any relevant questions. After a review, the board votes, and a majority vote is considered approval for membership. If the member is not approved, she is welcome to apply again at a later date.

Approved members also receive some privileges:

- An @ubuntu.com email alias that forwards to your real email address.
- An `ubuntu/member/your_nick` hostname IRC cloak.
- The right to print business cards with the Ubuntu logo.
- Syndication on Planet Ubuntu of their blog.
- An Ubuntu Member title at the Ubuntu Forums.
- A free subscription to the *Linux Weekly News* website.

The concept of Ubuntu membership proved to be hugely valuable, offering a reliable assessment for a person's commitment to Ubuntu and ability to contribute. This approach to membership could be applied easily to most communities.

Before I wrap up this section, I want to share an important note: despite Canonical investing millions of dollars each year in Ubuntu, this does not fast track employees or step over any existing community processes. I myself am an example of this. When I joined Canonical as the Ubuntu community manager, I was expected to step before the Community Council, present my application, and apply for Ubuntu membership. Fortunately I got it. It could have been a little weird if I didn't!

MEMBERSHIP EXPLOSION? SCALE IT UP!

Traditionally, membership in the community was judged by the Ubuntu Community Council. As Ubuntu continued to grow in popularity and the community grew at a similarly frantic pace, the Community Council found itself overstretched in its ability to keep up with applications.

This is the reason the Regional Membership Boards were set up: they offered a more scalable solution to the sheer growth of the project.

Developer

Development is a critical part of Ubuntu. It is developers who take the thousands of open source applications and tools available online, package them, and make them available for easy installation and use on an Ubuntu system. These developers also fix bugs, deal with security issues, and ensure that these applications and tools integrate properly in the distribution.

Anyone is welcome to apply for a developer role in the Ubuntu community. There is no requirement to work at Canonical. In fact, many of the top-level developers who work on the most significant parts of Ubuntu are volunteers. Having this level of openness has been a huge boon for transparency in the Ubuntu project, but it has also required a carefully thought-out process for assessing developers, in order to maintain packages' standards of quality. These standards of quality are equally applied to all prospective developers, irrespective of whether they work at Canonical.

I'll talk briefly about these developer roles as they exist at the time of writing, but it should be noted that we are currently going through something of a change and adjusting the processes behind these roles. As such, I will discuss the current published processes to give you a good idea of an approach that has worked well since the beginning of the project. Just don't be surprised if you look into the Ubuntu community at a later date and the landscape is a little different.

Ubuntu is split into a number of different repositories. These are the servers that contain the collections of packages that form the Ubuntu system. There are a number of repositories, but two primary ones:

Main

This is the primary, officially supported repository. All of the applications here are considered critical to Ubuntu. They receive security updates, and Canonical also supports them in its commercial support services. This repository contains everything that appears on the Ubuntu installation disk.

Universe

This repository is the entire archive from Debian (which Ubuntu is based on) but with packages built against Ubuntu. In other words, this repository contains the same software as Debian, but tweaked and tested to make sure it will run on Ubuntu. These applications are unsupported: end users are responsible for checking on security updates themselves, and Canonical does not provide support for *universe* packages in its commercial support services.

Each of these repositories is matched to a specific developer role:

Main→ubuntu-core-dev

Developers who have direct upload access to *main* are called *Ubuntu Core Developers*. These developers demonstrate extensive packaging ability across a wide range of areas in the Ubuntu system. To apply for this role, the developer must have produced a significant number of packages that have been reviewed by existing developers and demonstrated a consistent level of excellent work. The Ubuntu Technical Board is the governing body that makes decisions about who gets approved as an Ubuntu Core Developer.

Universe→ubuntu-motu

Developers who want to build packages for *universe* are called *MOTU developers*. MOTU is short for...wait for it...Masters of the Universe. Yes, the Ubuntu project loves He-Man.

Membership in MOTU is less stringent than for Ubuntu Core Developers, although a substantial competence is required. The process for applying for MOTU is to have a number of packages reviewed on the sponsorship queue (as discussed in [“Reviewing new developers: In depth” on page 107](#)) and then to put forward an application to the MOTU Council for approval.

Both of these roles require a consistent grade of quality. Having direct access to the *main* and *universe* repositories means that developers can have direct and lasting consequences on millions of Ubuntu users around the world.

WHY CHANGE?

Earlier in this section I informed you that we are currently changing the processes around how people join the Ubuntu community as developers. The reason for this is that we are streamlining the project to make it easier for people to get involved. This is intended to solve some problems with the current approach:

- Today we have two classifications of developer—Ubuntu Core Developers and MOTU Developers—which determine who works on which repository. We should instead have one classification of developer and determine which parts of Ubuntu they work on as permissions within their role.
- Right now if you become a developer, you get access to the full repository. We would like to provide access for contributors to work on specific parts of the archive.
- We are also adjusting some of the technical processes controlling how contributors participate.

These changes are ongoing.

Council or Board Member

The last type of membership role we will discuss here is that of a member on a governing body such as the Community Council, Technical Board, or Forums Council. How did the Ubuntu community approach identifying suitable community members for these different governance bodies?

For most new councils, the primary council members are often handpicked to get the council off on a strong footing. This happened with the Community Council, Technical Board, and most of the team councils, such as the Forums Council, IRC Council, and MOTU Council. Typically, the first generation of these councils is picked by the highest governing body. As an example, the Community Council was formed by Mark Shuttleworth and the forefathers of Ubuntu. Later councils, such as the Forums Council, had their initial members nominated by the Community Council.

When the council term ends, a new generation of members can be nominated. In most of these cases existing members are welcome to stand again for nomination, and there is an open call for other contributors to propose themselves for membership in the body.

Escalation

Every community has conflicts and other problems, and I devote [Chapter 9](#) to these issues. Before we get there, I want to discuss how the Ubuntu community handles the escalation of problems and conflict because this is a very clear responsibility that falls under the wing of governance.

The challenge here is simple: if there is a problem, how does someone get advice from an objective and more experienced third party? What then happens if the person with the issue feels that the objective third party was not objective at all? What happens next, and how is the issue escalated?

Our goal is to create an escalation path that starts at the most immediate decision-making body and then progressively pushes the issue further up the chain if a satisfactory conclusion is not reached. The primary reason why escalation is important is that you don't want to trouble the senior levels of governance unless absolutely required.

Let's look at an example. The Ubuntu Forums community defines an escalation path that looks like [Figure 8-2](#) (the process begins at the bottom of the diagram).

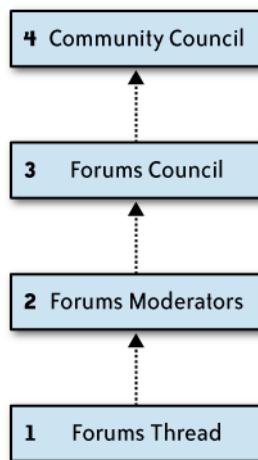


FIGURE 8-2. Forums escalation path

Imagine that someone is frustrated that one of his discussion threads in the forum was deleted because it was considered offensive. This is how the escalation path works (correspond each step to the diagram in the figure):

1. Forums Thread

The first place there is likely to be conflict is the main forum. The user will undoubtedly complain loudly, and other forums members will try to discuss the issue and calm the user down. If the contributor wants to escalate this further, the next stage is reached.

2. Forums Moderators

This is a special part of the Ubuntu Forums where users can take complaints. On this level, the forums moderators, who have insight and administrative control in the day-to-day runnings of the forum, can comment on the decision made in the original forum. Hopefully at this level the insight of the moderators is considered suitable third-party input and the issue can be settled. If the user still considers the issue a problem or feels the moderators are not judging it fairly, it can be escalated further.

3. Forums Council

The issue is now taken to the Forums Council. The council not only oversees the governance of the wider forums but is also there to judge the effectiveness of the moderators. At this level, the Forums Council members should assess the behavior of the moderators in addition to the issue itself. If the Forums Council considers the matter to be handled fairly and objectively by the moderators, the issue should be considered closed. If the user is still not happy, however, he can then escalate the issue to the final level.

4. Community Council

At this point the user is effectively saying that the forums moderators and Forums Council are all doing a substandard job in assessing the problem. The Community Council should now assess the verdict of the Forums Council as well as the issue itself and pass a verdict. If at this point the user is still unhappy, he is going to remain unhappy. But four levels of governance, each embodying a wealth of experience and insight into the community, have now weighed in on the matter.

Expanding Governance

One of the most wonderful attributes of communities is their ability to surprise. From tiny acorns great trees grow, and the branches of community often spread far and wide.

We have seen this countless times, be it Ubuntu, Wikipedia, the Barack Obama campaign, or other communities you may have joined. When your community manages to captivate and develop mindshare, not only does the focus of the community grow but so does your membership. More of these people want to be a part of the journey that your community is on, and they often share the same passion and devotion to reaching the final destination as you do.

Although no one is going to quarrel with this growth, it can generate a headache: how on earth do we scale our governance bodies up to manage this larger group of people?

Fortunately (in terms of hard problems) or unfortunately (in terms of not many people joining your community), this is a problem that only some of you will need to face.

Knowing When It Is Time

In an ideal world we would not need to set up any governance in the first place, let alone set up additional subcouncils. Each additional level of governance that you add to your community is another step away from being a simple and nimble entity that is clear for everyone to understand. It gets a lot more icky when there are Community Councils, team councils, delegated bodies, and other such pomp and grandeur. Your community should set up this additional governance only if *absolutely necessary*.

Of course, many communities need multiple levels of councils, and we certainly discovered that in the Ubuntu community. Despite its seeming complexity, I know that every piece in the Ubuntu community is *absolutely necessary*, and that is our ethos here.

Additional governance is required only when existing governance bodies are not able to scale or cater to your community's requirements. Unfortunately, some communities don't quite get this right and decide to set up vanity councils: governing bodies that achieve nothing more than making the members of the new council feel special.

This can happen in the simplest of scenarios. As an example, imagine the Tobe project that we discussed earlier really took off and became the hot new thing. If it is like any other successful community, a raft of additional resources will be created for the project. One typical example would be discussion forums. When the forums go online, we would likely see the same thing that has happened in other communities: a very passionate yet vocal community makes its home in the forums.

Although the Tobe Forums community may be small, the vocal few in the forums want to feel special: they want a governing council. In their eyes, governing councils bring all kind of fun things: a sense of control, authority, and power across the wider project. Not only this, but they see a council as required for the community to be a *real* community.

Nonsense.

There is simply no reason for this council. The Tobe Community Council already exists as a place to resolve issues, and the wider Tobe community is still fairly small. By creating this council you will effectively double the risk of bureaucracy.

Here are some indicators of when it might be time to reopen this chapter, blow the dust off, and read how to build a council again, but this time a subcouncil:

Bottlenecks

Many councils face bottlenecks. These typically occur when the existing council is either failing to organize meetings, a few people on the council are holding decisions back, or there is too much work for the council to do. The first two issues should be solved by fixing your existing council. For the third issue, you may want to first see whether the existing council can commit more time to the council. If not, another council may be required.

Specific knowledge required

When you form a large subcommunity in a specific area of your community, that subcommunity may require some specific knowledge that your general Community Council does not possess. When you see many issues occur that require this knowledge, a subcouncil may be required. One area that often calls for specific knowledge is conflict resolution in specific parts of the project. As an example, conflict in a developer community will require knowledge of both the people involved and the development tools and processes.

Separation of skills

When you have a general Community Council up and running, you may find that a number of requests fall into one specific category and that these requests need more focus and experience. An example is if you get lots of technical requests that require extensive knowledge. This may warrant the formation of a Technical Board.

Locale/language

If you have a large community forum around a specific language or locale and the people there are struggling to communicate with the main Community Council, a regional council or language-focused council may be required.

Each of these situations exceeds the time, knowledge, or skills of the existing governance body. The quantity of requests is the justification for the subcouncil: if you get only a few domain-specific requests that your Community Council struggles to deal with, a separate council may not be required, but if the council gets regular and repeated requests, it is worth considering.

There are many examples where the issues just described resulted in the formation of councils. One such example, mentioned earlier, was the formation of the Ubuntu Membership Boards. In terms of expertise, the Community Council was capable of handling each membership request, but the sheer volume made it unfeasible. These three subcouncils were formed to cover the Americas, Europe, and Asia, and took over the responsibility of reviewing Ubuntu membership requests.

Another example was the formation of the Forums Council, triggered by the overwhelming demand for governance and forums experience in the forums community. Similar considerations also spawned the Ubuntu MOTU Council, which governs the developer community that builds packages for parts of Ubuntu. The council was formed from existing

MOTU developers to provide a better level of knowledge for assessing new developer requests. All of these examples have ultimately been successful in governing their respective parts of the community.

Building the Subcouncil

Fortunately, once you have been through the process of building your main Community Council and have justified to yourself and your community that a subcouncil is required, forming this new subcouncil is relatively straightforward. All you need to do is repeat the steps earlier in this chapter that explained how to form a new council.

While following the process you should take extra care to codify the council thoroughly. You should consult heavily with your community and expect a barrage of questions seeking to justify the new council.

Your community is going to be nervous about new governance, and you will likely notice a slight paranoia that some of their rights are going to vanish as they perceive the cold chains of bureaucracy clanking down hard on the community. If you've followed the careful thought processes I've described, such concerns are ridiculous.

We are all here to show bureaucracy who is boss, and ensure it has no home in our communities. Make sure your community knows this. This will require ensuring that your codification is easy to read and understand, and regular reassurance to your community about the scope and responsibility of the new governance body.

When you have a broadly acceptable codification in place, the next step is to run it past the existing Community Council. You should expect the document to go back and forth with the council a few times before it is considered complete. When this is ready, you are all set to ask the Community Council to nominate members for the new council and optionally have an election with your community members to select between these nominees.

Escalation

With at least two councils in place, you should now ensure that your community is entirely clear on how each council works and also how they fit together. More specifically, your community needs to know how issues can be escalated from one council to another.

To do this, you should write an *issue escalation document* that illustrates clearly how an issue should flow through your governance structure. Here is an example of how this document could be applied to the Tobe community with its Community Council and Forums Council.

TOBE COMMUNITY FORUMS ISSUE ESCALATION

In the Tobe community, we have two governing bodies that are available to resolve and manage issues. These are:

- Tobe Community Council—this council is the highest serving governing body in the Tobe community. It discusses and decides on general community issues, policies, and processes. This council is also the top-level platform to discuss conflict issues.
- Tobe Forums Council—the Forums Council governs the Tobe forums community and reports to the Tobe Community Council.

If you have a problem or concern, these governance structures are available to listen and pass judgment on the issue in question.

If you have a FORUMS issue, you should first add your issue to the Tobe Forums Council agenda by clicking [[here](#)]. Do not take your issue to the Tobe Community Council; you will be immediately asked to refer to the Tobe Forums Council first.

If you are dissatisfied with the service you have received from the Tobe Forums Council, you are welcome to escalate the issue to the Tobe Community Council. Please note that by doing this you are effectively stating the following:

- You believe that either the issue was not fairly and objectively considered by the Tobe Forums Council or that the council members do not possess the knowledge to effectively pass judgment on the issue in question.
- You are happy for the Tobe Community Council to perform an assessment of how effective the members of the Tobe Forums Council were in assessing the issue.

To escalate the issue, add an item to the Tobe Community Council Agenda by clicking [[here](#)].

We discuss how these escalation incidents are specifically addressed in [Chapter 9](#).

Communicating Between Councils

When I originally discussed how to build a new council, I stressed just how important it is that you create communication channels for the council. This is an expected resource in each and every council that your community puts together. It allows the council to share ideas, deliberate on topics, and reach decisions.

Friends, the communication love doesn't end there, though. Now we must put our devious little minds to another opportunity: how can we help our multiple councils communicate with each other? The reason and inspiration behind this shared cross-council communication are those two magical words in community management: *best practice*.

If you have two councils that are formed (e.g., a Community Council and a Forums Council), you essentially have two groups of motivated, responsible, objective, and reliable people. Although each member on these councils should have a minimum level of suitability for the council based on the attributes that we discussed earlier in this chapter, there is oodles of potential for these council members to learn from each other and further improve and refine their governance abilities.

To do this, I recommend you set up an additional mailing list on which you ask every council member in your community to be a member. I know, I know, you are probably getting a little sick of setting up all of these mailing lists, and I am by no means encouraging *mailing list fetishism*, but this cross-council list will be a valuable resource. Another added benefit of this list is that it lets you address all of the governing members of your community in one place, which is helpful for sharing best practices.

Although it may seem a little less-than-transparent, I would again recommend that this governance mailing list be a closed, members-only list with closed archives that only the current council members in your community can access. The reasons for this are the same reasons behind the recommendations for each council to have a private mailing list: for honest and frank conversation to flourish, they often need a context that is unmonitored.

Summary

In this chapter we have covered one of the most critical components in community management. Although governance has the ability to glaze eyes, it is a key ingredient in producing a healthy, vibrant, and accessible community.

Of course, governance is a large and complex topic, and the academic brigade has written many books on the topic. I have tried my damndest to cover this sometimes-dusty topic as simply as possible and to give you an overview of the primary elements that are important in most communities. This chapter should give you a great head start in producing a governance system that is simple, unobtrusive, and fair to your entire community.

Handling Conflict

“The people to fear are not those who disagree with you, but those who disagree with you and are too cowardly to let you know.”

—*Napoleon Bonaparte*

BACK IN 2007, I WAS PACKING MY SUITCASE to head to a conference over in Europe. Tired of trying to cram everything into a tiny bag, I decided to check in on my email, where I had received a somewhat concerned note from a member of a user group.

The group (who shall remain unnamed) had been experiencing some rather ugly conflict between two members fighting for leadership. Both individuals felt they were the better choice for leadership but were rather ironically demonstrating their clear lack of leadership experience by having a public spat in the interests of securing power. The poor soul who sent me the concerned email informed me that the situation was making the user group a really not very fun place to be. I responded, but in the interest of not missing my plane, I needed to go back to resolve my own conflict between my underpants and travel bag.

A little while later, another email arrived about the same group, sharing similar concerns and calling out one of the leaders for being unprofessional, bullish, and impolite to the members of the group. While I was reading the email, another arrived. This one (replete with flurry of uppercase, highlighted, and bolded statements) defended the actions of the leader I was originally reading about. In contrast, this writer accused the other leader of corruption and

misconduct in the group. Somewhat concerned, I replied to each email to say I would be in touch but had to literally run out the door to get into the cab that would take me to the airport.

As I sat in the cab, I pondered how I was going to handle the situation. In between the barrage of opinion spurting from the cabby's somewhat foul mouth about how the economy was on the brink of melting, I was somewhat preoccupied with trying to flesh out a solution to the complex situation that confronted me. (Retrospectively, it seems that cabby wasn't as much of a whack job as I first thought....)

Between getting in the cab, getting to the airport, and getting online, all of which took about 45 minutes, I had received another six emails from the same group, all from different people. Clearly something was very, very wrong there. Each of the emails had a strong opinion. Some correspondents supported one leader or the other, but most were sick and tired of the arguments, shouting matches, and inability to listen. A community that was once rewarding and fun to be a part of had turned into a roller coaster with a chunk of track missing.

What happened next was the largest conflict resolution experience of my career to date. It resulted in a slew of meetings, soliciting feedback (which generated more than 60 emails in response over the course of a single day), and driving forward to a solution that sought to satisfy both of the proposed leaders and the wider community. We got there in the end, but it was a tense few weeks in an environment in which carefully chosen words, detailed next steps, and patience all took center stage.

The Nature of the Beast

Conflict is part and parcel of life. We see it every day on television; hear it every day on the radio; and read about it every day in our email, social networking websites, newspapers, magazines, and books. Most of the time, Louis Armstrong provides the soundtrack to our wonderful world, but every so often life gets a little edgy and our soundtrack is replaced with Slayer.

You should absolutely expect conflict in your community. There is not a community yet that has gone unblemished by conflict, and its presence is no reflection on your community, its members, or its leaders. You could tirelessly spend each and every day making the tiniest of considerations and changes, refining your processes and governance to the nth degree, and ensuring you are regularly checking in on the happiness of your members, and you will still find conflict lurking somewhere.

The reason is simple: people are people, and sometimes people just don't get on. You may have two people who have similar interests, similar backgrounds, and similar perspectives but just rub each other the wrong way. People are big bags of variables: different cultures, opinions, approaches, ideas, values, and more. When one big bag of variables doesn't match another equally big bag of variables, spats, arguments, and fuss ensue.

There is a science out there that explains how conflict occurs, but it is grounded in this plethora of variables, stimuli, nature-versus-nurture debates, and other elements. It is possible to devote your life to the topic: there is a sea of content about the psychology of conflict, anger management, cultural impact, expectations, and negotiation skills. Although you are welcome to submerge yourself in this academia, much of it will not be particularly useful when trying to figure out how to untwist the knickers of two people caught up in a fracas.

As a general rule, conflict is rare, and it doesn't need a lifetime devotion to the library of academia. What it needs are straight, practical, hands-on approaches to dealing with common situations. With this in mind I wanted to include this chapter as a summary of the most important things to know when dealing with your community's conflict. It will give you the tools for handling the level of conflict you are likely to deal with.

The Structure of Strife

Let's kick off the chapter by exploring some of the high-level elements involved in conflict. The following are three fundamental ingredients that are often present in a conflict:

Confident/free-thinking personalities

Typically the participants in the conflict are not exactly wallflowers. They are often strong personalities who are not afraid to speak up when they are unhappy.

Incompatible goals/values

There are invariably one or more goals or values that the participants disagree on. Importantly, these are typically *perceived* goals and values. The reality is often very different, as we will explore later in this chapter.

Interaction

The fact that two strong personalities disagree on goals or values does not in itself cause conflict; it is the manner in which these personalities interact that causes the sparks to fly. The source and nature of the interaction is often a key component in the conflict.

Conflict is a little like really bad food. There are many strong flavors out there (these are our personalities), and many of them have distinctive purposes (these are our goals/values). When the ingredients are in their boxes and bottles in your kitchen cupboard, they are innocent and innocuous. However, putting them together in the same dish can potentially create something that leaves an unpleasant taste in your mouth for a long time. On the other hand, strong flavors put together in the right way can complement each other and produce stunning outcomes and tastes that become memorable, long-lasting, and incredibly enjoyable.

Rather unsurprisingly, conflict does not make for healthy community. Conflict is the acid that slowly erodes away community: it causes uncertainty that thrives in an uncomfortable and unpleasant environment. With a community generally populated with volunteers, it doesn't take a rocket scientist to understand how unappealing this kind of environment can be.

Although an irksome aspect of human behavior, conflict does offer us valuable insight. It helps us to identify the personalities in a community, it demonstrates a sense of passion in your members, and it shows very clearly that your community is alive and thriving as opposed to a lifeless husk. Conflict is always going to be present in a thriving community, and strangely it is one metric of success.

This reminds me of an incident a few years back when I wrote a blog entry on <http://www.jonobacon.org/> and received an awfully mean-spirited and nasty comment: personal, full of vitriol, and entirely unnecessary. As soon as I read it I became very self-reflective and worried that the statement may have represented the views of more than that individual. In doing so, I entirely failed to balance the picture and consider the countless pleasant comments and wonderful email that I received. My mind zoned in on the negative.

In the midst of all of this, I logged on to discover a message from my friend Christian Schaller:

Hey Jono. I saw that comment on your blog, you must be delighted!

When I asked in what possible dimension that comment would make me happy, Christian responded with:

For someone to write that it means that they care what you think, so much so they felt the need to write it on your website. Congratulations!

In a strange and twisted way, Christian was right. When people care about something, it will often inspire them to take great lengths to protect against something they disagree with. When this happens, conflict brews.

As a community leader, your community will look to you for guidance and advice when conflict rears its ugly head. You need to be prepared to step in and provide security and confidence. You need the skills to handle conflict in a way that is professional and reasoned, and subscribes to the underlying values of your community.

Conflict resolution in a volunteer community is very different from conflict resolution in a formal organization such as a company or a government agency. Community conflict often requires more sensitivity and restraint. The risk associated with putting a foot in the wrong place with community conflict resolution is that any party who is unhappy with your solution may well leave. If you have to deal with conflicts too often, you may end up losing many great members in your community. In more formal settings, the likelihood of employees leaving after conflict is much reduced, as they have to earn a living and put food on the table for their family.

The Calm Before the Storm

Conflict resolution is a skill that is heavily grounded in experience. Dealing with and handling conflict teaches a new lesson every time, and these lessons help define an approach, a

sensibility, and a method. This experience has also been shown to develop one significant skill: the ability to detect and preempt conflict before it even happens.

Conflict does not just appear out of nowhere. A series of utterances and interactions occur, each one evidence of the brewing storm, before the conflict ultimately reaches the somewhat irksome plateau of a spat. If you can see and detect these indicators, you have the opportunity to step in before the going really gets rough.

Although every community is different, many of these warning signs are shared across all communities. Let's now take a spin through some of these indicators of a conflict on the horizon. As we explore these topics, we will end each discussion with a quick and dirty set of preventative bullet points that you can note down to help avoid these issues in the first place.

Contentious Personalities

One of the most delightful attributes of community is its ability to energize and enable people. Without having any prior experience or reputation, many go on to develop expertise and respect by starting at the beginning and leveraging the opportunity of community to do amazing things. One underlying attribute that makes this possible is the openness involved in community: everyone in a volunteer community is welcome to join and participate. With such openness as the norm, the exceptions to the rule—when people who actually get rejected from a community—are usually those who perform gross misconduct that socially disappoints the wider community. Thankfully these kinds of incidents are rare.

If we draw a conceptual line that puts the ideal polite and engaged contributor on the far left of the line and the frustrating, imbecilic clod on the far right, there is a significant scale in between where the vast majority of your contributors will find their home. Those personalities who edge further toward the right of that line are prime candidates for causing frustration and conflict. Let's explore this in more depth.

Profiling the polemical

In my time working with community I have experienced a range of personalities that have been controversial, disruptive, and at times outright infuriating. Within this realm, it is tempting to shun each of these figures as undesirable members of your community and try to disengage with them in the hope they will move on to annoy someone else.

In many cases, though, this is a crying shame. Some community members can appear irritating and troublesome at first, but when you give them the benefit of the doubt, you and they can often grow to work well together. This all boils down to understanding personality differences and the causes behind them.

When you deal with conflict, you are invariably dealing with cases in which someone just doesn't like someone else, largely driven by personality differences. If you can understand and

reconcile these differences, progress can be made. Let's first look at some of the reasons why people may appear vexing to others:

Age

Although many will deny it, age can dramatically affect how people engage in community. This can vary from methods of communication (younger people often shorten and codify their language, such as with "txtspeak" and "lolspeak") to different values, perspectives, opinions, and approaches. In many cases younger people are more trigger-happy, adventurous, and argumentative than older people, and this can be a prominent source of conflict. As many people get older, they become "set in their ways" and less likely to change and reflect on their views. Although this naturally does not affect everyone, it does apply to many.

Culture

With so many communities living online and garnering an international audience, culture can play a big role. This can affect temperament, preparedness, approach, and other elements. Culture can be a subtle contributor to conflict: often these differences are difficult to spot unless you are intimately familiar with both cultures. This is also a sensitive topic, as the acknowledgment of cultural differences can often be wrongly interpreted as racism. Tread carefully, Obi-Wan.

Opinions

It is interesting how some people just seem to have more opinions than others. These people can be both a blessing and a curse. If you put two of these opinion-laden folks together, things can get fiery quickly. The subtlety here is not in whether someone has an opinion but how constructively that person can express it.

Experience

The level of each person's knowledge of the subject and the community itself can play a huge part in conflict issues. Sometimes those with more experience in the community expect that those with less experience to know more, and consequently frustration builds. Not good.

The interesting commonality among all these attributes is that people can and have been known to mature in each of these areas. Understanding this maturity can be the key to understanding your wider community and its personal development better. It can also be an important topic to communicate when things get contentious.

Community participation is a little like starting a new career. In the beginning, you make unintentional mistakes. Failing to see the mistakes, you need to have people point them out for you so you can learn. The reason for these mistakes is not because you are stupid, but simply because you don't know the ropes yet. When you learn how things work, the mistakes usually decline consistently until they become rare blips in an otherwise perfect signal.

From a community leadership perspective, it is essential to understand the importance of allowing people to mature. You are going to come across some people who frustrate you. Some

of these people will make every mistake in the book, and at times it will make you question your own faith in community (“How can we ever achieve our goals with people like this in the world?”).

To really *feel* this growth in people, you need to experience it. After a few years of conscious awareness of how people mature, you are sure to see and experience examples of it happening in your community. For example, a guy joined a user group that I had formed and proceeded to break every possible rule, social convention, and principle in the group. He frustrated many, some of whom lit their torches and called for his ousting. Although there was little doubt that he was a frustrating force in an otherwise calm community, I had a hunch that he had the ability to change. I held strong, encouraged patience among my fellow community members, and before long we started to see improvement.

As the guy spent more time in the community he learned how it worked, began to take part in the culture as opposed to questioning it, and eventually became one of the most proficient and well-respected members. Today others look to him for advice and guidance; it just took him a little while to get there.

Poisonous people

While some people are able to grow and mature, some outright bad apples may unfortunately join your community. Some of these people will be obvious (posting loud, obnoxious, or offensive comments), but some will be far more subtle and devious.

The latter are often known as *poisonous people*. These are the people who not only express dissatisfaction with your community or direction, but also actively seek others to join their campaign of negativity. These people are not only interested in expressing their own concerns, but also want to build a coup of counteropinion against your community.

Many of these poisonous people often operate under the radar. The conversations will typically happen in private with only one side of the perspective represented, but their actions will lead to substantial conflict that threatens the entire community. Try to be aware of who these people are. Typically their names will come up in conversations where it becomes clear they are expressing private concerns and negativity with multiple people. Expressing concern does not make people poisonous, but privately rallying support against the community without any discussion or debate of the issues in the common communication space *does* make them poisonous.

You should handle these known entities with caution: jarring moves in their direction can dial up the poison even further. You need to fight poisonous people not with words but with evidence. Disprove their comments with calm and reasoned commentary amply bolstered with third-party references and evidence. You can then let your community members make up their own minds.

TIPS FOR AVOIDING CONFLICT

- **Understand the factors that can affect how people conduct themselves in a community.**
- **If someone is causing conflict, assess how much you feel they could mature. If you have faith, encourage the detractors to share that faith.**
- **Identify poisonous people and watch them carefully. Give them additional attention to limit their damage: identify what their concerns are, discuss them, try and weed out any communication errors, and generally try to calm their concerns and worries.**

Barriers to Input

One of the most basic concepts in most volunteer communities is *openness*. For a volunteer community to thrive, openness secures the impression that your community members can have a tangible impact on the direction and focus of the wider community. You always want your community to thrive on the opportunity to discuss, debate, and reason, and for their views to be listened to and acted upon where appropriate. In other words, your community should always feel that their input is welcome. It may not be acted upon, but it should be received in good faith.

As we discussed back in [Chapter 8](#), not all communities are created equal. Some are driven in a dictatorial fashion by specific people or by handpicked rock stars. Some communities are far more open, and some are leaderless and driven by consensus and action. It is important that the expectations around input are accurately communicated. If your community is pushing, as part of its core values, openness and the encouragement of input and contribution, you need to ensure that people can actually do that. If they can't, you are going to have some angry villagers on your hands.

So before you respond to criticism, don't start from the assumption that it is unfounded. Take a good, hard look at your community and determine whether a problem really does exist. If it does, fix it. Reread this book and ensure you are implementing all the tips scattered throughout regarding open engagement. And invite the critics to help, if they are at all competent and conciliatory.

Bickering can often occur in communities driven by a commercial sponsor. Irrespective of how open your governance is and how well the paid staff of the sponsor engage the rest of the community, volunteers will always worry about having less input in the community than paid staff members. It's easy to see why: if a company puts your community at the center of its universe and throws some dollars at it, it is likely that they will have a vested interest in making their money work for them. This could *theoretically* affect the weight of participants' opinions about the community's direction.

NOTE

Folks, notice the nicely emphasized *theoretically* in the previous statement. The reason for this is that I was playing devil's advocate a little. Many communities have paid sponsors, and in my experience most sponsors respect the existing lines in the sand that the community has drawn and don't throw their commercial weight around. It is important not to automatically associate "paid sponsor" with "in charge of direction."

If your community has a paid sponsor, you need to go out of your way to ensure the community members still feel they are enabled to contribute. If your community is open and transparent and has facilities to engage with community development, ensure you celebrate those facilities and regularly remind your community that they will be treated no differently from those representing the commercial interest. If a community member is citing claims of unfairness due to this commercial sponsor, the only way you can achieve this is to build trust. To do this, I recommend that you get on the phone with them, get all the issues out on the table, clarify areas of confusion and miscommunication, and build a schedule of regular communication to tend to and provide feedback on the issues that were raised. You don't need to always provide solutions, but you do need to provide an ear to the ground and sensitivity to problems and issues.

If you are a community manager for a commercial sponsor, you may find yourself in the difficult spot of the sponsor demanding something that the community doesn't want. This is a complex and sensitive situation to manage, and you should tread carefully.

You should begin by determining exactly what the sponsor (who may well be your boss) wants and how it can achieve this with the least amount of disruption to community goodwill, governance, or processes. If the sponsor is determined to get this work done and is willing to step outside published governance, you should get an outline of the proposed work, submit it to the community, and ask the governing body (if it exists) to gather community feedback on the proposed plan and provide a summary of the community's feelings on it. If the sponsor chooses to ignore this opinion and go ahead, you need to make it very, very clear that it could significantly harm community relations.

Irrespective of whether there is a commercial sponsor, the trick here is to keep your ear to the ground when people feel that they don't have input into decisions and/or direction. When there is a sense of lost control, people will often bottle it up and discuss it only with a few other friends who they feel will sympathize with them. These people are not poisonous, and there is no malice behind their actions: they just don't know where to turn and feel that they are not being listened to. In such circumstances, the issue can quickly blow out of proportion. If you get a whiff that something is not quite right, investigate it immediately and reassure those concerned that things have not changed and the community is still as volunteer-focused as it has ever been.

To reassure your community, you should remind them of the community-facing resources that encourage people to submit their ideas and views. In the Ubuntu community, we put the following in place to encourage an environment of openness and collaboration:

Mailing lists

A range of public mailing lists are available, complete with publicly visible archives.

Brainstorm

This is a website in which people can submit ideas to be voted on. This offers a great list of items that are high-priority in the minds of the Ubuntu user community.

Public IRC channels

A range of publicly accessible IRC chat channels where Canonical staff members perform most of their work, and therefore have a virtual ear to the ground.

Sponsored developer participation

Canonical also sponsors many community developers to the twice-annual Ubuntu Developer Summit. This encourages strong community participation.

These resources and their open and participatory nature should speak legions in the interest of welcoming your community's input and ideas. Although you should certainly remind your members of these open facilities, you should also go one step further. If you have a serious complaint on your hands from someone who feels that her input is not welcomed, you should dig through these resources and find examples when the community has been involved in a productive change. Pointing to a public mailing list is one thing, but pointing to five productive conversations on that mailing list that illustrate your community's openness is much more valuable.

I have one final note about this particular cause of conflict. You should expect that some people will generate conflict simply because they offered input and their input was not acted upon. As an example, someone may recommend that your community takes a particular direction, but for various reasons that direction is not taken and the original provider of the input gets agitated and makes claims that he is not being listened to. This is nothing more than griping.

You will always need to deal with these kinds of accusations: you simply can't please all the people all of the time. In this scenario the solution is to again provide evidence that your community has engaged in input. But do be frank: tell the complainer that sometimes not everyone gets their way. Provide examples of how you personally have not gotten your way as a counter to him feeling isolated and singled out.

TIPS FOR AVOIDING CONFLICT

- **Ensure that there are very clear channels in which your community can engage.**
- **Ensure that feedback, opinions, and ideas *are* discussed, considered, and engaged with.**
- **If part of a commercial sponsor, always ensure that your staff members are publicly engaging and not hiding in private communication channels.**

Problems with Responsibility

Every community contains groups of people who have clear and perceived responsibilities. These folks are commonly seen as leaders who make decisions and the real movers and shakers who represent the wider community. These people with responsibility can include:

Governing members

Members of councils, boards, advisory groups, and more

Team leaders

People who run the many and varied teams your community may have in place

Community managers

People specifically given the responsibility to help advise and smooth how the community works

Sponsored leaders

Prominent members of the community who fund significant chunks of it and possibly pay staff to work with it

Each of these roles brings an expectation around engagement. Your community will expect a certain degree of professionalism, responsibility, and responsiveness. When a member feels that these attributes are compromised, up pops the big hairy eyeball that points at said leader. I don't like hairy eyeballs, and neither should you.

There are two primary causes of this conflict that happen far more in communities than they should, and both are closely interlinked.

The first is having too many eggs in one basket. This is when a single person has too much responsibility and control over a given part of the community. This can happen a lot with resource management. It is common to have one person as the single point of contact for administering mailing lists, accounts, forums, or other resources. This is all fine and dandy when that person can keep up with the maintenance requests, but when that falters, frustration follows closely after.

This ties into the second issue: responsiveness, a problem that has plagued every community. The reason is fairly obvious when you consider it: community leaders are typically highly motivated, fidgety people who have no problem keeping themselves occupied. Busy people tend to become really busy people, and it is not uncommon for them to take on a little too much. Too many responsibilities and not enough time are a recipe for further community frustration.

NOTE

Remember that you could be a bottleneck, too. If you find that your TODO list is spiraling out of control, it is likely you are a bottleneck. Check which items are blocking on you, and try to spread them out to more people.

If left untended, these issues can cause wider annoyance and even nastiness. This happened in a community in which I played a loose role a few years back. There was a single community member who maintained many of the resources the community used. This person always denied additional help and contributions and wanted to handle the resources himself. As time went on he became increasingly difficult to get a hold of. Maintenance requests were largely ignored. Frustration set in and thus emerged accusations of ego, self-obsession, the willingness to put personal interests ahead of the community, and other irksome claims.

While at a conference, I was chatting to a friend of mine about these issues and he gave me a piece of advice that has stayed with me. He said:

You know, I can understand the frustration people have with [BLANK], but the reason he has become unresponsive is not because of any malicious intent but purely because he just got busy. We are all guilty of this.

He wasn't wrong. Even though many were claiming this guy was fast becoming the son of the devil, such ramblings became overstated because (a) they were largely expressed in private cabals of people who all agreed, and (b) they had never confronted [BLANK] with the issues and never got his side of the story. Movie fans may consider this all very reminiscent of the legend of the unseen Keyser Söze in *The Usual Suspects*: the legend was somewhat outstripping the reality.

The obvious solution to this problem is to avoid these bottleneck issues in the first place, but despite your best efforts, you are still likely to experience some conflict as a result of these issues. Your primary role here is to investigate both sides of the story, provide feedback from everyone involved, and make necessary changes that are in the interests of the community where required. Don't allow members of your community to become Keyser Söze: break down the issue, understand all sides, and bring balance to the discussion.

TIPS FOR AVOIDING CONFLICT

- **Try to avoid bottlenecks set up by specific community members. If you experience them, unblock them as soon as possible.**
- **Dissuade against personal comments and remarks if bottlenecks appear.**
- **Always consider whether bottlenecks have a malicious undertone. If they don't, make that clear to detractors.**
- **Encourage public discussion of resource issues as opposed to private bickering sessions.**

Lack of Justice

The final topic we should cover as a common source of conflict is the feeling that previous conflict issues or problems were not handled in a manner that was expected.

Conflict resolution is seen by many as a means of those with influence delivering justice where it is genuinely required. When that justice is not exercised, previously happy bunnies can dramatically take on the guise of an extremely agitated and frustrated negativabunny. That's right, folks, I just said "negativabunny." Let's see how much Google Juice we can squeeze out of that word....

Claims of ineffective conflict resolution and injustice are concerns that call on you to commit a significant amount of your energy. While there are many causes for conflict, your community will depend on its leaders to unblock conflict and restore order. When this happens, all conflict feels temporary and manageable. If your community loses faith in its leader's ability to unblock and resolve these problems, it can feel as if the earth is shaking. It can make the community feel like a lawless state in which agitation and arguments reign supreme. This is obviously not a good position to be in.

One trait of a firm and clear governance structure is to allow assessments of the very people who assess and handle conflicts. As an example, if you have a forums community governed by a Forums Council, that council should deal with conflict and unrest. If that council fails to reach a desirable outcome, concerned community members can escalate the matter to the top-level Community Council that the Forums Council is subservient to.

The most important advice in avoiding this kind of criticism is always to engage in conflict resolution in a detailed and effective way. If this is a public conflict, you should regularly engage with the community to show you are helping to deal with the situation. If the conflict is private, those who claim that conflict was not resolved may simply be unaware that it was resolved. In many cases you can settle any concerns by dropping a quick email to the concerned members to let them know that the situation was handled and resolved.

TIPS FOR AVOIDING CONFLICT

- **Always deal with conflict. Never leave issues languishing or unresolved.**
- **In public conflict, ensure you are seen to be acting. Provide input, feedback, and reassurance in public communication channels. Justice must be done and it must be *seen* to be done.**
- **Always be responsive to concerns.**

PHYSICAL VIOLENCE: NEVER ACCEPTABLE

It goes without saying that violence is never acceptable in your community. While it is rare, you must be prepared to take a zero-tolerance view in case it does happen. If someone is violent to another member, make it clear that (a) they must leave the community, (b) you will inform the police, and (c) they will be able to rejoin the community after they have cooled down and demonstrated that they will not resort to such behavior.

Informing the authorities—which includes the police as well as the leaders of institutions you’re involved with, such as universities—is important for several reasons. First, violent people tend to escalate their behavior imperceptibly, and law enforcement can benefit from seeing patterns of escalating behavior so they can ward off real danger. Second, you need to protect your community against legal retaliation, because violent people often turn around and invent charges against their victims.

Another subtle point is that you consider very carefully whether you should mention that violence is never tolerated. For many, the merest mention of the topic is so obvious that it can be interpreted as patronizing. This really depends on your community. For example, it is undoubtedly going to be more acceptable to mention this topic in an urban regeneration community as opposed to a local neighborhood knitting group.

The Conflict Resolution Process

Now that we have covered many of the warning signs and indicators behind conflict, you will hopefully have an opportunity to step in there and deal with the issue before it really flares up. Even if you don’t manage to catch the warning signs and step in, you can apply a similar process to managing the actual conflict.

Of course, experts have proposed many, many different approaches to resolving conflict, and each situation is in itself unique. There is no cookie-cutter approach to dealing with conflict, and experience is the best judge of how to move forward to resolve issues.

What I am going to present here is a broad description of steps I generally use to deal with contentious issues. These steps map out the primary elements involved in conflict resolution, and you can build on them to form your own style.

Interestingly, while I was reading some of the many books about conflict resolution, I discovered a theory developed by Johnson & Johnson in 1994. I have been living and practicing this approach to conflict resolution for years without realizing it had a name. I was all set to call it *Bacon’s Great Conflict Resolution Theory* when I discovered that those pesky Johnsons got in there first.

Irrespective of who coined the theory, it is a practical, real, and directly usable approach for communities. Let’s take a look at Johnson & Johnson’s approach:

1. Collect data—learn what the conflict is about, and develop an objective picture of all parties’ perspectives.
2. Probe—ask open questions, listen, and engage with all parties to get the full story.
3. Save face—work toward a winning resolution for everyone, and try to avoid embarrassing either party while always remaining objective and unemotional.

4. Discover common interests—finding common interests and alliances will help find points of commonality beneath the conflict.
5. Reinforce—where both parties share a perspective and agree, reinforce those perspectives, and particularly try to use data to back it up.
6. Negotiate—start simple, trying to get both parties to agree on simple solutions, and then continue to build toward the common goals of both parties involved.
7. Solidify adjustments—Review, summarize, and confirm the areas in which both parties agree.

Each of the steps is performed one at a time. Naming and being conscious of these seven steps is useful to break down the process of conflict resolution. A little later we are going to walk through the resolution process and flesh out these steps, but before we do that, let's cover some general best-practice elements involved in the entire process.

The Role of a Facilitator

When conflict occurs, the person who steps in to straighten out the issue has a role like a judge or magistrate: to investigate the issue fairly and objectively and to reach a conclusion based upon that fair and objective judgment. This is the role of the *facilitator* (also known as a *mediator*).

A facilitator can't just be anyone: she must secure the trust and confidence of the warring parties. The parties involved need to have faith that the facilitator is going to take a fair, reasoned, and thorough approach to the conflict.

The probability that the conflict will be resolved is hugely dependent on this faith. With this in mind, let's spend some time looking at the profile of a great facilitator. This can give you some food for thought for your own role as a facilitator.

Be objective

Objectivity is the foundation of all effective conflict resolution. As we discussed earlier, you need to build faith in the participants. To do this, you need to build faith in the wider community that you can act in an entirely objective fashion. This raises a difficult question, though: how do you demonstrate objectivity in your community yet still offer opinions and direction? To remain objective, do you need to never express your own opinion?

Not at all. Objectivity does not spring from remaining aloof and above the fray, but in the way you engage in discussions and topics. This is all about how you interact with the community. There are a few simple best practices that can help communicate your objectivity to the community:

Be honest

There is no such thing as an objective but dishonest person. If your community doesn't trust you, they won't believe in your ability to be objective. As such, be honest at all times, privately and publicly.

Admit when you are wrong

Sometimes you are going to get it wrong...in public. Instead of denying any mistakes or withering behind a wall of silence, put your head above the pulpit and admit when you get it wrong. This is an extension of being honest, and your community will respect you for it.

Don't wear the flame suit

Every community has flame wars, that is, the arguments and disagreements that happen on mailing lists, websites, blogs, and elsewhere. Don't get involved. Participating in the flame wars causes you harm in two ways. First, it raises the temperature, because a well-respected community member has sought to weigh in, which drags the war out longer. Second, you don't want a reputation for taking part in online spats. Instead, you want a reputation for resolving them elegantly, and the place for elegance is not in a flame war.

Don't be afraid

Another key component in building honesty with your community is to be honest and up-front when you disagree with someone. Every so often, you are going to need to grab someone by the virtual collar and let him know what a destructive force he is. Under the premise that your conduct is fair and balanced and that the cause is reasonable, many community members will respect you for your frankness. Just be careful not to veer into outright bollocking.

Always remember that building confidence in your objectivity works only if you are genuinely...objective. Objectivity doesn't grow naturally in most people; you need to consciously consider it and work on it.

OBJECTIVITY AND ANNOYING PEOPLE

There are many ingredients that can risk or taint your objectivity. One of the most significant is trying to remain objective when dealing with someone you just don't like. Is it possible to be objective with someone who previously humiliated you personally on his blog?

It is, but it requires careful and conscious consideration to ensure you keep on the straight and narrow and don't let personal animosity blur your ability to judge a situation fairly.

The solution to this is always to remember that the community is bigger than you, that guy, or anyone else. Your ability to resolve conflict is something that you are seeking to achieve for the community, and you need to keep its best interests at heart. That could mean engaging respectfully with someone you don't like, even someone who was disrespectful to you.

My dad is a magistrate for a small county court in Northern England. Every morning when he drives to the courthouse, he recites his oath of office to himself: “I, John Richardson Bacon, swear by Almighty God that I will well and truly serve our Sovereign Lady Queen Elizabeth the Second in the office of Justice of the Peace and I will do right by all manner of people after the laws and usages of this realm without fear or favor, affection or ill will.” It keeps his mind focused and keeps him objective.

Be positive

Conflict is not fun for anyone involved. A conflict situation is a mishmash of different emotions from the different parties: anger, frustration, annoyance, self-reflection, embarrassment, and more. You should do your best to lighten the atmosphere.

Being positive breaks down into several stances. First, you should be positive about the ability to find a solution and an outcome. You need to give the participants of the conflict a positive impression that you can help, that you sympathize with their plight, and that you are going to help them to resolve the problems.

Second, you should be positive about the wider community and the values behind the project. As an example, whenever I have dealt with conflict situations in the Ubuntu community, I have started by reminding participants why we are all involved. This positivity not only reminds the participants of the important wider picture, but also highlights a connection between the members of the conflict.

Finally, there is huge value in just offering a generally positive and lighthearted approach to the situation. Having a generally positive demeanor, smiling, and using upbeat language and subtle humor are all great methods of lightening the slightly cloudy atmosphere. Of course, there is a balance to be struck here: don't turn your role as facilitator into that of a stand-up cabaret comedian, but a few subtle, amusing references here and there will ensure that everyone stays as positive as possible.

Be open

I am going to let you in on a little secret. The word “open” irritates me. Well, let me be clearer: the *overuse* of the word “open” irritates me, and in recent years the word seems to be more prevalent than a furball in an animal shelter.

When used within the context of conflict resolution, a valuable application of openness is to seek equality with all participants in the conflict and thus avoid cliques and in-jokes.

Some years back, I was observing a friend of mine dealing with a conflict situation who managed to violate this goal of equality and openness. He did this by engaging differently with one of the participants: he knew that person more personally and referred to various in-jokes and private references. Rather unsurprisingly, the other person (who was just as bemused

about the in-jokes as I was) felt he never had a shot at an objective judgment. Don't make the same mistakes yourself.

Be organized when you put your feet firmly into the shoes of a facilitator. Make sure to schedule calls and meetings at times you can adhere to; under no circumstances should you simply skip meetings or stop responding to email. Timeliness is key.

KEEP RECORDS

Always document what happens during conflict resolution. Get permission to preserve emails and other communications from all parties. Write down notes from sessions. In this way, you can refer back to previous communications when people dispute the facts. Keeping records is also important because conflicts sometimes escalate to higher levels of the community, and members will want a history of what happened. Your conduct may even come under examination.

While it is important to keep records, don't be tempted to quote people out of context and use their words against them. This serves no purpose other than to make people feel cornered. The records are instead a useful resource for keeping track of the discussion as a whole.

Be clear

The most fundamental task when beginning conflict resolution is to communicate to all parties involved the expectations around your role as facilitator. The primary expectations that I communicate are the following:

Solutions are the goal

I am here to find a solution. There may be open wounds and cuts and bruises from previous exchanges, but we all need to find agreement on how to move forward.

Evidence is central

The process is going to concentrate on evidence as opposed to emotion. The facts and reality of a situation are the guiding force, and emotion, carping, complaining, and assumed perception are not going to have a place in the discussion. This does not prevent an aggrieved party from asking for discipline toward someone who has violated the community's standards regarding respect, tolerance for others, and basic etiquette.

Conduct must be under control

All discussion must be polite and respectful. You will not accept disrespectful, threatening, or violent behavior.

*Compromise is the *modus operandi**

The goal here is to find a solution that satisfies the majority of considerations in the main parties, but this solution may not be 100% of what everyone—or even anyone—wants.

By framing the conversation with these expectations on all sides, you will help get the conflict resolution wheels on the runway and prepare for takeoff.

Resolving the Conflict

Earlier we talked about the seven-step Johnson & Johnson method for dealing with conflict resolution. Over the following pages, we are going to explore this entire process by breaking it into five parts:

1. Calm and reassure.
2. Get the facts.
3. Discuss.
4. Document.
5. Reflect and maintain.

These five parts will embody different steps of the Johnson & Johnson theory outlined earlier. In each part, we will discuss in detail the different considerations facing you. Each of these considerations is driven from countless conflict resolution incidents that I have been involved in over the years.

To make this content easier to understand, it is always good to apply the theory to a real scenario. As such, through the following pages I will be discussing a real incident that occurred between two very prominent members of a user group who were in fairly serious conflict over how to handle monetary donations to the group. Naturally I have changed the names to protect the innocent, but let me introduce you to the two characters:

“Lee”

Lee was loud, at times obnoxious and slightly egotistical, but despite these traits clearly a sensitive guy. His commitment to the group was unwavering, and he was always coming up with ideas and creative methods of growing the group and doing exciting things. With this in mind, Lee was keen to set up a facility to handle money (bank account, accounting, tax returns, etc.) and a governance body to handle this facility.

“Alan”

Alan was quieter than Lee, but never afraid to voice his opinion. Alan was the kind of guy who would bear grudges, but would not engage in confrontation. Alan took an immediate and visceral dislike to the idea of a money-handling facility. He was a firm believer in keeping the user group as simple as possible, and felt that Lee’s idea would create an unnecessary and complicated bureaucracy.

We will call this example “the fantastical user group debacle” and refer to it in each of the five parts. All set, friends? Let’s go....

Part 1: Calm and reassure

Before we begin the Johnson & Johnson items, the first step is to provide as much calm and reassurance as possible to the parties involved in the conflict. The goal here is to set the tone for the conversation so it begins without aggression and shouting. If you are receiving an aggressive tone, you should first have a conversation with the person involved and make it clear that you are there to help, but you can't do anything until he calms down. Use reassuring and familiar language. For example, talk about "calm," "resolving," "peace," and "community." You absolutely cannot move forward effectively if an aggressive tone is used, because it will instantly deter the opposite side of the argument.

It is this very first step where you need to firmly assert your position as facilitator. You need to reassure the factions and seal your commitment to finding a solution, demonstrating absolute objectivity, and focusing on evidence in the interest of finding a solution.

You need to strike a delicate tone here: sound reassuring and caring, but don't sound like a pushover. You need to ensure both participants are clear that you will be firm and fair and that you will hear all sides of the story.

The fantastical user group debacle. Alan and Lee clearly disagreed over how to handle money in the user group. Although this was the current topic, the initial email I had received suggested this was merely the straw that broke the camel's back. It was becoming evident that the two just didn't like each other very much and saw the world very differently.

Both had exchanged testy words with each other, primarily over email, but they shared even more fervent words about each other with me. The language was very emotional: talk of "hating" each other, "refusing" certain ideas, and "demanding" various assurances.

To get this off on the right foot, I always prefer to discuss things on the phone. Both Alan and Lee were happy to talk, and I called both separately to begin the discussion.

It was important that these were individual phone calls. At the beginning of the calls, I received the expected vitriol toward the other, and I sought to calm them down first. I then made it clear that I was here to help, to offer an objective investigation into what had happened to cause this rift and to offer my input. With so much anger on display, I made it very clear that I had my own requirement before starting: a polite and reasoned tone. I stressed that aggression and bickering had no place in the discussion. Both agreed, and we were off to a good start.

My goal now was to help identify if there was a way these two passionate community members could straighten out their differences or at least work cohesively in the same volunteer environment.

Part 2: Get the facts

J&J Elements Covered:

Collect data

Probe

The goal of this phase is to assemble as much evidence about the situation as you can. The real focus and priority here is to find unequivocal evidence, that is, evidence of the situation that can be independently verified. Here we want to separate out emotion and get to the heart of what really happened.

You should first speak to those on both sides of the conflict and ask them to provide you with their stories. To engage in this discussion, you should first decide how to communicate with them. I would highly recommend doing this on the phone or via Voice over IP (such as Skype) if possible. A phone conversation is far more interpersonal and allows both parties to communicate more quickly than over email or a chat medium such as IRC.

When you gather this initial story, you should expect a fairly significant amount of venting and emotion. Expect both parties to speak quickly, dart the focus around different issues, and keep remembering details and frustrations that they had previously forgotten to mention in the conversation. While you have this conversation, bear a few things in mind:

Listen

Take your time and listen to the person carefully. Give her a chance to speak, get her frustrations off her chest, and provide you with as much data as possible.

Ask questions

Start by asking the person to start at the beginning, walk her through what happened, and ask lots of questions about each step in the timeline. If you are uncertain about something, don't be afraid to ask. Remember, you want to gather as much data as possible.

Don't offer opinion

Try your best to not offer an opinion or emotional reaction to what the person tells you. Even if you strongly agree or disagree with her sentiment, you should try to just gather the facts as best you can.

Make notes

Always have a pen and paper (or text editor) ready before you have the conversation. Note down the timeline, the issues, and other elements of the story.

After this initial conversation, you should have a firm idea of all parties' primary problems and concerns. At the end of the conversation you should ask for one more thing: ask both parties independently to send you an email with as much evidence and details that illustrate their concerns as possible.

The next step is to reenact a childhood dream that has been left dormant for years: to assume the investigative role of Columbo. That's right, folks, it's time to don the long mac, chomp on the somewhat dog-eared cigar, and begin hunting for additional third-party evidence. Here you want to augment the content submitted from the (naturally biased) participants of the conflict and find some other data that can help you figure out where things stand.

With the focus very firmly focused on cold, hard evidence, you need to know where to look. Some of these may apply to your community:

Public communication channels

Look for public discussion on mailing lists, in logged IRC channels, social networking websites, and elsewhere.

Hooks

Earlier in the book when we explored how to measure community, we talked about the hooks that can be used to extract meaningful data about your community. Think about which hooks could be useful to gather relevant data. As an example, if the conflict is surrounding a specific person screwing up bug reports, look into the bug tracker and make a note of the activity that occurred there.

Content

Take a look at blogs, blog comments, articles, and other written material that may be relevant to the topic. There may even be evidence lurking in podcasts, online videos, and elsewhere.

When you have gathered as much evidence as you can find, you now have to consider whether you need third-party opinions. If the conflict concerns the conduct of people in the community, or it affects the wider community or other public issues, you may want to gather some feedback.

There are two approaches to soliciting feedback, both of which have pros and cons. First, you could target specific people for their opinions on the issue. The benefit of this is that you can specifically target people who you know to be objective and unbiased on the issue. The downside is that you get a limited spread of knowledge that may not accurately reflect the situation.

The second approach is to have a call for more data from the general community. The pros of this approach are that it is open and inclusive to the wider community, but it does have problems: (a) it raises the profile of the problem, which is typically the opposite of what you want, and (b) it can bring out all manner of crazies.

I would personally recommend you take the former approach but ensure you have a wide spread of opinion and still maintain a strong sense of objectivity around the information that you receive.

ALWAYS MAINTAIN PRIVACY

When soliciting opinions about a conflict issue, you should do your best to obscure the identities of those from whom you solicit private opinions.

When you have solicited your evidence, input, and general feedback on the conflict, you will have enough material to begin forming a position on what has happened. Although you have to remember the possibility of bias in each piece of evidence and opinion, it is likely that you are beginning to get a broadly accurate idea of what has happened and what went wrong.

You now need to reconcile this evidence and input with the goals, values, and perspectives of the community. Have any of the actions of those involved fallen outside the culture of your community? Is there any evidence of personal benefit being put forward as the best interests of the community?

Now is the time to review the situation and adopt your unbiased perspective to draw a conclusion on what has happened. With almost every conflict situation, the fault will lie on both parties in different areas. You should note down how each party could have handled the situation better and if and when they acted outside the reasonable (and preferably documented) expectations of the community.

The fantastical user group debacle. Earlier I mentioned that I had initial phone calls with Alan and Lee to calm their frustration. I used my phone calls partly to gather some initial feedback on what the primary issues were (the donation system being one of them) and their respective viewpoints.

Alan gave me a patchwork of his problems with Lee and Lee's perspective, a stream of consciousness that came flooding toward me in one big disorganized mess of thoughts. I noted down everything he said, and regularly repeated key problems that he outlined to ensure we were both clear on what he was saying.

I then did the same for Lee. This time, an even more disorganized set of thoughts came flying in my general direction. Lee was less personally angry at Alan, but he was clearly frustrated with the situation and was growing impatient.

Because the environment that encased these issues was a user group, I started doing some third-party research. I looked into their mailing list, lingered on their IRC channel, and had a few private one-on-one conversations with people whom I knew to be objective in that community. As I received more evidence, a pattern was forming: Lee's requests were increasingly erratic, demanding, and personally driven. It seemed that Lee not only wanted a governed body to cover how money was handled, but he wanted almost absolute control himself. Reports of Lee's demanding nature and self-perception of leadership was concerning other group members. Although it was clear he was not seeking to make money from the group, there was maybe a little too much desire for power in his plan. My hunch was that Lee's desire for power was an insecurity caused by the rift between him and Alan, another very prominent community member.

At this point in the process, I was faced with two challenges. First, there were the specific relationship problems between Lee and Alan, but there was also the wider concern in the community surrounding Lee's conduct and leadership efforts. My feeling was that if I could reduce the venom between Alan and Lee, this would (a) bring less of their personal anger to general community meetings, (b) drive toward a solution on the money handling issue, and (c) inspire Lee to become more reasonable and less power-hungry.

Part 3: Discuss

J&J Elements Covered:

- Save face
- Discover common interests
- Reinforce
- Negotiate

The next step is to engage in a conversation with both parties that nudges them toward a conclusion. The goals of this part are to discuss the issues, hear all sides together, and then to find solutions and consensus in areas where both sides agree. This is all about searching through the chaotic claims and memories for patterns where you can lay down an eventual consensus. By finding these patterns, you can make progress toward a general agreement and also build a more positive atmosphere around shared values as opposed to differing ones.

The first step is to schedule the discussion. If the conflict is between two specific people, the best medium for discussion is typically a conference call. This can happen on a range of online telephony services (such as Skype) or by using a conventional telephone conference call service. Many conventional handsets even support three-way conversations at no extra cost.

If the conflict is public and part of a team or group, schedule a public meeting. I have found the most suitable medium for this to be IRC. It allows people to share thoughts quickly so long as they can all get online at a specific time.

Your choice of medium is heavily dependent on what is comfortable for your community. The most important factor is that it is real-time: the discussion needs the immediate give and take of a meeting of minds. Email and forums are fine for discussing general issues of the conflict, but the resolution really needs to happen in real-time.

The most important points about scheduling a public meeting are that it should (a) be in a neutral or, if suitable, public place, and (b) it should be at a time that is convenient for the primary stakeholders in the conflict. What you want to avoid is scheduling a meeting when one of the primary participants has to get out of bed at 3 a.m. to take part in the discussion. That person will invariably be a little grumpier than normal (shock, I know!), and said grumpiness will infect the discussion and complicate matters.

NOTE

Face-to-face meetings for conflict resolution are sometimes possible, and while valuable in many instances, I would generally recommend against them. Body language plays a huge role in face-to-face conflict resolution and can distract the participants from the issues and provide a more intimidating environment in which participants may struggle to find consensus.

Whether public or private, you should keep the length of the meeting to a set time slot (one hour is usually advisable). There are a few reasons for this.

First, if you have an open-ended meeting, it will first go on for a *long* time and some of the participants will grow tired of the meeting before others, causing frustration. Second, a set time slot will keep everyone focused on the details and mitigate inane rambling, diatribes, and monologues, which are always common in these kinds of discussions. Finally, you need to think of your own sanity, too: dealing with a juicy three-hour chunk of conflict resolution can drive you potty. If you break down the discussion into manageable chunks, it will mean that the members of the discussion are always fresh and so are you.

With the meeting scheduled and everyone either on the line or in the channel, you can now begin the discussion. Although there is no fixed formula for handling these contentious topics, we will now discuss a broad template that can get you started.

First, introduce the conversation. Say who you are and the role you are playing; state the purpose of your meeting; and reiterate the values, goals, and bigger picture of the community. Make it clear that you are here to help and will be taking an objective role in discussing all the issues involved.

Next, make clear the ground rules for the discussion. Make it clear that the discussion needs to remain polite and honest and that everyone should be driving toward a conclusion. Also make it clear that while everyone involved cannot fix the problems of the past, you can work together to produce a better future that avoids these issues. Reinforce that to achieve this better future it will require the commitment of everyone in the meeting to move toward a solution. Ask everyone to enter the discussion with an open mind to finding this solution.

Now it is time to get to the meat of the entire conflict resolution process. Friends, this is where the action really happens. Discuss the different issues openly and objectively, seek additional input, and focus on how you can make small agreements here and there. Remember that the goal here is to achieve lots of little victories.

Try to reframe the conversation away from what the participants disagree on and instead focus on what they agree on. Cover each issue one at a time and slowly build more and more little agreements. When you reach a consensus on a specific topic, make it clear that you are making progress (but don't be tacky when celebrating this progress).

Throughout the meeting, take plenty of notes and be careful to note down areas in which you reach consensus. These notes should preferably be public so there can be agreement on the wording: the devil is always in the details. Make sure everyone understands the consensus, as sometimes subtle differences in interpretation can blow up later in the discussion. As such, when agreement is made, repeat it and ensure you get a clear agreement from everyone. The devil is in the details here, too: when you repeat what you understand as the agreed-upon view, one of the members of the discussion is likely to disagree with a certain choice of words or require clarification. As such, adjust your language until everyone is happy.

When you reach the end of your time slot, you should first thank everyone involved for their contributions, express pride that you all made progress, and repeat the areas in which you reached consensus. If you didn't reach consensus on anything, indicate that you still made progress and that you are confident that more progress will be made in the next meeting. You should never end a meeting without scheduling the next one: this will ensure that the issue has a sense of continuity.

The fantastical user group debacle. To resolve the conflict between Alan and Lee, I scheduled a conference call to bring them together and discuss their different opinions. Fortunately, both were based in the same time zone, so I scheduled a call that was mutually convenient for them both. Before the call began, I refreshed my memory of the evidence using the notes that I had gathered. My goal in the call was to find as much consensus as possible, particularly around the money handling issue. While it was the group's decision as a whole about how they handle donations, I was keen to reduce the vitriol from the Lee and Alan relationship because this would then help the discussion with the rest of the group to be less emotional.

When the call began, I largely followed the steps I outlined in the previous few pages. I introduced myself and my involvement, made it clear that reasoned and polite discussion was required in the call, and started covering each issue one by one. I started with general attitudes toward funding resources in the group. The first piece of consensus we achieved was that the group did want to conduct activities that required equipment and therefore an outlay of money. This was a step forward: both Alan and Lee agreed that the need to handle money in some way was required.

The next part of the discussion was to cover typical activities requiring money. Was this a small amount of money for printing and CD duplication, or was it a larger expense for equipment and travel? Consensus struck again with agreement that costs would be for small to medium expenses, the most expensive being a few nights in a hotel and a train ticket.

At this point in the discussion, there was a general atmosphere of agreement in the call. Both Alan and Lee were loosening up and more pleasant with each other, and the growing sense of agreement was having an impact. As the call progressed, we edged further and further toward agreement on a series of different issues. After a few more small steps forward, the time was up. I reiterated our progress and we all agreed to have a wider meeting on the money handling topic with the wider community.

A week later, the community meeting kicked off, and I shared some of the agreements that Alan and Lee had formed. We then moved forward to discuss how donations would be handled, and the community agreed on PayPal. The next step was to discuss the crux of the issue: who would look after the money. Alan and Lee each expressed their perspectives in turn and other community members weighed in. A number of community members expressed concern over a full governance solution to handling money, and naturally Alan reiterated his views on the matter while Lee and some others provided their opposing views.

To find a middle ground, I proposed that instead of a full governing body to look after the money, Lee (who had been the primary champion of a money management facility in the group) would handle the funds but provide open accounting of the funds. I also recommended regular meetings with the group about how the money was being handled. Furthermore, I offered Lee the task of investigating what we would need to do to satisfy the local tax requirements.

The group unilaterally agreed. Bingo! This tentative and fragile agreement was the victory that we could build on to later reach a firmer sense of agreement and consensus. The next step was to document the extent of this agreement so everyone would be (literally) on the same page.

Part 4: Document

J&J Elements Covered:

Solidify adjustments

As you proceed through your negotiations in the previous part, you should document each agreement in detail and ensure that both parties agree to what you have documented. It is this agreed document that is going to form the basis of cooperation between the two parties in the future.

With this document, once again, the devil is in the details. Ensure that you use precise, descriptive language and try not to use conflated or ambiguous words. This document should be unsexy but accurate: its purpose is not to enthuse or inspire, but to accurately describe the agreement in all of its boringly descriptive and unexciting glory.

As you build up this document, check in formally with each party to gain agreement on each addition. By the end of the discussions, when you have consensus on all major areas, the document should be no surprise to anyone.

The fantastical user group debacle. In the case of our friends Alan and Lee, I spent a total of three calls with them fleshing out a conclusion to the conflict. In each call we made progress, and with each agreement I noted the precise results in a document we all shared. After each meeting, I emailed them with a summary of what we agreed and the total set of agreements. The act of documenting our progress and sharing it fostered a real sense of progress.

Part 5: Reflect and maintain

The final part of the conflict resolution process is twofold: to maintain progress on the agreed-upon outcomes and to encourage a general sense of personal development in the members of the conflict.

The former goal is an exercise in keeping your wheels on the road. Just because you have a set of written agreed-upon outcomes doesn't mean that anyone is going to stick with them. Don't hold the members' hands as they execute the outcomes, but check in every so often and ensure everything is running smoothly. This can often be as simple as a quick email to each party to see how things are going. To ensure you don't forget, it's often useful to put it in your calendar for a few weeks' time.

The second element is subtle yet important. In many cases of conflict, the participants often become very reflective at a later date. Many will review their actions and their conduct, and while they will remain resolute in some of their opinions, they may also express regret at how they handled certain situations.

These reflective lessons are valuable, and your community needs reinforcement when they happen. To see why, think back to when you have made your own reflections on your life. When someone has been by your side to cheer you on, it gives you much more determination to be consistent in the change. Because you are the person who helped get these two parties through the conflict, they will be looking to you for guidance and validation.

Offering this validation is something worth doing, but very carefully. There is a fine line between validating and patronizing. Help to validate them, but listen carefully to the feelings they express, and don't rush. Get it right and not only have you unblocked the conflict but you have helped to improve someone's life inside and outside of your community.

The fantastical user group debacle. After finishing the conflict negotiation with Alan and Lee and the wider group, I stayed in regular touch with them both and the team, and I still am today. I have had many an informal conversation with them on IRC, the phone, and blowing the froth off a few cold ones, and have helped to reinforce their growth. While I've helped them, they have helped me so much, too. The reason I picked their story for this example is no coincidence: it is their conflict that helped me to really understand the topic better—but more importantly, to understand them better. At its heart, conflict is about understanding people and helping them to understand each other, and sooner or later you will have your own Alan and Lee story to take lessons from.

Dealing with Burnout

So far in this chapter we have explored the importance of conflict, the different elements of the conflict resolution process, and how to step through it. The majority of this content can be applied when people fall out, get frustrated, get out of bed on the wrong side, and otherwise

end up in the ditch. Unfortunately, there is one other element in life that very specifically falls into the “not fun” category: *burnout*.

NOTE

Although I have always dreamed that when someone shouts, “Is there a doctor on the plane?” I can step up to the plate, unfortunately I am not a doctor, and you should remember this through this entire section. My carefully scrawled and spellchecked words are no replacement for the opinion of a medical professional. If you are worried about burnout, go and see the doc and get some advice.

Burnout is a problem that affects all walks of life, all people, and all professions. As such, it is a problem that affects all communities, and yours is no different. Burnout refers to long-term exhaustion that typically causes lack of interest and focus. Unfortunately it can be devilishly difficult to spot and prevent in your community.

Burnout appears as a series of often subtle changes in personality, perspective, values, and behavior in the sufferer. As these changes progress, it can be difficult to identify that members are suffering from burnout. Unfortunately, burnout often is instead misdiagnosed as irrationality, short temperament, unusual and strange behavior, lack of tolerance, or for the ladies out there, accusations of “the wrong time of the month.”

While it is difficult to identify categorically, fortunately there is some compelling research that was first published in the June/July 2006 issue of *Scientific American* in an article called “Burned Out.” It presented the findings of two psychologists, Herbert Freudenberger and Gail North, and their *Burnout Cycle*. The cycle is comprised of 12 phases that outline the progressively serious steps that are part of burnout.

These steps don’t necessarily happen in a sequential order (it can vary from person to person), and some sufferers will skip some of the steps whereas some will dwell longer on them. These steps offer an interesting list of warning signs for potential burnout victims. Let’s take a look at them:

1. A compulsion to prove oneself

Often burnout is triggered by an obsessive commitment to prove yourself. This desire is founded in demonstrating to your colleagues and particularly yourself that you can knock the ball out of the park.

2. Working harder

To knock the aforementioned ball out of the aforementioned park, hard work is needed. This is manifested in long days, longer nights, and an inability to switch off results.

3. Neglecting one’s own needs

In this stage, simple pleasures such as sleeping, eating, socializing with friends, and watching *Seinfeld* are seen as just that: pleasures, and as such a distraction from work.

(I am not sure if there is a proven connection between a lack of *Seinfeld* and burnout, but there should be...).

4. Displacement of conflicts

In this stage, you don't really understand the problems that you have. If they lead to discomfort or even panic, the victim dismisses the impressions because they feel threatening.

5. Revision of values

In this phase, the obsession and focus of work means that traditional values such as friends or hobbies are dismissed, rejected, and pushed aside. Here your only evaluation of success is being good at your job.

6. Denial of emerging problems

In this phase, cynicism, intolerance, and aggression raise their ugly heads. Colleagues are dismissed as idiots. Your increasing problems are blamed on lack of time, incompetent coworkers, and unfair workloads.

7. Withdrawal

You reduce your social interaction and contacts to a minimum and dial up your work to 11. You may start relieving the stress by boozing more often during the week or possibly even resorting to drugs. Whatever your choice of substance, you appear to be indulging in it a little more than usual—and dangerously so.

8. Obvious behavioral changes

Your strange and erratic behavior is obvious to your friends, family, and colleagues. You are not yourself, and your nearest and dearest can see it a mile off.

9. Depersonalization

At this point you feel like you offer no value to the world, and lack confidence in what you feel you could once do. Your life feels like one long series of mechanical and emotionless functions.

10. Inner emptiness

You feel an expressed sense of emptiness. You resort more to booze or drugs or possibly find relief in overeating, strange and exaggerated sexual behavior, or other activities.

11. Depression

Here you feel hopeless, lost, and exhausted, and see little in the way of rays of light for the future.

12. Burnout syndrome

At this, the most serious level, you feel suicidal and desperate for a way out. You are on the verge of mental and physical collapse and need medical support and attention.

Wow, by the end of reading that lot you may want to go and pet a small animal, watch *The Sound of Music*, or sniff a rose. It is pretty frightening stuff, and unfortunately it appears to be prevalent.

Some of you will have read the list and identified with many of these steps, whereas some of you will be identifying others who may have exhibited some of the steps. I have met and known people who have exhibited almost all the behaviors described in these steps, and when serious burnout takes its grip, it can destroy families, careers, and many other aspects of life.

Detecting and Treating Burnout

With the risks evident in the list of symptoms, you are sure to be wondering what is the best approach to manage this risk. Is there a way to identify and react to burnout in your community?

This is something I have participated in during various discussion sessions at different conferences. Unfortunately, there is no recipe or secret formula for dealing with burnout in a community. The best solution is to subscribe to one simple philosophy that has helped people deal with complex life changes and decisions for years:

I got your back, dude.

Although it may seem outrageously simple, the easiest and most applicable method is to first develop a nose for symptoms and to then extend a personal hand of friendship to the sufferer. Having that sense of companionship through a tough time can really help with burnout.

To detect the symptoms you should first read, reread, and then read again the 12 items in the Burnout Cycle. These items provide a core set of knowledge for understanding the nature of burnout. You should then keep a general eye out for these symptoms in your community. Specifically look for and be conscious of changes in behavior. If someone just “doesn’t seem herself,” she may be getting bitten by burnout. It is these changes in behavior that are the typical signs.

If you have a suspicion that someone is getting burned out, just strike up a personal conversation and be entirely frank. Tell the person you noticed she has been a little different recently and that you are concerned. Ask her if she is OK, and ask if there is anything you can help with. In many cases the person will tell you what is on her mind, what is stressing her out, and any problems she appears to be having.

With overwork as a common cause of burnout, you should also ask how she is coping with her workload and if there is anything you can do to ease it. This offer of help in itself can be a stress reliever—it is a validation that someone is there to help her get through her TODO list.

Required rest and relaxation

One of the most effective methods of shackling up burnout is to get away from things and unwind. It is amazing how a small vacation can help someone decompress.

This happened to me when I felt I was burning out. I felt like I wasn't myself and could feel how stressed and anxious I was. To deal with this, I went to Ireland for a long weekend to visit a friend. It is incredible how those few days with a friendly face, getting out in the countryside, having a few drinks, and getting away from a computer helped.

If you suspect you or someone else is burning out, tell him to do the same and get away for a few days. He will almost certainly claim he can't or doesn't need to, but stand firm: it is for his own good, and he will thank you for it.

VOLUNTEERISM ESCAPES NOTHING

When on the subject of communities and stress, looks can be deceiving. Although most communities are firmly wedged in the volunteer category, that doesn't mean that their participants don't develop, feel, and react to stress. The lack of compulsion behind volunteers' involvement and contribution does not mean that volunteers who feel stress can just go and do something else. People grow attached to communities, their ethos, and their sense of family. The involvement may not be contractually required, but it is often emotionally required inside the mind of the contributor.

Work/Life Balance

At the center of the somewhat unpleasant universe that is burnout is the problem of balance. Although there is little concrete scientific evidence to determine who burnout is more likely to pick on, mere observational evidence suggests that technical folks, musicians, counselors, authors, and teachers have a higher than normal risk of reserving a place on the dreaded Burnout Cycle.

Balance is a surprisingly complicated goal for many to achieve, particularly if your community is an online, Internet-based community. Years ago it was easier to get balance: you simply switched your computer off and went and lived the parts of your life that didn't involve a mouse and a keyboard. As the Internet has steamed into our lives more and more, the amount of time in our lives that doesn't involve said mouse and keyboard is being reduced.

In addition to the familiar tools of the workplace, such as email, office suites, web browsers, and accounting packages, we now have social networking websites such as Facebook and MySpace; blogging sites such as Blogger and WordPress.com; microblogging with Twitter and identi.ca; and online chat services such as Skype, Google Gchat, MSN, Yahoo! IM, and AIM. Let's also not forget the entertainment on the Web: countless websites, animations, videos,

and articles are all there to attract us to the computer. We can then seal the deal with the countless other online facilities such as Internet banking, reviews websites, mapping tools, online shopping, games, and more.

It is easy to see how this merry band of pixelated distractions can take Ctrl, and it is not entirely unsurprising that someone could spend an entire day and most of an evening in front of a computer. This is itself not exactly healthy: computers are great, but everyone should spend some time away from them to decompress, get some fresh air, and energize other attributes of the human condition, such as getting out, playing sports, spending time with friends, romantic embraces, and other fun things that don't involve staring intently at a screen.

Addiction

The problem is that when the rest of your life is wrapped with window borders, you are only ever a click away from either work or other commitments, such as community. While we want to encourage our community members to throw themselves into our goals and enjoy every moment of it, it is important to ensure that in the process of doing so they don't ignore and neglect other parts of their lives.

Addiction has affected many online communities: there are contributors and members who spend every conceivable moment of their lives embedded in the community. This can be seen everywhere. I know of many people today who appear to be constantly online at all times of the day, always responsive to chat messages and queries and seemingly never away from their screens.

For many this is an agreeable choice that they can step away from when needed. Many people can wake up at 7 a.m., work all day, spend the entire evening in front of the computer in pursuits of their own, head to bed at 1 a.m. or 2 a.m., and spend a valuable six hours sleeping, only to wake up and repeat. That may be OK because these people can easily go away for a weekend, spend a few evenings doing something else, and go on vacation without getting jittery. For some, though, even spending one evening—let alone a whole weekend!—away from their familiar screen can seem like too much. In these cases we are seeing strong signs of *addiction*.

You should be very cautious of addiction: it is never healthy in anyone. Unfortunately, the nature of the addicted beast typically means these people are in a state of denial about their condition. Just as with alcohol, cigarettes, or gambling, claims of "I could stop if I wanted to" are often thrown in the general direction of naysayers, but their claim is rarely, if ever, tested.

The reason for your caution is that at some point an addicted member *will* burn out. It may take longer than expected, but when it does, it could have catastrophic results. Keep an eye on your community members and how much they are online: if it feels too much, a quick and sensitive word in their ear can help them get away for a few days.

Summary

Throughout this chapter, we have explored some of the darker alleys of community leadership. While 90% of the time working with community is unbridled pleasure and joy, it is this 10% of life that can be the most difficult and sensitive to deal with. It is also this 10% that can bring the most stress to your table and is the most likely cause of uncertainty in how to move forward.

The most important thing to remember through all of this is that conflict and its associated resolution is not new. It is as old as the world itself, and countless generations of people just like you have learned how to deal with the contentious issues before you. While the topics and advice covered in this chapter will get you off to a great start, you should do your best to find a mentor. Find someone who has been around these parts before, and ask her if she will be happy to advise you from time to time about where to go when faced with a fork in the road. Accept that the first few times you mediate, you'll make mistakes—and don't feel bad if the outcome is less successful than you had hoped.

Great community resolution is based upon experience. The advice throughout this chapter, combined with the advice from a mentor and augmented with your own growing body of knowledge, will combine to help you handle these testing situations elegantly, and before long you will become the mentor to others.

Good luck!

Creating and Running Events

“The meeting of two personalities is like the contact of two chemical substances; if there is any reaction, both are transformed.”

—*Carl Gustav Jung*

ON JULY 3, 2001, MY FRIEND LEE JORDAN AND I BOARDED THE TRAIN from Wolverhampton to London. It was early, we had barely slept the previous night, and we should have been grumpy. We were not, though. We were on our way to the Linux Expo at the Olympia Exhibition Hall to run an exhibition stand for the KDE project, and we were jazzed up.

We’d spent two weeks building up to the event, gathering t-shirts, posters, fliers, demonstration machines, leaflets, customized name badges, and more. I was determined to make sure that anyone and everyone who walked past our 6’ × 4’ booth would leave with a memory. Of course, I hoped that this memory would be of the incredible technology behind the KDE desktop as opposed to the tired-looking, long-haired 22-year old with a guitar pick on a chain around his neck.

We got far more than we bargained for. The booth was well received and was in itself a learning experience. We got to understand expectations around KDE far better, got to show off and educate people about the project, and met a number of new contacts. In addition to this, we met many community members from elsewhere in the open source community. We also got to meet a range of open source companies and publishers (in fact, it was at that show where I

scored my first writing gig). Finally, while at a social event at a nearby pub, I found myself in between Alan Cox and Jon “maddog” Hall, two legends in our community.

Afterward, tired and weary, Lee and I dragged ourselves back to Euston station, boarded our train, and for the first time in three days, sat down and breathed.

My body was utterly exhausted, but my mind was exhilarated. I had just experienced a roller coaster of thrills. I had traveled to a place I was unfamiliar with. I represented a community I was proud of, met people I had only chatted with online, shook the hands of two of my open source heroes, and even managed to score the chance to get an article printed in a magazine. In the same way my very first encounter with community way back in [Chapter 1](#) inspired me to explore it further, this, my first event, secured my desire to make my passion for community my career.

Building Family Values

Way back in the first chapter, I waxed lyrical about *belonging*. This is the magical substance that builds strong, enjoyable, and motivated communities. Communities that instill a sense of belonging in their members are by definition mature and capable. Those who belong feel productive and empowered, and they form the firm foundation of your community.

Although many of the workflow approaches that we have covered in this book will build a sleek, efficient, and effective community, the impact of these approaches on belonging is primarily by association. When you have smooth processes and infrastructure in your community, it makes the community feel effortless, and it makes your community members feel productive. When they are productive and enjoy the fruits of their labor, a sense of belonging begins to develop.

Great communities are built on great relationships. When people *really* feel a sense of belonging they are enjoying not only being productive but swimming in the tide of your community’s personality. When you put productive people together in a room (real or virtual) and they feel a sense of *family*, your community will be inundated with belonging.

Family is the operative word here. Many of us traditionally feel it from our experiences with our parents and siblings, but we feel it elsewhere too. Many will feel a sense of family in their local bars and restaurants, at their school or college, in coffee shops, and elsewhere. The sense of family exists when you not only feel a sense of belonging, but you also share a very deep and personal connection with other people.

The difference between *belonging* and *family* is subtle yet important. It is entirely possible that people can feel a sense of belonging and yet have little in the way of a personal engagement with other community members. I have seen countless examples of members who enjoy an administrative role in a community, be it system administration, managing resources, or otherwise. These members often enjoy the nuts and the bolts of helping the community but rarely socialize or engage with their fellow members. They feel a sense of belonging because

they know their contributions are respected by the wider community and that their work is important for the community to prosper.

As community leaders we should aim for belonging but strive for family. A key technique for achieving this is *events*.

Events

Most mornings my alarm goes off at 7:30 a.m. I wake up, grumble at the alarm, and then drag my sorry arse downstairs where I bow at the altar of the coffeemaker. Still waking up, I struggle to comprehend the coffee-making steps. I bumble through the process, spilling water everywhere, getting coffee grounds stuck to my bathrobe, and generally hating the world until I have finished chugging my thermos of wake-up juice that I clutch onto for dear life as I read my email.

My day then consists of the same approximate set of ingredients. I spend the morning on the phone with my team, colleagues, community members, and journalists. I then attack my inbox with a piercing determination, grab a sandwich for lunch, and then spend the afternoon working on my objectives and initiatives. This, my friends, is my routine.

Your community has a routine, too. Every day your community members will engage in their own set of stock interactions with your community. This may be sending email, having conversations, producing things, or other activities. Whatever these actions may be, they will happen each and every day—neither different nor unexpected. So far in *The Art of Community*, the topics that we have already discussed have been the fodder that helps make this routine as enjoyable and productive as possible.

This is still, however, routine. It is healthy to break routine. For many, this means a vacation, friends and family visiting town, and other time away from the community. If you want to break routine yet still have your community contributing, an excellent solution is to organize and run *events*.

Events are special, focused times in which a group of people do the same thing. This could be a large gathering such as a conference or a small online meeting. Irrespective of what the event is, every event gathers a group of people together at a set time. This gathering of people can be hugely motivating for a community.

As we will discuss in more detail a little later, events don't have to be physical face-to-face meetings. They don't have to be large and formalized, and they don't have to be complex and expensive to organize. Events can be small, informal, and based online.

Whatever kind of event you organize, there is a range of benefits available to your community:

Building family

Bringing people together, particularly in social settings, helps to build the sense of family. You want to have your community not only get on well in a productive sense, but also get on well as friends. We will discuss this more later.

Breaking the routine

All events break the routine. As an example, if you are an online, software-oriented community, a physical event such as a conference or summit will get you out of the house or office and meeting people and sharing conversations. In the same manner, an online meeting will also break the routine: it will bring people together for a shared discussion, which is often fun and exciting to be a part of.

Focusing the mind

Events provide an excellent opportunity to focus your members on given projects or targets. Many events focus around a particular date, release, or project. An example of this is the Ubuntu Developer Summit, which focuses on the next release of Ubuntu.

Identifying of leaders

Organizing an event almost always inspires the leaders and strongest personalities in your community to bubble up to the surface. This is an excellent opportunity to note these leaders as you grow your community. Events also allow new people to shine: those who might not be programmers or master quilters—or whatever your community focuses on—can exercise other talents when organizing the event.

Taking the pulse

Events are a useful opportunity to get some insight into the goings-on of your community. This insight will most typically be shared in social events, such as casual meetings in the bar during the evenings of a conference.

These benefits all help not only to keep your community productive, but also to excite, focus, and energize them. Events are an excellent opportunity to really fire up your community, produce social bonds, and lay the seeds for long and rewarding contributions to your community. Events are not merely nice, optional variations of the norm; they should be a regular part of your community growth.

Although all events used to fall into the same broad category of face-to-face meetings, the explosion of the Internet and digital connectivity has opened up the door to a new breed of event that augments these traditional localized meetings.

As such, events can be divided into two broad categories: *physical* and *online* events. We will examine both of these areas in detail throughout this chapter, but let's first have a look at how to organize events in general.

Getting Organized

Events, particularly physical events, are big barrels of details. Everyone I have ever spoken to regarding event organization has always learned the hard way that being organized is the only way to ensure that all of your plans land on the ground running. Getting a detail wrong can undermine the entire event. For example, you may have a perfectly laid out conference but forget to get signs made with directions to the different rooms, so people struggle to find the location of the talks they are interested in.

Step 1: Identify Requirements

You need to choose which kind of event you would like to organize and then identify which steps are involved. We will defer discussion of these steps until a little later when we delve into the different needs for various physical and online events.

When you have an idea of these needs, write them down and flesh out an action plan for carrying out that specific component. As an example, if one of those items is accommodation, you should note down each of the different steps involved in sourcing accommodation.

Step 2: Find Help

The next step is to find people to help you organize the event. This is particularly important for physical events. For smaller events, such as small sprints, you may not require much help, but for larger events, such as conferences and summits, you will definitely need help.

For physical events it helps to find people in your community who are happy to look after specific roles. Here are some of the role divisions that may apply to the events you run (not all will apply to every event):

Accommodation

This person organizes the hotel options and reservations.

Sponsorship

If you are seeking sponsorship for the event, you should have a single point of contact to deal with the sponsors.

Merchandise

If you have t-shirts, lanyards, and other items that need to be arranged, this person organizes these.

Speaker liaison

This person will work to find and schedule speakers. This person also manages the scheduling of speakers.

Special events and catering

This person will arrange any catering requirements as well as social events, parties, and additional gatherings.

When you have these roles identified and filled, you should organize regular meetings where organizers update each other on their progress in organizing the event and discuss further work and any problems.

I would recommend that you hold these meetings every two weeks at first, and in the two months leading up to the event, accelerate the meetings to be held weekly. I would also recommend that you organize conference calls if possible to host the meetings, although an online chat medium will also work.

NOTE

Here I have listed only the roles that apply to physical events. Physical events are always a lot more complicated to organize than their online counterparts: there are a lot more components to figure out, and many of these components rely on outside parties, such as hotels, catering companies, equipment providers, and more. Although you can divide online event organization into roles, too, online events typically require far less work and can be organized by a single person.

If you find that your event is too much to handle for a single person, though, do break it into roles in a similar way.

Step 3: Set Deadlines

This is what I consider the most valuable step of all in organizing your event. You should look at the range of components in your event planning and produce a set of deadlines that provide plenty of time to achieve those tasks.

Imagine you are organizing a conference. For each of the major tasks listed in the previous section, you should prepare a set of deadlines. As an example, these could be some deadlines for the process of opening up a call for papers, choosing proposals, and announcing the final decided list of speakers:

- February 23—Begin planning Call for Papers and put together website form for submissions.
- March 9—Open up the event’s Call for Papers and publicize.
- April 13—Announce “one week left!” for the Call for Papers.
- April 20—Call for Papers closes. Announce.
- April 21—Begin reviewing submitted papers.

- April 27—Final list of papers decided. Notify all those who submitted papers of their success/rejection. Request confirmation of attendance. Begin developing the schedule.
- May 4—Announce the schedule.

When you set this collection of deadlines, you should ensure that your fellow organizers can see them, too. A great solution to this is to set up a shared calendar on a service such as Google Calendar and ensure that each of the organizers has access.

Step 4: Make Time

The world is becoming an increasingly busy place, and time is an ever-rarer resource. You are a busy person, too: you are involved in your community; you are working to improve, enrich, and inspire; and you have your own set of tasks and responsibilities to look after. With so much going on, making the commitment to run an event can be daunting. Every event organizer I have ever spoken to about their first few events has always waxed lyrical about how much more time was required than they expected. You should ensure that you block out a good amount of time so you can give your event the right amount of care and feeding that it deserves.

GOOGLE AND EVENTS

Leslie Hawthorn leads the outreach side of Google's Open Source Programs Office, which focuses primarily on Google's student programs, the Google Summer of Code for university students and the Google Highly Open Participation Contest for precollege students. Leslie has also been involved in the organization of many conferences, including the Ubuntu Developer Summits in 2006 and 2008, MeetBSD 2008 conference, GitTogether '08, LugRadio Live USA 2008, and DjangoCon 2008. Leslie's team has hosted more than 100 community events at the Google HQ over the past three years, and has worked with community members on hundreds of sponsorships for other community events during that time frame.

Leslie is intimately familiar with the problem of not allocating enough time for an event:

“My basic rule on timing is to assume everything, from writing up conference call notes to catering setup, will take at least two hours longer than I've planned and to build in that margin for error into the whole of the event plan. It's far better to have an extra few hours to brainstorm, have a coffee, and take stock than to run around fixing issues and leaving out details at the last minute. It makes for a much better experience for the event organizers, as well, as they can focus on the pleasure of interacting with and learning from the attendees instead of putting out fires.”

When considering how much time to allocate to your event organization, you should not only consider the preparation time needed, but also how much time you have available when the event is happening.

The days that the event is running are an incredibly stressful time for an organizer, and many of the frustrations and difficulties are caused by the pressures of time constraints. Always try to factor in

more time than you need: having everything ready and waiting is a lot more fun than running around like a headless chicken panicking about time. Leslie also has some words of wisdom on this topic:

“People generally underestimate the amount of time it will take to get things done, even when there are many hands to help with planning and execution. I’ve found this situation to be especially true the day of an event; folks often don’t leave themselves enough time for setup and attendees are left standing around waiting for the party to really get started.

“Folks also tend to assume that a larger group of folks organizing logistics is better, but I’ve found this scenario only holds true when you have one or two charismatic folks giving direction to the rest of the organizing team. Without clear instruction, people tend to get distracted by less important details or talking with friends, and a great organizer knows both how to detect these lags early (and often) and how to politely refocus efforts on the task at hand.

“Also, your event managers need to remain available to direct traffic rather than jumping into the fray until the finishing touches are required; shifting focus from directing to doing can leave some folks unclear on what should happen next, particularly when they’re in a brand new location and are unfamiliar with how best to utilize the space or a venue’s processes for setup, etc. When people aren’t sure what to do, this leads to distraction for those team members who are focused on a particular task. As tempting as it is to jump in and help when managing an event, it’s best to hold back and make sure the organizing team feels empowered to get things done and clearly understands what’s expected of them.”

Time is an important resource. Make sure you have plenty of it at your disposal, and it will put your event in good stead.

Organizing Physical Events

While I don’t have space to cover all the types of events you might want to hold, I’ll focus on the most popular and common events:

Sprints

Common in the software development world, a sprint is when a number of (traditionally online-based) developers get together to work in the same physical location. The foundation of a sprint is merely to be in the same room working, but people naturally use the opportunity of being in the same location to have discussions, solve problems, and carry out other activities.

Summits

Summits are organized events in which the members have a series of discussion and debate sessions. Summits rarely have prepared presentations where someone speaks and the audience listens. Summits are instead a series of discussion sessions that hopefully result in an outcome. We will talk more about summits a little later in this chapter.

Conferences

Conferences are composed of a series of presentations in which a speaker talks about a topic and the audience listens. Conferences are primarily useful for disseminating knowledge to others. Conferences are also a useful focus point to get people together for other events. Some communities deliberately collocate other events, such as sprints, at times and places near conferences that many people will be attending, in order to make it cheaper and easier for potential attendees to visit both.

Cons

Cons (usually short for conventions) are popular in the user community, particularly in story-related communities such as movies and comics. These events are focused on fans coming together to celebrate something (e.g., *Star Trek* fans coming together to discuss the show, showcase their outfits, and participate in other strange activities).

Release parties

Release parties celebrate the release of something. These parties are common in the software, movie, and gaming worlds. These parties are often social gatherings in pubs and restaurants and have real value in building community. Some release events are fuller and incorporate presentations, workshops, and other features.

Each type of event is composed from broadly the same ingredients: a group of people meeting at the same venue to do something. The way they differ is in the focus of the event, be it working together (sprint), discussing ideas (summit), sharing knowledge (conference), coming together to meet like minds (cons), or celebrating something (release parties).

It should also be noted that physical events are excellent opportunities for socializing with your community. With each of the events, there is invariably an adjoining set of social events and parties to get your community to enjoy each other's company.

Common Attributes

Although it may seem like the different types of physical events are very different, they all share some common ingredients. Before we move on to look at the specifics of some of these different events, I want to discuss some of these shared ingredients that apply to *all* physical events. When you have a firm idea of how these ingredients will look, you will be 90% of the way to having your event ready.

NOTE

For each of the following elements, ensure that you document all the details online on a website. Your event website is a critical component of your physical event, and visitors will expect it to be up-to-date.

It is also recommended that you have a private organizers' website that can act as a home for the many organizational details that develop as the different parts of the event are put together. Ensure this information is there, too.

Location/venue

All physical events need a home. The first choice you need to make is the location. If you are organizing an event in which you expect your community to fund their own travel to attend, you should try to choose a location that is as convenient as possible for most of your members who are likely to attend. As an example, if most of your members are in the Eastern United States, hold it there. If most of your members are in Australia, hold it there.

Choosing a venue for your event depends heavily on the kind of event that you are organizing. If you are holding a sprint, a small venue is likely to be suitable, such as a hotel meeting room. If you are organizing a large conference, you may need a larger and more complex venue.

In addition to choosing the size of the venue, you should consider some other, subtler elements:

Public transportation

Can your attendees get there on public transportation, and can they get elsewhere in the area, too?

Distance from airport/train/bus stations

Many of your attendees may be flying in from other countries or getting trains/buses to the event. You should try to aim for a venue that is a reasonable distance from those transport links. If people will incur a hefty taxi bill to get to the venue, you will get some frustrated attendees.

Distance from accommodation

Many people who attend an event may be in a hotel. Is the venue within a reasonable distance? There are few things more frustrating than a venue distant from the accommodation.

Eating/drinking

If you are holding an all-day event (as many of these events are), you should ensure that your attendees can easily get to local bars and restaurants. Ensure that the range here is as flexible as possible, too: are there vegetarian, vegan, and halal options? Many of your attendees will want to get together for some drinks and dinner. You should ensure the bars and restaurants are within reasonable distance, and if not, ensure public transportation can get people to them.

Cost

Venues range hugely in terms of cost. Here you should think imaginatively: consider some less obvious venues as suitable options. As an example, when we were organizing LugRadio Live in the UK, we considered options such as bars, music venues, and outdoor events. Unfortunately, desirable choices for many of the other items on this list tend to require a high-cost location, because that's the kind of location near airports, nice restaurants, and so on.

A subtle consideration when choosing a venue is how your audience will perceive it. As an example, if you run a business community, you may well want to choose a venue that is more

professionally oriented, such as a business center or hotel conference facility. If your community is more low-key, you will have more flexibility with venue choice.

When I was involved in organizing LugRadio Live, a loose, social, and informal event that was part of our LugRadio podcast, we discounted many venues due to the “feel.” We were explicitly looking to achieve a social environment for the show, and many available places didn’t fit, including university lecture theaters and hotel conference facilities. We instead chose student union bars, independent theaters, and football stadium facilities: the latter all immediately expressed a more social and fun feel than the former.

Another consideration is the environmental impact of the venue on the mood of the event. When we organized the Ubuntu Developer Summit in Seville, Spain, the venue was in a large underground conference facility in a hotel. There were no windows, making the summit feel more constrained and causing people to tire earlier. The lack of windows meant that attendees could not see the daylight, and as such could not judge the time of day as the light changed.

Accommodation

For many physical events, people from out of town will attend. This will usually mean that they need to stay in a hotel. Many will expect you to tell them about reasonable hotel options, so you should provide these on your event’s website.

There are three tasks that you should work on in terms of accommodation:

Provide a range of options

Everyone has different budgets when it comes to hotels. Provide details for a range of room rates and quality.

Negotiate a special room rate

Call each of the hotels in the area and ask them if they can organize a special room rate for you. Many hotels will provide a room rate if you ask your attendees to give the hotel the event name or a special code. One thing to be cautious of: some hotels want you to reserve a block of the hotel for the discounted rate. This is risky if this is your first conference. Whatever you do, do not put an unaffordable deposit down to reserve the rooms with the expectation that people will come. It is too risky.

Document your hotels

Put a travel page on your website that contains all of the hotels that people can stay at. List the hotels by order of the largest discount on the room rack rate. Ensure you include the full address of the hotel (with zip/postal code so people can put it in their GPS units), and also include the telephone number.

You will probably find one or two hotels that are particularly accommodating (pun intended) to you and your event. Many of these hotels are hoping for repeat business in future years if the event becomes an annual fixture. These relationships can often blossom with positive effects on your event over the years. When we organized LugRadio Live, our regular hotels

would go further and further out of their way to provide a great service for us and our guests. Keep these hotel representatives in your circle of professional friends.

You should always pick primary hotels for your event. When traveling to an event from out of town and not knowing any of the natives, your attendees will want to be in a hotel with other attendees so they can socialize in the hotel bar, share cabs, and coordinate other activities. These primary hotels at events become a social hub. As such, when deciding which hotels will carry this role, ensure you check facilities such as bars, restaurants, and fitness centers.

BARGAIN HUNTER

When you ask for discounted room rates, it is likely the hotel will push back at first. Persevere. Tell them how important your event is, and tell them that you really want to recommend their hotel as “an” (not “the”; that would be lying) official hotel.

Keep asking and pushing for the discount. You have nothing to lose and lots to gain. If you get that discounted rate, it will make your event more affordable and allow more people to attend.

Equipment

Every event has equipment requirements. This can range from pens and paper through complex IT and networking facilities. You should ensure you have a clear idea of your equipment needs and that you have a means of acquiring them in time for the event.

Although seemingly a simple consideration, figuring out what you need to take to the event may be harder than you think. Imagine you are organizing a sprint. You may think of some of the obvious equipment requirements such as whiteboards, pens, networking equipment, cables, and power strips. There are many less obvious pieces of equipment, however, that you may not have thought of. This could include signs to show people where to go, tack, sticky tape, pins, lights, converter cables, speakers, notepads, whiteboard dry wipes, and more. These simple and easily forgotten parts can make a big difference!

To ensure you don't forget anything, you should note down your full equipment needs and consult with your community to see if there is anything you have forgotten. And like all your planning, the resulting list will be a useful reference point for future events.

Date/time

You should announce the date of your event at least four months before it happens. For events where you expect international visitors, I recommend you provide six or preferably eight months' advance notice so there is enough time for visas to be organized. Don't let the wheels of government cause problems. International visas were a scourge on events for the first few

years after the attacks of September 11, 2001, and can still get in the way of travel between some countries.

When picking a date for your event, you need to decide on two things:

How long

How long will your event last? We will discuss this in more detail later when we cover specific types of events.

When

Picking a time can be complicated. You need to ensure that your event does not conflict with events of similar interest, that the dates are suitable for the venue, and that they do not conflict with any holidays. You will never completely bypass every competing event and holiday, so use your best judgment.

One important tip that so many conferences and events still seem to get wrong: always put the date of your conference on the front page of your website. Don't bury the date three pages deep into the website. It will frustrate potential attendees and might suppress attendance.

Cost

One of the most difficult decisions to make for many events is whether to charge for entry, and if so, how much. Much of this discussion is entirely dependent on the type of event that you are running, its expected attendee demographics, and whether the event seeks to make a profit.

This was something we considered heavily for LugRadio Live. The event was very much intended as a community event. It was never intended to make any profit. The only monetary concerns were covering the costs of the event (and much of this was reduced due to sponsorship, which we will discuss later).

One conscious decision for LugRadio Live was to make it affordable for everyone. We were not especially big fans (to say the least) of open source conferences for which community members were expected to pony up \$800 to attend. We felt that this produced a false economy: a conference in which significant portions of the demographic of open source members were unable to attend.

On the other hand, even though we had enough sponsorship to make the entrance to LugRadio Live free, we decided to charge a nominal fee (£5 at the door). The reason for the charge was subtle: if something is free, people will rarely commit to it. Even with a tiny cover charge, it would mean that people who registered to attend would actually show up.

Registering attendance

For many events there should be a means for attendees to let you know in advance that they are going to attend.

Having this information is useful for a few reasons. By far the biggest worry in organizing an event is whether people are actually going to show up. This is typically less of a worry for invitation-only sprints, but a significant cause of nervous twitching when organizing a conference. It is always advisable to have a way for people to register their attendance so you have an idea of where your numbers stand. Another benefit of knowing these numbers is that if you have an outright success on your hands and are likely to have many more people attending, you can make preparations to deal with the increased traffic at the event.

It is important to remember, though, that conference preregistration figures often don't reflect the final attendance, particularly for events that are free or where people can pay at the door. In these cases, where there is simply no justification for them to preregister, there needs to be a reason and a catalyst, such as limited availability or discounted ticket prices. Justification in the eyes of your prospective attendees is not "it helps the organizers know how many people are coming"; justification needs to be something the attendee will feel is a personal benefit. This is where the cost of your event plays a large role. If you are charging \$1,000 for a conference ticket and preregistration allows people to get a \$400 discount on the ticket, you will see more preregistrations.

Although many formal events keep attendee lists private, or share the lists only with other attendees, many community events ask attendees to register publicly so everyone can see who's coming. This provides a small incentive to register in advance: attendees can then encourage colleagues whom they want to meet to come.

Events have different approaches to registering attendance. For the majority of you reading this book, I am willing to bet that you are primarily organizing small summits and sprints for your community. In these cases, a wiki page is a perfect solution: simply ask your community members to add themselves to a page if they are planning on attending.

For much larger conferences and sprints, you may want to use a conference management system or a custom website to handle the attendance registration process. You also may want to include an online payment process with your registration facility.

Catering

You need to decide whether you want to have food at your event. For a daytime event, this is most typically lunch and breaks. If it is an evening event, the catering is typically dinner and drinks.

Catering can make the cost of the event balloon, so you need to consider it carefully. An added complication is that many venues will restrict you to using only their on-site kitchen or a specific vendor. You should check this when investigating the venue. Such locked-in vendors usually lead to outrageously inflated costs, but you can't really blame the venue, because it has its own advantages in maintaining a relationship with a vendor who gets to know it well.

Expectations around catering vary. For a small, locally organized event, catering is not typically expected. If you are organizing a large professional conference, lunch will be expected, particularly if you are charging a significant entrance fee. Irrespective of what you decide, it is highly recommended that you have drinks available: people get thirsty throughout the day, particularly if there is lots of discussion.

If you do decide to cater your event, a buffet-style format is recommended for both lunch and dinner options. It is easier to organize, promotes socializing (instead of people sitting at the same place at a dinner table all night), and is often cheaper.

IF YOU DON'T CATER...

...you should ensure that you provide details of local, reasonably priced food and drink providers. I always recommend providing a list of fast food restaurants as well as sit-down restaurants and bars. Again, the key is to provide a range of options that can suit as many tastes and budgets as possible.

Insurance/unions

One area that you should properly investigate when planning your event is the insurance needs and possible union requirements for your area. These issues vary tremendously around the world.

I experienced this firsthand when comparing the experience of organizing LugRadio Live in the UK and in San Francisco in the U.S. The events were a world apart in terms of insurance and union requirements. The San Francisco event was a far more complicated affair with more rigorous and complex requirements than its UK counterpart.

With the laws and requirements varying so much around the world, it is difficult for me to give any concrete advice other than to ensure you check these legal elements thoroughly. If you are unsure, ask the venue management for its advice.

NOTE

You should also make a point of looking into basic first aid facilities. If someone slices his hand open or passes out, you want to be able to react appropriately. Preferably you should have someone available on site who knows first aid.

Organizing a Sprint

Sprints are events in which your community gathers together to work in the same physical location. Sprints rarely have any special time-tabled content, such as talks or presentations, and their primary focus is more to get people working individually or on shared projects, taking advantage of the face-to-face time to have discussions and solve problems together where required.

The primary requirement for a sprint is to provide a working environment. This is dependent on the kind of work your community performs, but for most communities it's likely to be a conference room with tables and chairs where your community can sit together to work.

I have organized a number of sprints for different purposes, including specific team sprints for my own team and wider community sprints. You should always allow a sprint to be a sprint. Over the years I have seen many people organize sprints and instead try to turn them into more of a summit, replete with brainstorming sessions. I was one of them. When I used to organize sprints for my team, I would run them largely as mini-summits, and we would discuss and flesh out plans for the coming cycle. At one wider team sprint I could not be there with my team, and they reported back that having the opportunity to just work, as opposed to brainstorm, was hugely productive. Since then I have been careful to ensure that sprints are primarily focused on working together. To achieve this, I reserve the mornings for brainstorming and the afternoons for sprinting.

Additional notes

Here are some additional notes building on the topics I covered in general earlier:

Location/venue

Sprints are typically smaller events and require smaller venues. Bear in mind that the sprint is intended to be a working session, and therefore you may need facilities for this work, such as Internet access, plenty of power outlets, and tables and chairs. For a small sprint the venue can be informal and loose, with many communities sprinting in houses and apartments.

Accommodation

There are no special accommodation requirements for sprints.

Equipment

The equipment requirements depend on the type of work that you will be doing. As an example, if you are doing software development, people should bring their own laptops, but you may want to provide blank media, commonly requested cables, power strips, USB storage devices, etc.

Date/time

The length of the sprint should reflect how long you can reasonably expect people to be together. Do remember that people will need to book time off work to be there, and some may be away from their families and children. I recommend that a sprint range from three to five days. I would not recommend a sprint that lasts longer than a week.

Cost

To the best of your ability, sprints should be free. You should seek to cover your costs with sponsorship if your organization can't pony up the funds itself.

Registering attendance

You will want to get a firm idea of who is coming, but also make the sprint open so anyone is welcome to attend. Specialist sprints will require specific invitations to the people whom you want to attend.

Catering

Many smaller sprints provide either a buffet lunch or defer lunch to nearby restaurants. You should ensure there are plenty of water and cups available, and preferably some sodas or coffee. Some caffeinated beverages are particularly important if you have a long sprint: people will rely on them to wake up in the mornings.

Insurance/unions

As always, check into the insurance requirements for the sprint. There are unlikely to be union issues.

Organizing a Summit

Summits are organized events in which your community gathers to discuss and debate a set of topics with the purpose of developing an outcome.

An example of this is the Ubuntu Developer Summit (UDS). During every release cycle, the Ubuntu community gathers together to discuss, debate, and design the next release of Ubuntu. The summit is broken into a number of tracks (community, desktop, server, mobile, quality assurance, etc.), each containing a number of sessions. Each session has a leader who focuses the discussion on the topic, and the attendees weigh in and discuss the best choices. By the end of the session the expected outcome is a broad solution that can then be documented as a specification from which the community can work.

The UDS is a large and comprehensive summit that has been running for a number of years. It attracts more than 200 attendees, involving a huge range of sessions over multiple concurrent tracks, and remote participation. Organizing and running the event is a large and comprehensive undertaking, and it would require another book the size of *The Art of Community* to discuss all the details. I do, though, want to distill some of the key lessons learned from organizing UDS that you can apply to your own summits.

Structure and scheduling

The primary goal of a summit is to ensure people can discuss and debate topics to the point of producing a solution. For this you need to have a template for how many sessions will appear in your schedule. As an example you might have the following schedule.

8:30 a.m.	Doors Open
9:00 a.m. – 10:00 a.m.	Session
10:00 a.m. – 11:00 a.m.	Session
11:00 a.m. – 12:00 p.m.	Session
12:00 p.m. – 1:00 p.m.	LUNCH
1:00 p.m. – 2:00 p.m.	Session
2:00 p.m. – 3:00 p.m.	Session
3:00 p.m. – 3:30 p.m.	Break
3:30 p.m. – 4:30 p.m.	Session
4:30 p.m. – 5:30 p.m.	Session
5:30 p.m. – 6:00 p.m.	Closing/Wrap-Up

This schedule is productive but not too taxing on your visitors. It provides six session blocks with breaks for lunch and other rest, ensuring that your attendees don't go for longer than two hours without a break, which is always recommended.

If you find that you are going to require more than six sessions on one day, you may want to consider multiple tracks—sessions running at the same time. As an example, at the last Ubuntu Developer Summit there were seven concurrent tracks. Of course, there is an obvious downside to this approach: people can't be in two (or seven!) places at the same time. With this in mind, you need to ensure the tracks are distinct and will attract different interests.

When having multiple tracks, you should account for a lot of time spent moving around.

Inside a session

For each session, ensure that the room is set up to encourage discussion. Chairs should be in circles or around tables, not facing front, which implies only the session leader matters.

Each session should have two people serving specific roles:

Leader

The leader is responsible for ensuring that the session is kept on-topic. Sessions can easily get sidetracked, so the leader must be prepared to bring discussion back. The leader should also ensure that everyone gets a chance to speak and that the session drives to a conclusion.

Every session should result in a set of actions to implement the work that was discussed in the session.

Notetaker

In the interests of referencing the session as well as transparency and openness for people who can't attend the summit, a person should take notes about what was discussed and what the concluded actions were. Some sessions find it useful to put notes up on a bulletin board or easel. In any case, the session leader should not be the note taker; each role requires full concentration and a different kind of thinking.

Always be conscious of community members who can't attend sessions. Your notes should help them feel a part of the session, but you may also want to consider options for remote participation. This could be as simple as a conference phone that people dial into or as complex as a video link. Twitter and/or identi.ca could be great options for this. Physical attendees could post messages of what is being discussed, and online attendees can then read and respond.

I have been involved with a variety of methods of remote participation. A conference phone to dial into is always a useful option: it is low maintenance and requires little conscious thought. A video link is more complex and more intrusive in the session because it relies on members of the session to operate the camera so that it points at the person who is speaking. Another possible option is an audio feed that streams onto the Internet. People can then respond to topics via an online chat service such as IRC, or even a microblogging service such as Twitter or identi.ca. You should evaluate your options and see what is doable with the resources and time that you have available.

Event-specific notes

Location/venue

Summits are typically small to mid-size events and often include multiple concurrent tracks that will need rooms. You may also want to provide a morning plenary presentation for all the guests, which will require a larger room. Bear in mind that the summit is intended to be a working session, and you may therefore need facilities such as Internet access, plenty of power outlets, and tables and chairs.

Accommodation

There are no special accommodation requirements for summits.

Equipment

Summits are generally entirely discussion-led, but you may need to supply an Internet connection, data projectors, whiteboards, writing pads, and pens and pencils.

Date/time

The length of the summit should reflect how long you can reasonably expect people to be together. Do remember that people will need to book time off work to be there, and some may be away from their families and children. I recommend that a summit range from three to five days. I would not recommend a summit that lasts longer than a week.

Cost

To the best of your ability, summits should be free. You should seek to cover your costs with sponsorship if your organization can't cover the funds itself.

Registering attendance

You will want to get a firm idea of who is coming, but also make the summit open so anyone is welcome to attend. As such, it is recommended that you have an attendee list but also publicize the open nature of the event.

Catering

Many smaller summits provide either a buffet lunch or defer lunch to nearby restaurants. You should have plenty of water and cups available, and preferably some sodas or coffee. Some caffeinated beverages are particularly important if you have a long summit: people will rely on them to wake up in the mornings.

Insurance/unions

As always, check into the requirements for the summit.

CASE STUDY: GOOGLE SUMMER OF CODE MENTOR SUMMIT

For some time Google has been running a program called Google Summer of Code, which provides funding for open source projects to develop new code, features, and initiatives. The program has been overwhelmingly successful. Hundreds of projects have benefited, and millions of dollars have left Google's ample wallet as part of the program.

Each year Google invites two individuals from each successful project involved in Google Summer of Code to their annual Mentor Summit at Google HQ. Leslie Hawthorn of Google reports:

"I'm particularly proud of the feedback we've received on the Summits, our attendees repeatedly telling us that the connections they make that weekend lead to collaborative development between projects. It's also an excellent opportunity for these seasoned Open Sourcerers to share best practices and not just around participating in Google Summer of Code; some of the most important knowledge-sharing that takes place at the Summits is when contributors finally get to meet face-to-face; exchange ideas; and form the social ties that cause patches to be reviewed and merged more quickly, requests for support answered rapidly, etc. The typical view of FLOSS (Free/Libre and Open Source Software) development is that everything takes place online, but that's only a part of the interactions that fuel Open Source; without these in-person meetings, establishing a reputation and securing the trust of one's fellow project members certainly occurs, but much more slowly. Bringing together these experts from each project helps everyone to build rapport rapidly, greasing the wheels of online activity through social bonds.

"We primarily organize the Summits by using our mailing lists and a wiki. Each participant is encouraged to propose sessions in advance of the unconference, with all suggestions collected on the wiki; the mailing list is primarily a vehicle for reminding folks to update our shared online resource. Once attendees arrive, we ask them to write out their ideas for session topics on large

pieces of paper, which are then posted for all to review. Each attendee is given sticker dots so that she can +1 sessions, meaning adding a dot to the topic's poster signifies interest in attending a particular discussion. This system allows us to easily determine which sessions require the largest amount of space for participants, which is particularly handy when managing logistics for an unconference, as there's no set agenda in advance. The posters also give us all the opportunity to discern which ideas are most compelling—and which challenges are most daunting—within our community. By structuring the meeting as truly participant-driven, our attendees are guaranteed to get precisely what they need from their participation provided they put energy into the wider discussion.

“During sessions, we encourage everyone to take notes on the conference wiki to most widely share what they've learned. As is typical during any conference, there are many more sessions that attendees would like to go to than they actually are able to, so these notes allow folks to glean something from the discussions that took place and to know whom to follow up with later for additional exploration or collaboration. The good folks at Oregon State University's Open Source Lab host this community wiki for us, which is globally readable so that everyone has the benefit of our collective experience. Several community members have volunteered to administer the wiki and are actively (re)organizing the content so it's most useful to would-be Google Summer of Code participants and anyone else looking to run a similar outreach program. See http://gsoc-wiki.osuosl.org/index.php/Main_Page.”

Organizing an Unconference

Unconferences are a relatively new addition to the menagerie of commonly organized physical event types. In its goals, an unconference appears to be the same as a normal conference: a group of people gather in a venue to watch a series of talks and discussions. There is, though, one key difference: an unconference has its schedule created on the day of the event in an entirely free-form way. One such example of an unconference is the Community Leadership Summit, an event which I organize each year to bring community leaders, managers and organizers together. You can find more details about the event at <http://www.communityleadershipsummit.com>.

The history of unconferences traces back to O'Reilly's invitation-only geek event, FooCamp (“Friends of O'Reilly” camp). The real success and growth of unconferences has been exhibited by the BarCamp spin-off events.

BarCamp was originally a joke between Chris Messina and a couple of friends regarding some somewhat disgruntled people who had not been invited to O'Reilly's FooCamp. Curious as to why these people were complaining, Chris and friends decided to run their own equivalent event, coining it BarCamp, as a nod to the (ironic in this case) foobar references in O'Reilly books. Chris notes how the event came about:

We just thought it'd be fun to get a bunch of friends together and have an emergent (or "open space") event to offset FooCamp. The crazy thing is that we planned the whole thing in only six days. I was on Instant Messenger and email and calling people trying to get a venue: originally thinking of doing real camping in the mountains! When nothing panned out, Ross Mayfield from SocialText saw Andy Smith's call for a space, realized that we were just down the street and offered up his new space down the road. Once we had a venue, everything fell into place.

Since such humble beginnings, there have been over 500 BarCamps to date in all the major inhabited continents. There have also been less-nerdy spin-off events, with one such example being WineCamp, a derivative of BarCamp in which a mix of nonprofits and technology fans camp out at a vineyard with no water or power. Chris noted that not having Internet access and power was actually a boon:

On the second day of the camp, we went to a winery where we had wifi and power and worked on producing all the ideas we'd brainstormed offline the previous day. It was seriously productive and hugely interdisciplinary. It's events like that that blur boundaries and encourage diversity that I think are the most rewarding to me.

Unconferences are an excellent way to host discussions that everyone has the opportunity to drive. They are by definition intrinsically open events. By allowing your guests to set the schedule on the same day, you are opening up the event to all manner of topics, even if attendees prepare a session before they arrive and volunteer it.

From my experience with unconferences, the free-form scheduling always uncovers unusual and intriguing topics. There will be topics proposed that you or a wider scheduling body would have never thought of, and this can make for some really interesting and intriguing discussion.

Fortunately, unconferences are devilishly simply to organize. In addition to the obvious resources, such as a venue and attendees, the primary consideration that you need to account for is a place where people can add their sessions to the agenda. Most unconferences feature a large whiteboard (or two whiteboards side-by-side) on which you write the conference grid. This is a box that shows the rooms along the top and the times down the side. This will result in a number of session slots in which people can write in their sessions. The whiteboard should be in a central location that's easy for people to check regularly.

NOTE

Normally with an unconference there'll be 100 or so attendees and quite a lot of talk tracks (say, 7), so each talk is expected to attract only 10 or 20 people; this means you need lots of small rooms, not one big one, and it means that a talk is less intimidating for a speaker because talks normally end up being discussions anyway when there are only 6 of you.

Another consideration is to provide a wiki and other resources to wrap around the event. Even Chris Messina, one of the originators of this style of event, likes to keep things simple:

[For BarCamp] we really relied on the wiki, the mailing list, blog posts and photos posted to Flickr. Early days, I'd say that made up 98% of the documentation. There was also word of mouth of course—individuals became spreading vectors in and of their own right. The rules themselves were also fairly viral—I mean, we basically stated, along the lines of *Fight Club*, “If this is your first time at BarCamp, you must present!” We weren't draconian about it, but that was an important aspect of the event: no spectators.

Given the slightly unusual format of an unconference, it is recommended that you attend a few of these events before you organize your own.

Event-specific notes

Location/venue

Although many unconferences are called camps, a campsite is not typically required as a venue. Provide a number of breakout rooms to have the different sessions in. These rooms will need to have tables and chairs.

Accommodation

Again, camping facilities are not typically required. If you do want to have a camp, ensure you have an area in which your attendees can pitch tents. Also ensure there are toilet and washroom facilities. Some unconferences that take place in offices allow people to sleep on the office floor. If you do this, ensure you remind people to bring sleeping bags.

Equipment

The main equipment that you will need are a large whiteboard and dry markers that your attendees can use to contribute their sessions to the schedule.

Date/time

Unconferences are typically no longer than one or two days.

Cost

Costs vary between these events.

Registering attendance

Unconferences vary: some are open events with open attendance, and some are closed, invitation-only events.

Catering

Many unconferences provide either a buffet lunch or defer lunch to nearby restaurants. You should have plenty of water and cups available, and preferably some sodas or coffee. Some caffeinated beverages are particularly important if you have a long unconference: people will rely on them to wake up in the mornings.

Insurance/unions

As always, check into the insurance requirements for the venue.

Getting Sponsorship

Everyone reading this book is lucky, because community is one of the rare places in the world in which we can exist, build bridges, and reward victories without bowing to the filthy lucre. In our palm-lined oasis, money is rarely a consideration. Sometimes, though, it is.

The vast majority of physical events cost money. Venues need to be hired, equipment needs to be rented or purchased, and other costs need to be covered. Unless you are charging an entrance fee that covers costs or someone you know is feeling particularly generous, it is likely you are going to need to find sponsorship to cover these costs. Let's now take a little time to talk about what is involved in finding sponsorship.

Understanding Your Needs

Sponsorship is a somewhat hit-or-miss process. Although your community does good and valid work, what you are essentially doing here is asking someone for some free money. It doesn't matter that the person you are asking may be a large company. Large companies still need to account for the purposes of their expenditures, and particularly when the economy is facing difficult times, justification has never been more important.

Fortunately, I have a surefire way of improving your chances at getting sponsored. This is a theory that has been designed, refined, tested, and further refined to present a cookie-cutter concept that can be applied perfectly to your community—a cookie cutter built from reason, experience, and perfected mathematical ingenuity:

The less money you ask for, the more likely you are to get it.

Pretty stunning stuff, eh?

OK, back to the point. Consider yourself in the position of Chief Checkbook Holder for a large organization. If a community comes up and asks for \$500 and another community asks for \$5,000, which are you more likely to scrutinize? From which are you more likely to demand your money's worth in associated advertising and favors? Which are you more likely to say no to?

NOTE

Although this theory can be applied in a general sense to most volunteer community events, if you are organizing a large conference event (particularly if the audience is composed of professionals), asking for more money can legitimize the event. Although true, you should tread carefully with this approach: get it wrong and you may get nothing.

If you are considering asking for a significant sum of money, I recommend you speak to someone with event organization experience first and get that person's take before you push the Send button on your proposal to the sponsor.

As such, you need to perform an exercise in cost cutting. You should first produce a big list of everything that is going to cost money, either in rental or purchase costs. This list should be accurate and complete. Everything from pens to the venue to expensive computing equipment should be on that list.

Your next step is to go through the list and turn it into a trimming exercise. The goal here is not to remove things that you actually need, but instead to find ways to source those things without paying for them. I know I shouldn't need to say this, but people...stealing is not an option. If I get a letter from my attorney telling me that a community project leader is in prison for stealing 20 rack-mount servers and is pinning it on "the British dude who wrote *The Art of Community*," I am not going to be particularly happy.

What I am instead suggesting is that you try to source those things by borrowing, sharing, making, or otherwise gathering. Follow the general theme of the book here: think outside the box. Inspire yourself to get as much of what you need as you can without merely going and paying for it.

When we started organizing LugRadio Live in England, we did a lot of this. We borrowed projectors from some friends, and the screens from others. We got donations of paper and pens from an office worker with plenty of spare supplies; we produced the name badges, posters, and programs ourselves; we asked a conference to give us spare lanyards, as they didn't need them; we provided the audio equipment ourselves; and we used a cut-out potato and ink to stamp the hands of attendees (really). Although some of this may seem a little cheap, what many of you want to achieve is not a large business conference. We are a volunteer community, and it is OK to be a little rough around the edges. Rough around the edges and endearing are close bedfellows (no one has ever forgotten that potato stamp...).

When you have been through your list and have a final tally of things that you just have to rent or buy, calculate the final cost of those elements. You now have a much lower figure to request sponsorship for, and you are much more likely to get it.

SWEAT VERSUS SPONSORSHIP

I hate to belabor the point, but after my previous diatribe about thinking outside the box to source what you need without having to ask for sponsorship, I know many of you will think, "Well, it's just going to be easier to ask for the sponsorship."

It is easier. I am not denying that. But being frugal with the mighty buck is not only a positive exercise in how to put together an event, but importantly it is the *right thing to do* for your sponsor.

If you treat them with respect by asking only for exactly what you need, you will be putting down the seeds for a long and fruitful relationship.

Finding and Handling Sponsors

By now you have your sponsorship figure. The next step is to determine how many sponsors are likely to be able to cover it. Of course, anything to do with figures is difficult to provide concrete advice for, so take some of these words as general guidelines only.

If you need \$2,000 or less, it is likely that you can find a single sponsor who can probably cover the full cost. Still, you may want to consider breaking the figure in half (e.g., \$1,000 per sponsor) and therefore asking each sponsor for less. Remember the golden rule:

The less money you ask for, the more likely you are to get it.

When you have an idea of how much you want to ask for from each sponsor, you can begin thinking of potential sponsors.

The best bets for potential sponsors are companies that are related to your community's activities. As an example, if you are an open source community, there is a raft of open source companies and wider IT companies with an interest in open source. Naturally, another indicator toward a potential "yes" on your sponsorship request is that the company has money. If the company is known to be struggling financially, save your energy and focus only on cash-positive organizations.

Determining potential sponsors can be a tricky road to navigate. The best way to do this is to run the idea of sponsorship by some people you know well in existing companies who are potential sponsors. They may be able to help you get up to the next rung in the ladder. Every time I have organized an event that needed sponsorship, this has been my first port of call.

Another approach is to meet someone involved in an existing event that is similar to your own and ask how that person handled sponsorship. Another useful technique is to look at the sponsorship lists for these other events. Often the primary sponsors are listed on the front page of the event website.

Setting expectations

When asking for sponsorship, it's important to remember that you are engaging in a business transaction. The organization giving you money expects something in return. Specifically what they expect varies tremendously between sponsors.

Some sponsors will expect almost nothing. A good example of this is a company called Bytemark Hosting, which has sponsored every year of LugRadio Live since it began. The company has provided venue sponsorship year after year, even back when no one knew the event or its expectations. Bytemark not only had faith in the event but was gracious in their expectations: all they wanted was an exhibition table.

On the other hand, some sponsors want the moon on a stick. Another (unnamed to protect the innocent) sponsor I have dealt with wanted branding scattered across the venue, weekly calls with their demanding marketing manager, regular branding mentions in the podcast that

we did, control over the size and location of their booth, and other requests. The experience with that sponsor was frankly a huge headache that none of the event organizers needed, especially with so much else going on.

You need to set your own balance of what to offer and ensure that sponsorship requirements don't impinge on the values of the event. For LugRadio Live, we decided on a set of opportunities that we would present to sponsors in exchange for sponsorship, offers that we felt did not compromise the ethics or atmosphere of the show. These included:

- Sponsor logo printed on the back of the program.
- Small sponsor logo printed on the back of the presenter and crew t-shirts.
- Sponsor logo and link presented on the event website.
- Exhibitor space at the event in a location where the most traffic flows.
- A thanks to the sponsor in the LugRadio live show in front of the event audience.

In addition to this, we also clarified with all sponsors that no editorial content or changes could be mandated by a sponsor. In other words, we would always have complete control of the content of the podcast, as before.

When you want to approach sponsorship, you need to have your own list of bullet points indicating what you can offer the sponsor. The ones I listed are a good starting point.

When you put together your own list like the one just shown, be sure to clarify how sponsorship intersects with editorial privilege. For instance, a logo is obviously a form of advertising and simply indicates you made a deal for much-needed financial support. In contrast, some forms of sponsorship are a bit insidious. When a sponsor gets the right to deliver a keynote, you are pretending to offer your attendees useful information when all you're doing is delivering them up as a captive audience for marketing.

The pitch

With your sponsorship figure and your set of bullet points to indicate what you will offer ready, you now need to put together your pitch. It should be a short document that outlines the event, what you need sponsorship for, and what you can offer the sponsor.

The size of this pitch should reflect the amount of money that you are asking for and the complexity of the event. If you are organizing a large conference with 2,000 expected attendees and are asking for \$50,000, you should sharpen your pencil and prepare a comprehensive, detailed, multipage sponsorship request.

I am willing to bet that 98% of you reading this are not in that position and are instead asking for a fraction of that amount for an event with no more than a few hundred attendees. As such, your pitch can be much more straightforward and can fit into a one- or two-page document or a single email to the sponsor.

In your pitch, you should include the following details:

Key information about the event

The name of the event, where it is located, the date(s) that it is happening, and the number of expected attendees. You should also do your best to describe your attendee's interests; sponsors want to know that their company logos will be seen by people who could become customers.

Purpose

Why the event is important and unique.

Requested figure

The required sponsorship amount and the date by which you need it.

Reason for sponsorship

What the sponsorship money will cover. Be honest here; don't say that \$500 is going to cover way more than it can reasonably cover.

What the event provides

List here the set of bullet points indicating what you can provide in exchange for the money.

Contact details

Include your name, email address, and a daytime and evening phone number.

Your pitch should be straight to the point, respectful, frank, and complete with all of the details just shown. When you have your list of sponsors and contact people, you should send off your pitch and cross your fingers.

NOTE

Whatever you do when emailing sponsors, don't email multiple sponsors at the same time.

In other words, don't send an email with a CC list as long as your arm. It is cheap and disrespectful to your sponsors.

Each of the potential sponsors that you send your pitch to should get an email directed specifically to that contact, personally addressed to the contact (e.g., "Dear Alan").

Handling the Money

If you manage to source an agreed level of sponsorship from a company or two, congratulations! The next step is to know how to deal with the money.

If you are dealing with a small sponsor, the transfer of the money will likely be quick and efficient. Some sponsors may just cut you a check or perform a bank transfer. Some sponsors may have more complex requirements and request invoices, purchase orders, or other paperwork.

When you agree to the sponsorship amount and conditions, you should ensure that you are entirely clear on how this process works. The reason for this is twofold. First, satisfying these

requirements may take time, and you want to ensure that this time is factored into your plans. What you don't want to do is get into a situation in which you have ordered a lot of equipment and resources and have not received the sponsorship money yet to pay for it. That happens way more than it should, so be cautious of it.

The second reason is that sponsorship can open up a rabbit warren of other headaches. As an example, consider this trail of dependencies:

1. To get the sponsorship money, the sponsor requires a bank account to transfer the money to.
2. To set up a bank account, you may need to be a registered organization in your country. This will require certain community members to be signatories on the account. And this in turn may require some kind of community assessment of who takes on these responsibilities.
3. To be a registered organization, you will need to file your taxes. This will require a formal paper trail. You will also need a regular reassessment of how well the members who are representing this organization are functioning.

None of this work is enjoyable. It is a painful bureaucratic necessity for accepting sponsorship money. You should ensure you are entirely clear on the ramifications for accepting the money from sponsors and what is involved. Keep it as simple as possible.

AVOID HEADACHES: INVOICE DIRECTLY

A great approach that can avoid the problems of managing money is to never handle money in the first place and simply have the sponsors invoiced. As an example, if you need to spend \$500 to hire a venue, just ask the venue to bill the sponsor. This means that the money never passes through your hands.

If you would like to pursue this approach, you should obviously ask whether the sponsor is happy to do this. Some sponsors are simply not set up for this method of dealing with events.

You should also ask the venue. Some venues will not invoice to people other than the primary contact for the event.

Organizing Online Events

In recent years the growth of the Internet has produced an increasingly interactive Web. Gone are the days of an Internet largely populated by static web pages. Today the Internet is the same thriving and growing library it ever was; it's just that now it is a library in which you can talk to the other visitors.

When we look back at the history of communities in which people worked together on the Internet, virtually all communication was handled in two environments: email or Usenet (groups in which you send email). Both of these media have always been slow. When you discuss a topic over email, it is entirely expected that a conversation can last days or even weeks, with hours in between each message.

Although email still dominates the world as a primary medium for general communication online, advances in real-time discussion facilities have made it possible to hold real-time meetings in which people converse together in the same time slot. This has raised the opportunity for online meetings in which multiple people can join and have a conversation.

Online events are something that I have used extensively throughout my work with Ubuntu, but it surprises me how little other communities have made use of them. In the Ubuntu community, they have been hugely useful and always netted upwards of 300 attending each event.

If you have a geographically dispersed community, here are some of the types of online events that could be useful to run:

Tutorial weeks

These are special weeks in which a series of teaching and best-practice sessions are run to educate your community in how to do things. This has been used extensively in the Ubuntu community.

Release parties

Many communities have online release parties to celebrate the release of a new piece of software, initiative, or some other project. Instead of meeting in a bar or restaurant, these parties happen online in a chat room, and people sit at their computers and have a few drinks while having a good time with each other.

Focused activity days

These days are intended to bring the community together around a specific initiative. In the Ubuntu community, these events come in the form of Bug Days (designed to focus people on fixing bugs) and Docs Days (designed to focus people on improving the community documentation).

I have not included team meetings in this, as they are less special events and more an expected component in teams and governance bodies. Online events related to governance and conflict resolution were discussed earlier in the chapters devoted to those topics.

Common Attributes

Earlier, when we explored some of the organizational characteristics behind physical events, we discussed some common elements that apply to all events. This included accommodation, date/time, equipment, etc. We are now going to do the same for online events. The following are the common considerations that apply to all online events.

Medium

The first and most important consideration is what medium you want to use to host the event. Each medium that you use will need to be in real time: it is the instant gratification of real-time communication that makes events feel like events.

These are some of the attributes that you should look for in a medium:

Appropriateness

Will the medium meet your needs? As an example, if you are running a training course for a few hundred people, a voice teleconference is inappropriate because only one person can talk at a time. However, you might deliver a talk and answer questions over a voice connection while accepting questions and allowing chat on a text medium.

Accessibility

You should ensure that all of your community members can access the medium you choose. This depends a lot on your community. With this consideration, you need to determine not only whether your community has the technical facilities (e.g., computing power and Internet bandwidth) but also whether members are familiar with or able to learn how to access the medium.

Values

You should ensure that the chosen medium meets the values of your community. As an example, if you are part of an open source community that values software freedom, you should not choose a medium that is closed source.

Whatever your choice of medium, you should ensure that access to it is well documented and that your community is fully aware of where to find the documentation and how to connect.

Internet Relay Chat (IRC). Examples include:

- Freenode (<http://www.freenode.net/>).
- EFNet (<http://www.efnet.org/>).

Internet Relay Chat is a simple and efficient medium that is popular in the technical community. IRC is also open and transparent, and there are free clients for all operating systems. Another benefit of IRC is that it can host literally hundreds of participants at the same time, yet is low-bandwidth: it does not require a fast Internet connection. This is important if your community members may be using dial-up Internet access.

IRC has two downsides. First, it is text only, and some may not find it quite so engaging. Second, it is largely unknown in nontechnical communities. As such, if you decide on IRC for a nontechnical event, you will have to tutor your community on how to use it.

I have found IRC to be an excellent medium for organizing events. I have used it for many Ubuntu-related events, such as the Ubuntu Open Week, Ubuntu Developer Week, LoCo Documentation Days, release parties, and more. While IRC is useful, I have always been conscious to remember that most members start out unfamiliar with IRC and how to use it.

You should always ensure there are nice, clear instructions (with screenshots) showing how to connect with IRC.

Voice over IP (VoIP). Examples include:

- Skype (<http://www.skype.com/>).
- Ekiga (<http://ekiga.org/>).

Online telephony such as Skype or a Session Initiation Protocol (SIP) client such as Ekiga is the equivalent of having a conference call. As such, the same benefits and limitations apply: you can't practically have more than 5–10 people in a conversation, but it does feel engaging. A downside of this medium is that it requires (a) a reasonably powerful Internet connection, and (b) sound hardware and a microphone, which may not be as common as you would expect.

I have found VoIP to be useful for meetings, but not for general-purpose events due to the scaling issues. Another blocker for VoIP is that while Skype works great for many people, other clients require a significant amount of fiddling with firewalls and other networking mumbo-jumbo to get them working. When you organize an event, the last thing you want is your attendees fighting with the tools they need to connect.

Virtual worlds. One example is:

- Second Life (<http://secondlife.com/>).

These 3D environments offer an interesting possibility for events. The largest and most popular is Second Life, which has literally thousands of people online.

Virtual worlds offer an interesting physicality to an event. Second Life, for example, has hosted many events inside its environment, including gatherings, concerns, book readings, presentations, and more. In a virtual world, you have a physical avatar that you can move around in the world. While there, you can text or voice chat with other in-world avatars, buy and sell items, and create buildings and other things.

A particularly interesting feature with the voice chat in Second Life is spatial sound, in which the location of the sound in the stereo field changes based upon where the person is. As an example, if you sit between two people in a discussion, the person on the left will come out of the left speaker more.

Virtual worlds do, however, require significant computing resources to use (high-powered graphics card and audio hardware, significant Internet bandwidth) and also a good knowledge of how to use the world, navigate, and communicate with people. Although clients such as Second Life seem like an exciting bridge between the real and the online world, you should ensure they don't block accessibility for your community. If the majority of your community can use Second Life, great, but if many struggle to meet the requirements, choose a different medium.

Date/time

When choosing an event date and time, be conscious that you are potentially open to a worldwide audience. As such, you are faced with the complex task of picking a time that will suit most of your likely attendees.

Here you need to take a reasonable survey of where most of your attendees live in the world and what times are going to be suitable for the majority. You are never going to make everyone happy all the time, and some people will complain because they won't be able to attend.

A common approach I have used is to pick a time that is in the afternoon in Europe. As an example, with Ubuntu Open Week (a week of tutorial sessions), I ensure that each day there are only four or five hours of available sessions but that they are spread throughout the week during European afternoon hours. This ensures that the West Coast of the U.S. can get online early in the morning, while on the East Coast of the U.S. is midday, and in Asia and beyond it is later at night. This approach has helped our events to hit the most people within the worldwide spread of contributors that are involved in the Ubuntu community.

Another complexity in picking times is communicating the time. There are many ways of describing different times, such as Pacific Time, Central Time, Eastern Time, Greenwich Mean Time, and Central European time. Daylight saving variations make it even more complicated. Fortunately, there is a solution to this in the form of Coordinated Universal Time and its ludicrously jumbled acronym, UTC.

When everyone uses UTC, people can calculate their local time zone offset based upon the difference between UTC and their local zone. Although many people are still unaware of what UTC is, I highly recommend you use it: it is becoming a more common reference as more communities have online events. When using UTC, it is recommended you still add other time zones next to the statement, e.g., “our next meeting will take place at 9 p.m. UTC (10 p.m. London, 2 p.m. San Francisco, 5 p.m. New York)”.

NOTE

I highly recommend sending out a reminder two hours before the event via email, your website, Twitter, identi.ca, etc., to remind people that the meeting is happening soon. This will help remind people of the time and reduce time zone confusion.

Online Discussion Meetings

Communication within your community should always be your top priority. When the communication channels are open and conversation flows freely, your community gains momentum, progress is made, and your members will feel engaged.

Internal communication should seek to satisfy a range of needs that are involved in running a successful community. These needs are oriented around communicating progress on your goals, identifying direction, regular social connections, and day-to-day discussion.

Meetings are a key component in ensuring your community is running effectively. They provide a number of opportunities:

Discuss progress

With your strategic plan and road map in place, you can use meetings as an opportunity to communicate progress on Objectives, Goals, and Actions.

Discuss solutions

Meetings are a chance to discuss the implementation of solutions and any problems that your community may have.

Assign tasks

Meetings are useful for identifying what needs to be done and getting people to volunteer to work on specific tasks.

Resolve conflict

Every community has conflict, and yours will be no different. Meetings are a useful place to air issues and resolve personal conflict. We discussed conflict resolution in detail in [Chapter 9](#).

Within every community, a primary medium for regular communication should exist. Every community member should know they can come together with other members at an agreed place and at a regular interval to discuss and debate the issues that are before the community. This is the function of a regular meeting.

The first step in getting regular meetings up and running is to choose a location. There are various options available, such as IRC, conference calls, and more. Personally I would recommend IRC, as these meetings will be open and relatively easy to access, and the conversations can be logged. You should ensure that wherever you choose to hold your meeting, the meeting is open to all and the tools required to join the meeting are freely available.

Choosing a time

The next step is choosing a regularity and a time. Regularity is how many times you want the meeting to occur. This could be weekly, biweekly, monthly, or otherwise. Whatever you decide, choose that regularity and stick to it. Ensure that your regularity is bound to a day. As an example, your meeting could be “the first Monday of the month.” Regularity is largely dependent on your community and how much discussion needs to occur. I would recommend a minimum of one meeting a month and to increase the regularity if required.

Choosing a time is a complex task for international online communities. You should look at your primary set of contributors and where they are based, and ensure you have your core

contributors at meetings. Try to pick times that average out as suitable for the majority of contributors. This may involve some early mornings or late nights for some people to attend. If you have a truly global spread of contributors, you may also want to alternate meeting times so that the early mornings and late nights don't bite the same contributors every time. Some contributors will simply never be able to make the meetings due to the times. You should ensure these people can access meeting notes or a log of the discussion.

Advertising the meeting

With these details in place, you should advertise it clearly to your community. The most common place to do this is on a website. Make sure you have all the details clearly available. You should also specify the time zone of the meeting. A useful tip here is to use the UTC time zone, which is internationally recognized.

Here is an example of how you can make your meeting available:

MyProject Monthly Meeting

The MyProject Monthly Meeting is an opportunity for our community to come together to discuss progress, technical issues, problems, and other issues relevant to MyProject.

Everyone is welcome to attend.

WHEN: First Monday of every month

TIME: 20.00 UTC – 21.00 UTC

WHERE: #myproject IRC channel on irc.freenode.net

With the meeting text ready, it is time to ensure your community knows that the meeting exists. You should publicize your meeting in the places where your existing community and prospective community are likely to find it.

The first and most obvious place is your website. Ensure that your meeting is prominently located: it should not be buried away behind some obscure menu options. You should have a link to the meeting on the front page of your website, and it should be common knowledge where to find the page.

You should ensure that the URL to the meeting page does not change. This is particularly important if the meeting time and date is changing and if you are using the page as a place to put the agenda.

You also should announce and publicize the meeting in your community's primary communication channels. This could include your mailing list, in the topic of your IRC channel, and on blogs. The whole point of the meeting is to get your community together, so you should make every effort to ensure the community knows about it.

When the meeting is complete, you should also publicize the results. Many of your members will be keen to see what was discussed and what the outcomes were. Publicizing meeting results also indicates to your members that important things take place there. This will help them decide whether they want to join the next meeting.

NOTE

With guerrilla marketing, not only are you marketing something, but you also encourage your community to market the same thing in their own way. An example of this is asking your community to do the marketing on their websites or in the signature of their emails.

This is a useful technique for publicizing important community features and events such as meetings. You should encourage your community to share in getting the word out there.

Setting the agenda

Every regular meeting should have an agenda set. This can be set by those who organize the meeting, but a preferred approach is to allow your community to submit agenda items. This ensures that everyone is welcome to use the meeting as an opportunity to raise issues.

The method of soliciting agenda items is really up to you, but I would recommend that people add their items to an existing agenda so everyone can see the full agenda. This avoids duplication of agenda topics. A good method of doing this, and one used in the Ubuntu community, is to use a wiki. Put your meeting time and details on the wiki, and use the wiki page as a place for the community to add agenda items.

Running the meeting

It is essential that your meeting is run well. There are few things more frustrating than a well-organized community meeting with an agenda that fails to reach any conclusions, fails to keep to time, and fails to cover all the topics in the agenda. Your goal is to keep the meeting moving along, ensure everyone gets their chance to speak, and ensure that outcomes are generated. The most important aim for any meeting is to generate *output*. At the end of your meeting, you want to ensure that there is something to show for it.

Now I want to share some best practices for chairing these kinds of meetings that will help you get the most out of your own meeting time:

Explain the structure

Kick off the meeting and issue a general welcome. Make it clear at the beginning of the meeting that everyone is welcome to contribute to the discussion. Next, explain how the meeting works: a series of agenda items will be raised and discussed, and then action items will be generated for each topic.

Keep your eye on the clock

Always keep a keen eye on the clock. If the meeting has a fixed length (usually an hour), keep to that time. This will mean determining how much time each agenda item has for discussion. This may also mean stopping the discussion on an agenda item to move onto the next topic.

Ensure speaking equality

Some people in your meeting will dominate the discussion, and some will be more reluctant to chip in. To ensure that everyone gets a say, you may need to prompt the quieter people and ask if they have any comments. Naturally you should do this only if they are likely to have a comment: don't just pick on people because they haven't said much.

Focus on outcomes

Always do your best to keep the discussion focused on finding outcomes and actions for moving forward. Every meeting has the potential to turn into a long discussion with no outcome or next steps. Always keep your mind focused on what needs to happen next to move the topic or goal along to the next stage.

There are pages and pages of content written in books about how to chair meetings well. Doing it effectively is very much a learning process, and it will take time for you to find your feet. I highly recommend sitting in on other, longer-established community meetings to learn how those chairs keep their meetings ticking along.

Organizing Online Tutorials

When I first joined Canonical as the Ubuntu Community Manager, I set myself a career-long goal of trying to understand how to bridge the gap between a user and contributor.

Fundamentally, there is no difference between a user and contributor. Both are big bags of flesh and bones, but some people manage to get up and running as a contributor and some don't.

A key driver in my mind was education. While we had oodles of documentation, people really learn when they sit in the same room, sharing a computer and pointer, gesturing, and sharing knowledge. Although I could not get our entire community in the same room, I was keen to get as close to that educational nirvana as possible.

With this in mind, I developed the concept of the Ubuntu Open Week: a week of IRC tutorial sessions that teach attendees how to do something. The week includes sessions for a wide variety of hands-on topics such as how to file a bug, how to triage bugs, how to create patches, how to translate the messages displayed by applications, and more. Each session is delivered by a competent community member, and most of the sessions have included upwards of 250 attendees. I am going to use my experiences with the Ubuntu Open Week as a basis to explain how to run your own online tuition sessions.

Scheduling

To create the Ubuntu Open Week, we first decided on when it should occur and how many days it should last. Traditionally I have run the event from Monday to Saturday so that those who cannot attend mid-week can join for at least one day. Each day usually has between three

and five sessions. With this schedule I knew how many session leaders I needed to find. It is up to you and your community to determine what kinds of sessions are scheduled.

Announce the completed schedule primarily within your internal community communication channels, such as mailing lists, chat channels, and on your website. You may also want to consider doing some external publicity to encourage new contributors.

Preparing for a session

Before you kick off your tutorial sessions, it is recommended that you ask your session leaders to prepare for the session. Even though each session lasts only an hour, that can feel like an awful lot of typing in the thick of things. As such, it is recommended that your session leaders prepare a script for a reasonable chunk of the session, with each item of discussion on a separate line. This script could look a little like this:

```
hi everyone!  
welcome to my session about how to apply as an Ubuntu developer  
in this session we are going to discuss the process of how you prepare, document and apply  
for approval as an Ubuntu developer  
...
```

Grammar fans may have noticed that the script lacks uppercase letters and punctuation. IRC as a medium tends to lack these attributes, and as such, looks more free-flowing and conversational. You want your session leaders to sound this way, and so it is recommended they deliberately keep the script in lowercase letters.

Running a session

The way we have typically run the sessions in the Ubuntu Open Week is to have two IRC channels open at any one time:

#ubuntu-classroom

This is the channel where the session leader delivers the tutorial content. It is expected that attendees do not speak in this channel while the session is in progress. If you do get excessive chatter, you may need to ask a channel administrator to mute the attendees.

#ubuntu-classroom-chat

This channel is where people can discuss the session while it progresses and ask questions.

Questions are an important part of a tuition session, but questions could easily get lost in the flurry of discussion in the *#ubuntu-classroom-chat* channel. So we defined and published a convention for people to ask questions. The attendee simply prefixes a question with "QUESTION." As an example:

```
QUESTION: How do you attach a patch to the sponsorship queue?
```

This convention makes it simple for questions to leap out within the mass of discussion in the channel.

NOTE

One of the excellent attributes that makes IRC so suitable for tutorial sessions is that the session can be logged easily. You should ensure every session is logged and put somewhere on your website for people to access.

Session logs are a great source of content. Many of the questions that are featured in the sessions could be source material for an FAQ, or the full tuition session log could be expanded into a full HOWTO document. These are excellent tasks that some members in your community might be interested in performing.

Event-specific notes

Medium

IRC is the recommended medium for this type of event. IRC allows many people to join, is ideal for concurrently running a session and also allowing people to discuss it, and allows the session to be logged.

Date/time

There are no additional considerations beyond the general notes that were discussed earlier.

Summary

In this chapter we have explored many of the fundamental attributes involved in putting together an event. I not only explained the importance of events in the wider scheme of community, but also explored the different types of event available, discussed the core elements involved, and provided a few examples of some events that you could organize.

The topic of event organization is huge. Countless books, papers, and courses have been created for it. As such, we have only scratched the surface of the science of running great events, and there is still much to learn. What this chapter has provided, though, is a firm foundation that will get you up and running to produce some useful and effective events for your community. Use this chapter as a starting block, run some events, learn from other people, learn from yourself, and continue to refine and perfect your events.

Every event needs at least two or three tries before it really finds its feet. This has been the case with every event that I have been involved in organizing. It takes time to really understand the event, the needs of your attendees, the organizational requirements, and the amount of time required to get everything ready.

Events can be the glue that connects your community together with its core ethos and reinvigorates and reinforces the reason why people are involved. Get it right, and you and your community have so much to gain. Good luck!

Hiring a Community Manager

“I like work: it fascinates me. I can sit and look at it for hours.”

—*Jerome K. Jerome*

LIKE MANY FOLKS, I HAVE HAD THE GOOD FORTUNE and opportunity to do my fair share of travel to conferences as part of my work. As a result, I have been able to experience many of the conferences I once dreamed of attending, but there were still some I never quite had a chance to get to.

One such show was Ohio Linux Fest, a mid-sized community-run conference devoted to all things Linux and open source. I had always heard great things about the show from a member of my team and members of the community, but for some reason the Fates always conspired against me and I was busy every time the show was scheduled: I was either traveling, in meetings, on vacation, or otherwise unable to get out to Ohio.

Back in early 2008, though, I received an email about the show inviting me to speak. Knowing full well that something was going to get in the way at some point in the future, I slam-dunked the dates into my calendar with a little note next to them:

MAKE IT HAPPEN THIS TIME.

I responded to the email expressing interest, and the organizers kindly offered me the keynote speech. Somewhat flattered at the invitation, I happily accepted.

As time meandered on toward the show, I started working on the keynote and created a presentation that would eventually become the foundation for writing *The Art of Community*. It was that talk specifically that inspired much of the thinking behind the first two chapters of this book. I wrote my slides, carefully tweaking each word, and my presentation, rehearsing my endless stream of (often awful) jokes and ensuring that everything was as shiny and buffed as it could be.

The Fates cooperated this time, and when I got to Ohio and to the conference, I was instantly deluged with meetings, discussions, and other work that I needed to tend to. While hugely productive and useful, there was also a negative aspect to this barrage of activity: I didn't get a chance to see and mentally prepare for the room I was to deliver my keynote in. I knew the talk was going to pull a decent audience, but I didn't fully understand the sheer size of the room until I arrived, 20 minutes before my speech. It was enormous, and while speaking was by no means new to me, this was a new level. It was the first time I had been nervous about a speech in a long time.

Fortunately, the keynote went well and was well received by the audience. As I packed up my laptop and chatted with some of the audience members, one guy stood there quietly and waited for me to finish my conversations. As everyone left, he walked over and said:

Nice to meet you, Mr. Bacon, thanks for the speech, but I really fucking need your help.

Somewhat stunned at the presence of an f-bomb in the opening line of a new conversation, I let the guy continue to explain that he had been tasked with finding a community manager for his company (a small independent software vendor), but was struggling. He had flown here from Canada desperate for help. Although the seriousness and extent of this poor guy's panic was rare, the confusion surrounding community management and how it fits into an organization is not.

It is becoming increasingly common for an organization that either has a community or is seeking to build one to hire someone to lead this work. Unfortunately, in many cases they have no idea of what specifics they want or who to look for. Many look to existing community managers for advice and guidance.

This book has focused on teaching a prospective community manager and community leader how to build and inspire a strong and productive community. However, this book has not explored how to bring such a role into an organization and help get that person up and running easily. This in itself presents its own set of challenges and opportunities, and these are the focus of this final chapter.

Although this chapter may initially seem to be the stuff of managers tasked with hiring staff, it can also be of real value to (a) existing community managers who want to build a team, and (b) community managers who are looking to understand the needs and expectations of their communities.

Why Community Building Has Become a Big Business

The subject of community has become quite the buzzword in many business circles, particularly in IT. Its presence is felt more and more at conferences, in papers, on blogs, and across the current global sensation that is Twitter. Irrespective of the medium, this explosion of interest in community has happened for three closely interlinked reasons.

Community is implicitly a positive word. It speaks of openness, participation, awareness, and an agreeable intention to engage in an environment driven by merit as well as caring for others. For open source companies, this is powerful inferred meaning that speaks well to their audience. As such, it makes entire sense for a company to light up their website like a Christmas tree with references to “community.”

Second, for those interested in open source, community has become synonymous with “engagement in the open source space.” Open source companies are fully aware that if they don’t have an answer for their community relations strategy, they simply won’t be taken seriously by a significant demographic of people. Whereas five years ago this demographic of people was often seen as strange, hygienically challenged, bespectacled nerds who lived in their mother’s basement adorned with *Buffy the Vampire Slayer* posters, it is now well known that those with buying capacity and/or influence are placing importance in the community attributes of open source. These are *real* customers who have developed this value expectation due to the constantly reinforced open source mantra of participation, community, and technical quality. When the industry cradles open source and its associated values, the big cats in the ecosystem need to adjust to reflect that.

Finally, irksome economic times have resulted in very real consequences for small businesses. Executives have been forced to reassess how they can achieve their goals and ambitions with a more painful awareness of the bottom line. Multiple marketing and engineering people can be expensive—a lot more expensive than a community manager.

The accumulation of reasons for building community in a business environment has created a strong commercial justification for community and for those who can build it, along with a set of expectations around what these community builders can deliver.

AVOIDING THE HYPE

A problem befalls those who (a) seek to understand the scope of community, and (b) those who communicate their expectations and experiences of building community.

In every industry, certain words that once had reasonably obvious illustrative attributes and consequences have subsequently become colloquial references. We have seen this extensively with trademarks: Aspirin, the Hoover, Cellophane, Thermos, and even Heroin were all once trademarked to specific companies (Bayer, Hoover Company, DuPont, Thermos GmbH, and Friedrich Bayer & Co., respectively). Using Hoover as an example, in England many people will refer to any brand of vacuum

cleaner as a “hoover.” At one point in time, though, “hoover” pointed to a very specific representation of focus, quality, and expectation in a vacuum cleaner, embodied in products from the Hoover Company. Since then, the trademark has been somewhat genericized in different parts of the world, and what some refer to as a “hoover” will often bear little resemblance to the virtues of Hoover Company vacuum cleaners.

Community managers face similar risks built on misguided expectations. With all of the buzz, focus, and excitement around community management flowing, some more public representatives of the role have been a little guilty of stepping over, hiding, and downplaying the very real day-to-day focus of this work in favor of academically pleasing social science. When the high-level representation of community managers overplays the very real “on the ground” focus of the role, the balance becomes upset and there is a risk of genericizing community management as “the theory of working with groups of similar interests” as opposed to connecting the term firmly with hands-on best practice in building real communities that do real, measurable work.

As you are looking to fill your community manager role, be aware of this hype and how it affects candidates. To avoid it, look for substance in the day-to-day interactions, achievements, and efforts in the candidate’s work. If there is an overwhelming level of theory and social science in response to direct questions about this day-to-day work, alarm bells should trigger.

The Role of a Community Manager in the Corporation

In recent years the tubes have been ablaze with chatter over where community management should fit into an organization. Is *marketing* or *engineering* an apt destination for the reporting line? Do we expect our community managers and representatives to report to the Director of Marketing or the Chief Technical Officer? More specifically, when you bring a community manager into your organization, which of these two teams do you feel can most effectively support and enable a community builder to actually build a great community?

I firmly believe that community management is a tale with both marketing and engineering story lines flowing through it. If one is missing, community can feel unbalanced, misrepresented, and ineffective. We should always seek to celebrate and market the opportunities and importance of community, but that means nothing if you are not willing to roll up your sleeves and build and reinforce the collaborative technical groundwork in your community.

Ultimately, your community manager should be well versed in the mechanics and technical/social foundations of collaboration in open source communities and should be able to strategically structure and execute objectives that enable your community on the ground to do great work. You should ensure your community manager has a close connection to your technical leaders, but also a close connection with your marketing department to help them articulate and express your community story.

Setting Expectations

Before you put pen to paper on a job description for your new community manager, it is important to get some expectations straight around the role. Community management is fundamentally complicated, and many organizations with mismanaged expectations have made some very costly mistakes in finding someone to effectively build their community. Unlike positions with a longer history in companies, where roles and expectations are clear, community management embodies a lot of ambiguity and can confound hiring managers.

Scope of the Role

Busy hiring managers who already have many other roles to fill often fail to recognize the importance of understanding a community, so their job description contains unreasonable expectations around the role. A mismatched job description means that when a candidate is found, he resultantly has something of a bumpy road in getting a grip on the role. If you are lucky and get a hardy candidate who is willing to take the knocks while the role is being fleshed out, you may come out all right in the end, but the adjustment is stressful for most candidates and wasteful for the company.

To understand the scope of the role, you need to first understand (a) what the scope of the community is, and (b) what your expectations are for someone to work with that community to enable and extend it. Let's break these two separate points into a series of questions that you should answer before you craft the job description, beginning with the scope of the community itself:

- Who is in your community?
- How big is it?
- What kinds of skills and diversity are present in your community?
- How does the community interact and work with your company?
- What kind of governance infrastructure is in place?
- Who are the contentious people, and what are the contentious topics?

To determine the expectations around managing that community, we can approach the previous set of questions but assess where we see them trending in the future:

- How do you want to better understand who your community is?
- How would you like to grow the skills and diversity in your community? What are the primary skills and roles that you would like to focus on?
- How would you like to change, improve, and otherwise focus on how your community works with the company?
- What new and improved governance is required, and where should you focus your efforts first?

- How would you like to resolve and improve relations with the contentious people and topics in the community?

When you have answers to all of these questions, you should have a set of broad, high-level goals and ambitions that you would like to see your community manager focus on. You should now run these goals and ambitions past some other members of your organization to ensure they are doable for a community manager candidate. As you receive feedback, you can refine this set of goals and expectations.

When you have hired your manager, you can use this set of goals and expectations to build a full strategy for the manager. We will discuss this a little later in this chapter.

Risk

One element you should be entirely clear about is the risk associated with bringing a community manager on board. Fundamentally, what you are doing is hiring someone who will become a public face and representative of not only the community, but also of the company. If you get the right person with a strong commitment to both, you have the potential for a long, productive, and lasting relationship. If you get the wrong person, your difficulties may be embarrassingly exposed to the world.

The primary risk here is that when someone represents your community, she acts as a delicate middleperson between the organization and the community. This person plays a careful role in balancing expectations and making it clear that she will represent the best interests of each to the other.

With this responsibility lies the risk of the community manager getting frustrated with both the organization and the community. If the agitation is sourced from the organization, the manager may decide to leave and potentially defame the organization to the community as revenge. On the other hand, if the source of the frustration is from the community, the manager may get burned out and generate problems internally that ultimately reflect back on both the community and the organization. We discussed the topic of burnout earlier in [Chapter 9](#).

In summary, you need to be extremely careful about whom you hire into your community management role. You and your community manager are going to need to be able to understand these risks and be able to handle them in a professional and upstanding way. You need to find someone whom you implicitly trust and who is not going to put a personal agenda before the community.

Breaking Tradition

Another expectation to adopt is that community managers may well need to step outside the traditional boundaries of the business world. For a community manager to really build a rapport with the community, he needs to fundamentally be a member of that community and

exhibit the culture of that community. That in itself can ruffle some feathers in the organization.

Much of this is based around seemingly unimportant elements that actually play a key role in community management. As an example, most volunteer communities are very casual places. People wear casual clothes, and you should expect your community manager to do this as well. You should expect your community manager to be vocal, loud, and opinionated. It is this over-the-top personality that will pump the community up and get them excited. You should expect exciting and vivacious presentations at conferences, laptops covered in stickers, amusing and conversational interviews, and other elements. While I am not suggesting that you should demand these attributes from your community manager, don't be surprised if you get them.

In my own experience, I have brought a huge amount of my personality to my work. My communities know of my wacky sense of humor, my love of heavy metal, my loud and gag-laden presentations, and my up-front and frank discussions with community members. I am proud of this approach to my work, and other communities are likely to be the same. You should ensure your hires are free enough to exercise individuality in this role.

Despite this, of course there, is a line. When you have the opportunity for vivacious and excitable representation in the role, there is an increased possibility of your community manager putting a foot down wrong, offending someone, stepping out of line, or otherwise embarrassing herself, the community, and the company.

Expect it. Every community manager steps over the line at some point, and you should expect it to happen within her first year on the job. When it does happen, use it as an opportunity to sit down with her and have a discussion about where the line is drawn and how to avoid problems in the future. A frank and open discussion will not only make it clear that the behavior was unacceptable but also ensure that your relationship with the community manager is built on a foundation of openness and frank discussion.

Control and Reporting

The final clear expectation you should have in your mind is where your community manager sits in the company. Earlier we talked about whether the role fulfills more of a marketing or engineering approach, and you should consider this carefully.

To make this decision, look back at the questions we discussed when talking about the scope of the role earlier in this chapter, and determine whether your community manager is going to be focusing more on your message about the community or the collaborative infrastructure, processes, and growth of the community. In other words, are you expecting your community manager to help talk about the community and company at conferences and to customers and journalists, or to actually help grow, build, refine, and optimize the nuts and bolts in the contributor community?

It is likely that the answer to this question will be “both,” but consider which is the prevalent focus. As an example, if your community manager is primarily looking after a contributor community that produces an open source product, I would heavily recommend that she reports to an engineering manager. If, however, you primarily expect the community manager to fly around the world to talk about and represent the community at conferences and tradeshows and to speak to the press, I would recommend two things. First, have him report to your marketing department, and second, change his title from community manager to something else: he is representing the community, not managing it, and many people will expect the latter and be disappointed.

The ability to enact change

Hot on the heels of which part of the organization the community manager reports to is how much control and opportunity this person has to enact change. This is a complex but unfortunately common problem with many who have hired community managers: they succeed in bringing information about the community into the company, but lack the power and contacts to implement any of the change they recommend.

As an example, imagine your community manager recommends a series of improvements to an element of your community’s technical workflow. These improvements may require technical changes or refinements in other parts of the organization, such as a specific engineering resource. If the community manager rolls up to that person’s cubicle and plonks down a big list of required changes, it is unlikely those changes will be implemented anytime soon. That engineer has her own big ol’ list of work to focus on, and any additions, such as these proposed refinements, are likely to be gently shaken down to the bottom of the TODO list.

To mitigate this blockage, you should ensure that your community manager has a degree of control over changes and refinements to parts of the organization upon which the community’s work depends. While I am not expecting you to have engineering teams report to a community manager, make sure engineering managers make community changes and requirements a priority when gathering requirements in a new strategic plan or cycle.

Another excellent approach that has worked for me is to form a small ad hoc team with approval from the managers. This small team works together to solve problems that involve multiple teams in the organization. As an example, if you have an engineering problem that involves a source control system and a website, have one person who represents the source control system, one who represents the website, and your community manager. Form an agreement to work together on combined problems and get approval from the manager to regularly discuss the prioritization of these problems.

The Responsibilities of Community Engagement

Another important attribute to consider is how your organization expects to interact with the community after a community manager comes on board. In many, many organizations, once

they proudly hire someone into an explicit community management role, other people in the organization assume that the community manager will carry out interaction with the community on behalf of the teams. As an example, if a software company has a department that builds a particular product, it may be assumed that any and all community growth and interaction with regard to that product will be performed by the community manager.

You need to determine whether your community manager will indeed perform this function in the organization or whether he will instead act as more of a consultant to help other people in the company work directly with the community. Whatever the decision, it needs to be clearly communicated across the organization and the community.

My recommendation is that your community manager should perform direct interaction with the community for some teams but act as a consultant for most community-facing teams. This way you will have a high quality of interaction in key places where it matters most, and can build a culture of community interaction in your organization. This approach is also far more scalable: having a single person act as the community representative will always end up blocking on that person.

Salary

Salary is a hugely complicated topic when it comes to hiring a community manager. The reason is fairly simple: the role is so new and unknown for many organizations that it is difficult to gather fair expectations around salaries and a market rate.

Unfortunately, many of the common resources HR departments tend to use to gather data provide mismatched and unrepresentative information to work from. As an example, you can do a search for “community manager” and find an average salary in many recruitment and job database websites, but in almost every case the search returns entirely unusable data.

The reason for this is that the term “community manager” points to all manner of different types of responsibilities in all manner of different types of organizations. I know “community managers” who work for tiny nonprofit charities and organizations and who look after a community of less than 30 people. I know of others who look after communities of hundreds of thousands of people with complex collaborative frameworks, technical workflow, and significant public focus and responsibility. These are not the same types of role and should not demand the same salary.

Another consideration is the “manager” part of “community manager.” Many community managers don’t actually manage anyone: the “manager” in their title points to their influence over the community. As an example, in my current role I manage a team of community managers who work on many different aspects of a community. So while part of my role firmly involves influencing a large and thriving community, I also have the formalized, traditional management responsibilities of a team inside an organization. This kind of management role should be reflected in the salary decision.

Unfortunately, I can offer limited advice on a suitable salary for your community management role. The decision requires a firm awareness of the market (the supply and demand) and deriving a figure at that point. I have previously advised companies about salaries, so feel free to contact me if you are unsure. Otherwise, take the pulse of your community as well as the wider marketplace, come to your own figure, and run it past a few other hiring managers who have hired community managers. This should help you find a compelling and fair figure for both your company and potential candidates.

Communicating Expectations to the Candidate

After resolving in your own organization the expectations we have covered over the last few pages, you should make them clear to potential candidates. Some of them should be expressed up front in the hiring process and as part of the interviews: the primary expectations around the role, what they will be working on, and what they will mainly spend their days doing. These expectations are the meat and potatoes of what will constitute the role.

There are some other, more subtle expectations that may be involved in the role, though, and you should make these clear also. Many are rife with potential for causing trouble if they are not made clear from the get-go. So take a look at the following list of expectations and see whether they apply to the community management role that you are looking to fill. If so, make them clear to candidates in the interview process:

Adjustments to the role

The role of community manager in your organization will likely adjust, grow, and change on two levels. First, the expectations around how your community is managed will change as you learn more about the community, as your organization changes its focus, and as your company grows or shrinks. Second, the specific areas of focus and strategy that are covered in each strategic document will undoubtedly change. Your community manager needs to be flexible in both of these areas.

Unusual hours

Many communities span multiple countries, connected by the Internet. With this in mind, your community manager may have to keep some unusual hours. This is common when scheduling online meetings and events: to ensure that a chunk of the world can attend the meeting, it can often mean your community manager getting out of bed in the middle of the night to attend. You should ensure she is willing to do this.

Travel

Many community roles involve extensive travel to conferences, sprints, and other events. Although many jump at this opportunity, others have families and partners to consider. Also remember that circumstances change: people get married, give birth, and want to get off the road. You should expect that while your candidate is happy to travel right now, that's likely to change in the future.

Public attention

Many community managers become public representatives of the community and the company, and you should make it clear if this is an expectation. If it is, the candidate should expect to have to answer tough questions sometimes in interviews, on panels, and elsewhere. If this is the case, you should make clear to him that he will be expected to fulfill this role, but you should also support him extensively in handling these issues.

Conflict resolution

Many community management roles involve dealing with conflict and contentious situations. Some people are uncomfortable with these expectations, and you should clarify them up-front. More details on the specifics of conflict can be found in [Chapter 9](#).

Technical knowledge

Some communities are very technical places. These communities often require a community manager with extensive technical experience in the workflow, infrastructure, and processes of a technical collaborative community. You should make these requirements clear to your candidate and set an expectation that these technical requirements are likely to increase as your community evolves, changes, and grows. The candidate will be expected to learn these new concepts and skills.

Presentation skills

Many community managers get out on the conference circuit and perform presentations about their work. You should check whether they are happy to do this. You should also research whether they are actually any good at presenting: sometimes no presentation is better than a horribly delivered presentation. As such, you may want to pass on the presentation helmet to someone else.

You should get these issues on the table in the first interview. The expectations should be clear and up-front to avoid any uncertainty in the future, and you should preferably document them for future reference. Where you document them is up to you and your department: some may feel the contract is the most suitable place, whereas others may consider an addendum document for employment that sets clear but noncontractual requirements.

Managing Your Community Manager

When you have recruited someone to fill the community management role in your organization, the next step is to ensure you get that person settled into the role easily and effectively. While there are plenty of books available for hiring people and having them settle in, I want to share some hints and tips with specific regard to community managers.

When I have hired people to join my team and manage a community, I have broken down the process into three primary areas:

Induction

The first month is critical when someone joins your organization and, more specifically, your team. These are the topics involved in this very early part of the community manager's new employment in the organization.

Strategy

This is how you define and build a set of agreed-upon objectives so you and your community manager are entirely clear on where to focus her efforts.

Management and communications

These are the weekly operational elements involved in managing your community manager.

Let's now take each of these three areas for a spin and explore some of the attributes involved. Let me state again that I am focusing only on elements that specifically apply to community managers. You should augment these words with the general best practices used when new employees enter your organization.

NOTE

As we work through each of these three areas, I will be presenting the information from the perspective of the manager of a community manager. As such, much of this information will be of primary use either to readers who need to hire and manage a community manager or to community managers who are hiring a team. As I mentioned at the beginning of this chapter, though, if you are a current or prospective community manager, this content can help illustrate some of the expectations that your manager will have of you.

Induction

When your community manager first joins your organization, you will have to cover the usual set of induction-related items: setting up email addresses, accessing company resources, creating calendars, and other such activities. Although these are important, they are not of interest to us here. More important is the way your community manager exploits the induction period to define his reputation in the organization and the community.

Internal reputation

Reputation is critical for all community management. From the perspective of the company, a community manager is seen as an important entity who has knowledge, focus, and awareness of not only how the community members think but also how they work. Remember: many people simply don't understand how community works, and the community manager acts as a translation layer to help other parts of the organization understand how this seemingly random entity functions.

With this in mind, you should carefully consider how your community manager builds his reputation in the organization. This is important for a few reasons. Your community manager is going to need to know how to work with and interact with different parts of the organization, and vice versa. He will be faced with many different challenges and queries within the community, and many of these topics will need to be forwarded and integrated into other parts of the organization.

Another important reason to build these bridges is to help your organization understand what a community manager does. With the role still very new and unknown in the minds of many, some people will either not know what the purpose of the role is or misconstrue or simply hold a cynical idea of it (“Oh, community management—that’s just hanging out with nerds and writing blog entries” is one verdict I once heard!).

It is important to clarify to everyone what the role involves and build a solid reputation for your community manager. It helps to schedule a set of introduction meetings or calls with other departments in your organization and make sure to involve your community manager in wider discussion and topics where appropriate.

Community reputation

Your community manager’s reputation is critical with the community she is hired to work with. One of the very first tasks she should focus her efforts on is building a strong reputation with the community.

It is possible that this reputation already exists if you hired her out of an existing community. If not, you should ensure that she has the time and encouragement to get out there and get to know the community. This can involve a range of approaches:

Participation in communication channels

The community manager should participate in communication channels such as mailing lists, forums, and other resources. This should not just involve reading, but responding and participating in discussion, too.

Blogging

If the community is one that is likely to read blogs and online articles, your community manager should be actively writing about her work. This is an excellent platform on which to build a medium for writing about her efforts, where she is working, and what achievements the community has made.

Travel

If your community needs to secure a reputation with the wider community in which your specific community fits, you may want to send your community manager off to some key events and encourage her to make some presentations. Travel is also an excellent opportunity to network and get to know important members of the community well.

To help your community manager build her reputation, I recommend you ask her to read other parts of the book, most specifically [Chapter 6, *Building Buzz*](#).

BE WARY OF TOO MUCH TRAVEL

I have an important tip when it comes to travel, based on a mistake made with many new community managers.

The recipe is often the same when a new community manager starts. He joins, there is a little bit of press and hype, and then the organization sends them around the him to every conceivable conference in the interest of building his reputation.

Although this is excellent for building relations with the community and beyond, it is very costly, both in terms of time and money. Aside from the obvious monetary costs and the effect on personal life, conference travel is going to seriously disrupt your community manager's ability keep up with email, have calls, and hold meetings. Regular and consistent participation in these more mundane activities is critical to getting the basic work of the community done.

So don't make your community manager travel too much. His primary value is on the ground working with your community: don't compromise that in the interest of presenting at conferences.

Strategy

From the very start of your community manager's term of service with your organization, you should have a strong focus and strategic priorities. A strategic document that outlines your community manager's responsibilities, split into objectives, goals, and success criteria as we did in [Chapter 2](#), will communicate what your community manager should be working on and formalize your expectations.

A strategic plan is valuable to guide all your employees, of course, but it's particularly important for your community managers because there is simply so much scope in their work. A community is a huge, boiling pot of activity that can produce a potentially epic collection of information on which a community manager can focus. It is important, therefore, for you and your community manager to understand, prioritize, and focus on a selected set of objectives and goals.

When your community manager comes on board, therefore, show her the strategic plan you developed from [Chapter 2](#), agree on a set of objectives, break them down into goals, and flesh out the strategic document into tasks for the community manager. The strategic plan will also help you track progress on that work as it proceeds.

Management and Communications

As with any other staff member, you will need to manage and check in regularly with your community manager. Although many of the common approaches for managing staff can be easily applied to a community manager as well, I'll note here a few specific considerations.

These considerations are primarily related to how your community manager works with other teams inside your organization and how he brings those teams community input. We can highlight two primary areas of focus:

Weekly engagements

Each week, your community manager should work with different parts of your organization to determine what goals to work on and to quickly identify anything that stands in the way of those goals.

Community feedback

Your community manager is an excellent funnel through which the community can channel feedback and opinion back to the organization. You want to ensure that this is factored into your community manager's workflow.

Let's now take a look at both of these topics in more detail.

Weekly engagements

As soon as your community manager joins your organization, you should set up regular calls or meetings in which you cover the following:

Areas of focus

Discuss the current topic and areas where the community manager should focus her attention. Ask her where she feels she should be focusing her efforts, then help and advise her on how to achieve these goals.

Strategic review

I recommend that in each meeting you review your strategic document and the community manager's own goals that you drew from it, to track progress. This is an excellent opportunity to highlight any blocks or problems with progress and to ensure you are in the loop.

Opportunities

Always explicitly discuss new opportunities and areas where you may want to consider focusing your efforts, either now or in the next strategic time frame.

Problems and concerns

You should always ask your community manager what issues, problems, concerns, and blocking factors are causing difficulties for his work. In many cases, relatively simple issues will be blocking work and initiatives, and you might be able to help solve this.

Community problems and concerns

You should also check in to see whether there are any problem with initiatives and work in the community itself. Are members of the community blocking this work? Are there concerns and problems in the community for which you can offer help to your community manager?

In addition to your personal meetings, determine what other parts of the organization should communicate with the community manager on a weekly basis. You may want to organize a team call that bridges several parts of you organization. By making your community manager a member of a wider team, you can ensure the flow of information between your product engineering teams and the community manager.

Finally, you should advise your community manager on who else she should check in with on a weekly basis. This may include marketing, senior management, other specific parts of the organization, or people who are working on specific projects that involve the community.

Community feedback

When your community manager joins you, do your best to ensure that she is able to get feedback from the community to key members of your organization. Community managers are always seen as primary contact points in communities, and you should expect them to receive input, opinion, concerns, worries, and other useful feedback from the community. You want to ensure that this feedback gets delivered to the right ear.

The easiest method of doing this is to make sure your community manager knows where feedback should go. When a new community manager joins an organization (particularly a large one with a complex hierarchy), this can be a complicated topic. It could be useful to directly provide her with a list of contacts to get in touch with when different topics of feedback come up. In addition to this, you should always ensure she feels comfortable asking you to whom she should forward feedback if she is unsure.

Summary

Throughout this chapter we have explored many of the topics involved in hiring a community manager. We have covered how to set reasonable expectations, how to merge that role into your organization, and how to build a measurable strategy to keep you and your community manager on the same page. Even if you faithfully carry out these steps, you should still expect a certain amount of the unknown.

Community management is still a very new profession. As community management becomes a more common role in organizations, and as I hopefully put out further editions of *The Art of Community*, I look forward to solidifying this chapter with more and more advice. Fortunately, the information I have provided here should get you off to a great start. If you

temper this chapter with your own experience and what your community manager tells you, you will get everyone off on the right foot.

End of Part One

And this, my friends, brings us to the end of our journey. Way back in [Chapter 1](#) we started with a bird's-eye view of community, and we have gradually zoomed in closer and closer to look at the day-to-day details. Throughout our journey, we have talked through the major topics in building a strong community.

When I conceived the content for this book, I was keen to put together a solid foundation to get the new community manager, leader, or organizer up and running as quickly as possible. I wanted to cover a diverse range of topics without bogging you down with impractical details and academic hand-wavery. With these goals in mind, I am proud of the outcome: I think *The Art of Community* provides the springboard to help you all build great communities.

This is only the beginning, though. The French critic and poet John Valery once said, "A poem is never finished, only abandoned," and the same can be said of this book.

Community management, leadership, and organization is a new and young science. There is still a long road ahead and many things to learn on the way. As such, I see this first edition of *The Art of Community* as part one in an ongoing journey to document the art of building strong and effective communities. The content here has provided a snapshot of knowledge that is begging to be furthered, extended, and augmented as we explore the road ahead and discover more answers about this sometimes strange but always fascinating science.

To continue the journey, I absolutely need your help. Now that this book is released, I am going to start preparations for the second edition. From the very beginning of *The Art of Community* project, I have maintained a website for the book at <http://www.artofcommunityonline.org>, with updates, articles, and a place for you all to share your feedback. On that website you can post your ideas, and meet and mingle with others passionate about building great communities. The website is an excellent place to share your stories and to join our very own community.

In addition to this, I organize an annual event called the Community Leadership Summit (<http://www.communityleadershipsummit.com/>) that brings together people like you and me to discuss, debate, and share our journey in learning how to build community. I highly recommend that you check into the event and come and join us there.

So without further ado, it is time to close this book and go out there and build an incredible community. Good luck, and let me know how you all get on!

Numbers

5-A-Day bug fix, 110, 165, 169

A

accountability, 213
 Ad Hoc Peer Review, 107
 anonymity, 206
 Aq, 21
 areas of collaboration, 45
 AUDIENCE, 122

B

BarCamp, 321
 Basinger, Mike, 211
 belief, 11
 belonging, 4, 5, 302
 events, fostering with, 302
 bikeshedding, 78
 Block, Peter, 8
 blog wars, 181
 blogging engines, 158
 blogs, 180
 brainstorming, 52–55
 bug fixing, 201
 bug reporting, 135
 bug triage, 135
 bug-tracking, 134
 process example, 125
 bureaucracy, 92, 226
 avoiding, 99
 burnout, 294–299
 achieving work/life balance, 298
 addiction, 299
 detection and treatment, 297
 Burns, George, 218
 buzz, 146
 advocacy, 151
 announcements, 167
 announcing your community, 170

attracting contributors, 171
 building alliances, 175, 176
 professional press, 176
 building blocks, 148–153
 mission, 148
 buildup phase, 165
 buzz cycle, 163
 deadlines, 165
 honesty, 153
 inspired words, 150
 planning, 164
 reviewing, 169
 targets, 170
 Tony Blair's New Labour campaign, 149
 trust, gaining or losing, 152
 websites, 154–163
 blogging engines, 158
 conversation, 156
 fresh content, 156
 microblogging, 160
 syndicated feeds, 159
 Bytemark Hosting, 121

C

capacity, 28
 Change Congress, 170
 charismatic leaders, 222
 cloud, 131
 cloud toolchain facilities, 131
 codes of conduct, 40
 collaboration-driven ethos, 3
 communication, 7, 66–74
 channels, 67
 choices, 69
 clarity, 68
 communication fetishism, 70
 communications infrastructure, 129
 leading by example, 75–89
 bikeshedding, 78
 daily communication, 75

We'd like to hear your suggestions for improving our indexes. Send email to index@oreilly.com.

- experimentation, 79
 - netiquette, 77
 - longer writing, 80–89
 - mediums, 71
 - communities, 3
 - (see also councils; teams)
 - belonging
 - events, fostering with, 302
 - communication (see communication)
 - contributors, 104–109
 - assessing, 106
 - attracting, 104
 - councils versus teams, 226
 - designing, 43–52
 - division into teams, 57
 - measuring, 187
 - anonymity and privacy, 206
 - approach to tasks, 188
 - defining purpose, 189
 - hooks and data (see hooks and data)
 - using feedback, 189
 - mechanics of communication, 201
 - read-mostly communities, 34
 - TODO list
 - conducive environment, 125
 - write-centered communities, 35
 - community building, 23–30
 - as a business, 343
 - communication flow, 27
 - environment, 29
 - hype, 343
 - teams, 25
 - TODO list, 24, 57
 - Community Leadership Summit, 357
 - community managers, 13–19, 342
 - community reputation, 353
 - corporate role, 344
 - breaking tradition, 346
 - control and reporting, 347
 - responsibilities of engagement, 348
 - risks, 346
 - scope, 345
 - hiring process, 342
 - communicating expectations, 350
 - induction, 352
 - internal reputation, 352
 - job description and expectations, 345
 - management of, 351
 - communications, 355
 - setting strategy, 354
 - personality, 13, 17
 - salaries, 349
 - strategic traits, 13
 - theory versus action, 16
 - travel, 354
- conferences, 309
 - conflict, 268
 - allocation of responsibilities and, 277
 - avoiding, 276
 - barriers to input, 274
 - commercial sponsorship and, 274
 - injustice, 278
 - personality differences and, 271
 - physical violence, 279
 - poisonous people, 273
 - conflict resolution, 220, 270, 280–294
 - clarity, 284
 - facilitators, 281
 - maintaining objectivity, 282
 - Johnson and Johnson approach, 280, 285–294
 - calm and reassurance, 286
 - collect data and probe, 287
 - discuss, 290
 - document, 293
 - reflect and maintain, 294
 - keeping records, 284
 - openness, 283
 - positive attitude, maintaining, 283
 - cons (conventions), 309
 - Coordinated Universal Time (UTC), 333
 - councils, 226–245, 226
 - (see also communities)
 - codification of governance, 239
 - communication between councils, 264
 - council charters, 241
 - designing, 227–238
 - commercial sponsorship, 232
 - communication, 237
 - council responsibilities, 228
 - membership, 234
 - structure, 230
 - forming new councils, 243
 - nomination and election of members, 242
 - Cricket, 195
- ## D
- data-gathering APIs, 194
 - deadlines, 165
 - deep-level diversity, 39
 - delegated governance, 225
 - designing community, 43–52
 - dictatorial leadership, 222
 - direct invoicing, 329
 - discussion forums, 72
 - diversity, 28, 39
 - documentation
 - importance in conflict resolution, 284
 - of planning processes, 61
 - of processes, 112
 - of surveys, 198

dot family, 24
Drucker, Peter, 148

E

East Bay and Tri-Valley SPCA, 104
economies, 5
“elevator pitch”, 107
enlightened dictatorship, 223
Ettrich, Matthias, 224
events, 303

- accommodations, 311
- belonging, fostering with, 302
- benefits, 303
- catering, 314
- common attributes, 309–315
- cost, 313
- date and time, 312
- equipment requirements, 312
- insurance and unions, 315
- locations, 310
- online events (see online events)
 - common attributes, 330
 - date and time, 333
 - media, 331
- organization, 305–308
 - allocating time to, 307
 - bargain hunting, 312
 - deadlines, 306
 - finding help, 305
 - identifying requirements, 305
- registering attendance, 313
- sponsorship, 324–329
- sprints, 316
- summits, 317
- types of events, 308
- unconferences, 321
 - event-specific notes, 323

F

fables, 8
facilitators, 281
family, 302
feedback, gathering, 111
feedback, managing, 109
fixed software cycles, 49
FooCamp, 321
Ford, Henry, 38
forums, 72
Free Culture communities, 36
Free Software community, 10
Fry, Stephen, 35

G

Geiser, Ian Reinhart, 225

GNOME, 126
GNU General Public License, 222
Gobby, 141
Google Open Source Programs Office, 307
Google Summer of Code, 220
governance, 28, 213–221

- accountability, 213
- bureaucracy and, 226
- case in favor of, 215
- community councils (see councils)
- conflict resolution, 220
- delegated governance, 225
- dictatorial leadership, 222
- engaging the people, 218
- enlightened dictatorship, 223
- expansion, 260
- inspiration, 219
- leadership, 216
- subcouncils, 263
- TODO list, 221
- types of, 221
- Ubuntu example, 245–260
 - escalation of issues, 259
 - membership, 254–259
 - structure, 247–254

Graen, George B., 39
Gwibber, 162

H

Hawthorn, Leslie, 307
hooks and data, 190–205

- gathering perceptions, 203
- mechanics, measuring, 201
- observational tests, 198
- statistics, 192
 - graphing, 193
 - interpretation, 193
- surveys, 195

Hudson, Paul, 177

I

identi.ca, 141, 160
imagination and opportunity, 11
IRC (Internet Relay Chat), 73, 331

J

Johnson and Johnson approach to conflict resolution, 280, 285–294
Jokosher, 23
JonoEdit, 22

K

KDE project, 224

KHTML, 224
Kroc, Ray, 91
Kubuntu Ubiquity usability test, 200

L

Langridge, Stuart, 21
Launchpad, 127
 hooks and data, 191
Lessig, Lawrence, 170
Linux, 10
Linux Demo Day, 145
Linux Format magazine, 177
LoCo Teams, 230
LugRadio, 3, 18
 LugRadio Live, 11
 recording software development project, 21

M

mailing lists, 71
 top versus bottom posting, 77
making it suck, 55
markets, 5
mechanics of collaboration, 124, 125
mechanics of communication, 201
media relations, 176
 amateur press, 179
 blogs, 180
 podcasts, 182
 professional press, 176
 video, 183
Mentor Summit, 320
meritocracies, 37
Messina, Chris, 321
microblogging, 160
mindshare, 146
 school meal project, 147
mission, 44, 148
 mission statements, 47
Miyamoto, Shigeru, 54
mrben, 4

N

netiquette, 77
Nielsen, Jakob, 167
Nintendo Wii games, 54

O

OBJECTIVE, 122
objectivity, 282
observational tests, 198
Ohio Linux Fest, 341
Oliver, Jamie, 147
online events, 329–339

online meetings, 333
 advertising, 335
 agendas, 336
 running online meetings, 336
 scheduling, 334
online tutorials, 337
 preparation, 338
 running sessions, 338
 scheduling, 337
open source, economic aspects, 5
OpenAdvantage, 152
openness, 45
opportunities, 45
OUTCOME, 122

P

Packard, Keith, 100
Pandy, Laszlo, 107
Paul, Celeste Lyn, 199
perceptions, 203
pitches, 327
Planet Ubuntu, 159
planning (see strategic planning)
podcasts, 182
poisonous people, 273
privacy, 207
processes, 94
 assessing contributors, 106
 attracting contributors, 104
 consequences of bad processes, 94
 designing, 96
 needs assessment, 101
 planning cycles, 102
 Ubuntu release process, 103
 documentation, 112
 feedback, 109
 getting buy-in, 112–114
 process design, 95
 avoiding bureaucracy, 99
 considerations, 98
 reassessment, 115
 scheduling, 116
 simplification, 92
 transparency, 100
product recalls, 94
programming languages, 120

Q

questioning assumptions, 53

R

Raymond, Eric, 134
read-mostly communities, 34
release cycles, 49

- release parties, 309
- required skills, 45
- resource fetishism, 133
- roles, 121
- Ross, Blake, 174

S

- scope, 33
- Second Life, 332
- SEO (Search Engine Optimization), 160
- sessions, 318
- Severed Fifth project, 170
- Shuttleworth, Mark, 212, 246
- social capital, 6
- social economies, 5
 - communication, 8
- software development
 - LugRadio audio recording software, 21
- source control, 136
- sponsorship, 324–329
 - finding sponsors, 326
 - handling money, 328
 - the pitch, 327
 - setting expectations, 326
 - sweat versus sponsorship, 325
 - understanding needs, 324
- Spread Firefox, 173
- sprints, 308, 316
- Stallman, Richard, 222
- strategic documents, 46
- strategic planning, 43–63, 48
 - documenting, 61
 - planning cycles, 102
 - structuring the plan, 50
- summits, 308, 317
 - event specific notes, 319
 - sessions, 318
 - structure and scheduling, 318
- surface-level diversity, 39
- surveys, 195
 - documentation, 198
 - reports, 197
 - targeting of questions, 196
 - variance in ratings, 196
- Sweet, Adam, 31
- syndicated feeds, 159

T

- tales, 8
- teams, 25, 31–43
 - belonging, process of, 32
 - collaboration among teams, 60
 - diversity, 39
 - division of communities into, 57

- effective communication, 60
 - Henry Ford's team-building strategies, 38
 - meritocracies, 37
 - read versus write communities, 34
 - team scope, 59
- thinking outside the box, 54
- Thorp, Ben, 4
- Tobe Community Council (TCC), 227
 - responsibilities, 230
 - rules for council membership, 236
- toolchain facilities, 131
 - resource fetishism, 133
- transparency, 45
- Trekkies, 34
- Troy, Ryan, 240
- Twitter, 141, 160

U

- Ubiquity usability test, 200
- Ubuntu Code of Conduct, 40
- Ubuntu community, 36
 - bug workflow, 125
 - governance example, 245
 - structure, 247
- LoCo Teams, 230
- openness, fostering of, 276
- release process, 103
- Ubuntu Community Council, 212
- Ubuntu Forums Council (FC), 212, 239
- Ubuntu Team Councils, 240
- Ubuntu Developer Summit (UDS), 317
- Ubuntu Global Bug Jam, 194
- Ubuntu Launchpad, 127
- Ubuntu Open Week, 337
- Ubuntu Upstream Report, 127
- unconferences, 321
- upstream applications, 36
- upstream projects, 126
- usability testing, 199
- UTC (Coordinated Universal Time), 333

V

- Veitch, Nick, 177
- Verduzco, Cristina, 104
- video, 183
- virtual worlds, 332
- Voice over IP (VoIP), 332
- VoIP (Voice over IP), 332

W

- Walli, Stephen, 5
- WebKit, 224
- welfare capitalism, 38
- Whedon, Joss, 35

- WineCamp, 322
- workflow, 120
 - assessment, 141
 - bug tracking, 134
 - bug-fixing workflow, 129
 - building workflow, 122
 - key themes, 128
 - communications, 139
 - maintaining transparency, 137
 - reporting, 140
 - source control, 136
 - technical considerations, 133
 - tool accessibility, 138
 - toolchain facilities, cloud versus self-hosted,
 - 131
 - Ubuntu bug workflow example, 125
- write-centered communities, 35
- writing, 80–89
 - achieving clarity, 82
 - cultural differences, 81
 - inspiring the community, 88
 - mechanics of good writing, 81
 - setting tone, 85
- “Writing for the Web”, 167

X

- X.org foundation, 101
- xFree86, 100
- Xubuntu, 46

ABOUT THE AUTHOR

Jono Bacon is an award-winning community manager, author, and consultant. He currently works at Canonical Ltd. as the community manager for Ubuntu, one of the largest open source projects in the world, with a diverse community of thousands of contributors. He is a well-respected and acknowledged leader in open source, community management, and best practices, and is the founder of the annual [Community Leadership Summit](#), an unconference that brings community managers and leaders together. He is a prolific writer, having written 4 books and more than 500 articles published across 15 magazines and online publications.

Bacon has also acted as an extensive consultant, and has worked with a range of organizations and as a senior open source consultant at OpenAdvantage, the award-winning UK government-funded service advising organizations, governmental institutions, and educational establishments on how they can use open source and build strong and vibrant communities.

COLOPHON

The cover image is a stock photo from Jupiter Images. The cover fonts are Akzidenz Grotesk and Orator. This book was authored on Ubuntu with OpenOffice.org, converted to DocBook XML 4.4, and typeset to PDF via XSL-FO. The text font is Adobe's Meridien; the heading font is ITC Bailey.