Extracted from:

Program Management for Open Source Projects

How to Guide Your Community-Driven, Open Source Project

This PDF file contains pages extracted from *Program Management for Open Source Projects*, published by the Pragmatic Bookshelf. For more information or to purchase a paperback or PDF copy, please visit http://www.pragprog.com.

Note: This extract contains some colored text (particularly in code listing). This is available only in online versions of the books. The printed versions are black and white. Pagination might vary between the online and printed versions; the content is otherwise identical.

The Pragmatic Bookshelf
Raleigh, North Carolina

The Pragmatic Programmers

Program Management for Open Source Projects

How to Guide Your Community-Driven, Open Source Project

Ben Cotton

Edited by Michael Swaine

Program Management for Open Source Projects

How to Guide Your Community-Driven, Open Source Project

Ben Cotton

The Pragmatic Bookshelf
Raleigh, North Carolina

Pragmatic Bookshelf

For our complete catalog of hands-on, practical, and Pragmatic content for software developers, please visit *https://pragprog.com*.

The team that produced this book includes:

CEO: Dave Rankin
COO: Janet Furlow
Managing Editor: Tammy Coron
Development Editor: Michael Swaine
Copy Editor: Corina Lebegioara
Indexing: Potomac Indexing, LLC
Layout: Gilson Graphics
Founders: Andy Hunt and Dave Thomas

For sales, volume licensing, and support, please contact *support@pragprog.com*.

For international rights, please contact *rights@pragprog.com*.

ISBN-13: 978-1-68050-924-3
Encoded using the finest acid-free high-entropy binary digits.
Book version: P1.0—July 2022

Everyone does program management, some just do it poorly.

➤ *Ben Cotton*

Introduction

This epigraph from Ben Cotton isn't simply a fun, self-congratulating laugh line I use in presentations. It represents a fundamental truth of complex technical undertakings: you can't avoid program management, so you might as well do it deliberately. Whether or not you realize it, you're already *doing* the work of program management. This book gives you the tools to do the work consciously and do it well.

What a Program Manager Does

As a program manager, you coordinate the efforts of the contributor community to meet the project's goals. The title "Chief Cat-Herding Officer" is a joke, but an apt one. You're working with a group largely or entirely composed of volunteers to produce software that other people will want to use. Individually, contributors have their own interests and goals. When you're a successful program manager, you get these individuals aligned in the same general direction to meet the overall goals of the community.

If you've seen the movie *Finding Nemo*, you may have a visual in mind (except with fish, not cats). Nemo is a fish separated from his father. In one scene, he's swimming with a school of tuna that gets scooped up in a fishing net. The tuna are swimming every which way as the net slowly hauls them in. But Nemo gets them all to swim downward. Once they're all working in the same direction, the fishing boat's motor can't haul the net in. The rope breaks and the fish all escape the net. This is what you're doing, except with lower stakes.

Some open source projects have a role explicitly named "program manager." Others have a "release manager" or similar title. Most projects have neither, but someone is still doing the work. It could be the project leader, a member of the technical steering committee, or an organized and enthusiastic contributor with no particular role. Wherever you are in the project's structure (or lack thereof), you're keeping everyone informed and helping folks find the information and resources they need.

This book covers all aspects of program management, from scheduling to feature management, bug tracking, and beyond. You might find as you read this book that you're sharing some of these duties with other members of the project. That's okay. The important part is that *someone* in your project is consciously working on each aspect. If you're not thinking about the work, that doesn't mean it's not happening. It means it's probably not happening well.

Why Program Management Is Important

Open source projects produce software, but they're run by people. People are dynamic and sometimes unpredictable. The more people are involved, the more unpredictable the project becomes. As a program manager, you bring order to the chaos. You don't do this by imposing order from the top down—that's not how open source projects work. Instead, you *give structure to the agreements of the community.*

When you're doing your job well, it can appear effortless. But I suspect that most projects that thrive in the long term aren't successful by accident. They don't keep everything working smoothly without effort. Instead, someone (or several people) does the work to coordinate effort across the project. And that can get complicated quickly. The number of communication channels in a team goes up exponentially as the team grows.[1] Communication overhead gets even heavier in open source projects where members of the team come and go, sometimes without warning. A successful program manager reduces the overhead by collecting and distributing information across the project.

In addition, an experienced program manager is an expert on processes. Some open source developers may think their project doesn't need any processes. But a process is simply the way you do things. You always have processes. The only question is whether the processes are predictable and serve the needs of the project or they're ad hoc and subject to the whims of whoever is doing work at the moment. Process development is a skill like software testing or documentation writing. But again, you can't impose it from above. As the program manager, you work as a consultant to the rest of the project, helping everyone come up with processes that best meet their needs.

Open source projects, like all human systems, are naturally chaotic. Deliberate program management gives the community, users, and sponsors reliable and predictable insight into the current state of the project and the trends that

[1] https://hbr.org/2009/05/why-teams-dont-work

hint at the future. It's much easier to do something well when you're doing it on purpose.

Why You Want This Book

Hopefully, you now have a clear picture of what program management is and why it's important to your project. This book will help you put this knowledge into action. Think of it as a do-it-yourself kit, except the instructions are clearly written and you don't have any Allen wrenches to keep track of. You'll come away from it with the tools and perspective you need to build the right processes for your project.

Your journey will start with a look at the basic principles of program management. Next, we'll briefly cover project management to help you understand the similarities and differences. Then you'll learn about a few fundamental skills that will touch all aspects of your program management work: relationships, decision-making, and meetings. From there, you'll learn about what's necessary to take a release from start to finish. You'll learn how to develop a schedule, plan features, track bugs, and more.

You'll see how you can apply the discipline of program management to your own open source project. You'll be guided to think about how the lessons in this book apply to your project. You'll see examples of what has worked well—and I'll also share some of my mistakes.

No matter how large or small your project is, you will be able to use what you learn in this book to be more intentional in managing your open source program.