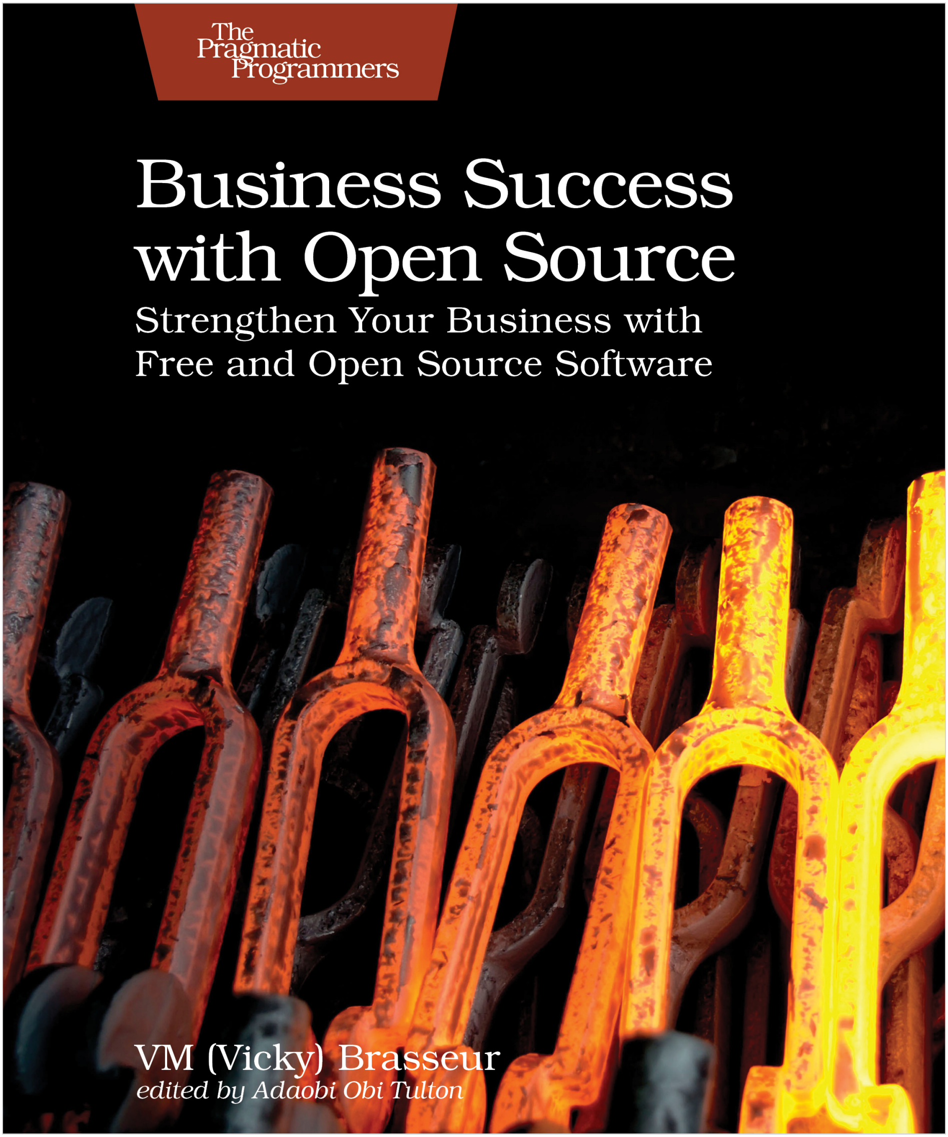

The Pragmatic Programmers

Business Success with Open Source

Strengthen Your Business with Free and Open Source Software

VM (Vicky) Brasseur

edited by Adaobi Obi Tulton

This extract shows the online version of this title, and may contain features (such as hyperlinks and colors) that are not available in the print version.

For more information, or to purchase a paperback or ebook copy, please visit https://www.pragprog.com.

Preface

Free and open source software (FOSS) is everywhere and is the driving force behind nearly all software developed today. According to some estimates, it's not uncommon for FOSS components to make up over 90 percent of a modern piece of software. A programmer will take one component—say, one that does authorization—then write some code to integrate it with another that provides communication over the internet, then to one that logs all errors and traffic, and then to yet another that displays a user interface based upon a selection of predesigned templates: dozens or hundreds of FOSS components, integrated and customized to support your company's unique business logic, brought together in days to create software that previously would have taken your company months or years to develop.

How much would you pay for that sort of boost to your company's innovation? Don't answer yet!

What if I told you that along with blazing fast innovation you also get a robust and resilient community of software developers who continually grow, maintain, and support this ecosystem of components? And what if I also told you that right this moment, somewhere in your company, someone is a member of that community and is using FOSS to provide a great deal of business value to your organization. You may not know what they're using or how. You may not know who is doing it. You may not even know it's happening, but I assure you that it is.

Now how much would you pay? $25,000 a year? $100,000? $1,000,000? What would you say if I told you that your company is already benefitting from free and open source software and it's doing it for the low, low price of *free*. That's right, you get the rapid innovation, the caring community, and heck, we'll even throw in increased software security, all for software that's probably not costing you a penny in licensing fees.

That, my friends, is already pretty amazing. But you can do even better.

It doesn't matter what industry your company is in or even whether it sells any sort of software; learning more about FOSS and how to use, contribute to, and release it can be the strategic edge that your company needs. With the proper knowledge and approach, FOSS can form the cornerstone of a digital transformation effort, increase developer retention, decrease recruiting cycles, ensure cutting-edge security, and reinforce the company brand—all this and more, by shifting your company's FOSS strategy from something that's accidental to intentional.

That's why you're here, reading this book. Welcome! You're going to learn a lot of useful stuff here, and when you use it you'll not only strengthen your company, you'll also strengthen the FOSS ecosystem that helps to support it. By the time you're done reading this book you'll have the knowledge to do all that and more.

Who Should Read This Book?

We'll prevaricate and say otherwise in public, but if you corner most authors, you can get them to admit that they'd like *everyone* to read their book. They'd like you to give it as presents at baby showers and graduations, as rewards for accomplishments big and small, and as thinking-of-you gifts to long-lost friends and lovers. That it may be a textbook about orbital dynamics doesn't matter; the more eyes on the work, the happier the author. However, when we return from dreamland into the real world, we sadly confess that, no, not everyone will get value out of our work. This book, alas, is no different.

Since that value is the most important thing here, the question of *who* should read this book is less relevant than *why* someone should read this book. What problems are you, the reader, facing and can this book help you with them? If your answer really is "orbital dynamics," then this will not be the book for you (maybe you should attend some baby showers or graduations). If your answer is that you need to work with Free Software or open source in some way in your job, then you're in the right place.

If you're faced with participating in a digital transformation effort, or reducing business risk, or building a product roadmap, or creating a company or team strategy, or if you need to round out your existing FOSS knowledge by filling in some gaps, then you'll get a great deal of value out of the book in your hands right now.

While the usefulness of the book is not limited only to this venue (I hope), the expectation is that you're reading it in some sort of a business context. You are a leader of some sort in your company or organization, be it C-suite or line manager, marketing or programming, entrepreneur or chairperson of the board, strategist or tactician. Your company's business model may not turn on creating and selling software products, but it does rely on software in some way to deliver value to its customers.

In the case of companies undergoing digital transformation, your company may now be experiencing whole new paradigms of software, such as microservices, containers, continuous integration and delivery, DevOps, and cloud computing. All of these situations depend upon FOSS. Your company and your career will thank you for reading this book and learning about how to make the most of free and open source software in a business context.

Note that one problem I didn't mention is when you, your team, or your employees would like to contribute to FOSS projects. This book certainly will help there, as it will walk you through creating policies and procedures for those corporate-sponsored contributions as well as selecting any projects on which your company would like to focus its efforts. The nuts and bolts of those contributions, however, are covered in my other book: *Forge Your Future with Open Source.*[1] I encourage you to get it for yourself and your team members so they can be more successful with their contribution efforts.

How the Book Is Organized

The matter of FOSS in the business world is larger and more complex than most leaders know. If they're aware of the concept at all, they see only the small part of it that concerns them most: license compliance, security audits, product releases, and so forth. Nearly every company with which I've worked has missed the bigger picture of how FOSS is an asset both to their business and to the world at large. By seeing only the trees and not the larger forest, these companies are overlooking not only huge strategic opportunities but also possibly some potentially company-ending risks.

This book provides an overview of that larger forest as well as some guideposts to help you get through it safely, though of course only you and your company can take the actual journey itself. You'll learn more than enough to start developing a comprehensive open source strategy, focusing on the elements that are most important for the success of your company and lines of business. Those elements tend to fall into one of two buckets: inbound and outbound. Along with the fundamentals, they form the three pillars of FOSS and business. They're also the parts into which the book is organized.

[1] https://fossforge.com

Part I: Fundamentals

In Part I, Fundamentals of FOSS, on page 1, you'll learn about the definitions of open source and Free Software, the difference between them, and why it matters for your business. You'll also learn the essentials of copyright and licensing in general, the types of FOSS licenses, and what makes them special to you and to all companies. This part of the book discusses some of the potential risks your company may face by using—or, more accurately, incorrectly using—FOSS as well as the many business benefits of doing so. Finally, you'll learn about the Open Source Program Office (OSPO) and the strategic role it can play in your company's business plan.

Part II: Inbound

Part II, Inbound (Using) FOSS, on page ?, first covers the "what even is?" question (hint: it's right in the name) before introducing the concept of FOSS supply chain. It then builds upon this to discuss matters like license compliance, scanning, archiving, auditing, and related complications, such as containers. You'll learn about FOSS sustainability and why a company ignores it at its own peril (not to mention that of the FOSS projects on which it relies). Policies and procedures receive some attention here, along with discussions on the various stakeholders to ensure they're included in all relevant conversations. Leaving them out sets those policies and procedures on a very steep uphill climb to success. Of course the policies you have only matter if people know about and follow them, so you'll also learn a bit about training and internal communications.

Part III: Outbound

Part III, Outbound (Contributing to and Releasing) FOSS, on page ?, follows Inbound's lead by getting us all on the same "what even is?" page before diving into a FOSS bucket that holds enormous but often neglected strategic potential for businesses. How can your company contribute to critical FOSS projects while still protecting its intellectual property, including patents? How can you give credit where credit is due but also ensure that your company receives the kudos it deserves for supporting FOSS? What does success look like for your company's FOSS contribution strategy, and what are some of the metrics that can help reveal whether efforts are on track?

Of course outbound FOSS is so much more than contributing to existing projects; it's also releasing new ones into the ecosystem. Why would your company do this, and how does it choose what to release? Once you've chosen what to release, how can your company do so in the most effective and correct way possible? Releasing FOSS projects is a lot like getting a puppy: it's fun to do, but once you have, then you need to take care of it and ensure it's healthy. What does that mean in this context, how can your company make that happen, and why should it even care—which is to say, what's in it for them? All of this information comes together when considering the topic of business models and how your company can incorporate FOSS into its plans for profitability.

As with most questions posed with respect to FOSS and business, the answers to all these are unique to your individual business and its goals. By the end of this book, you'll understand the variables involved, what that means for your business, and how to answer the question that makes the most strategic sense for your company as well as for the greater FOSS ecosystem.

Suggestions for Tools and Technologies

The tools and technologies available today are not going to be the same tools and technologies available tomorrow. The entire world of software moves at a fast pace, but free and open source software makes the rest look like snails. The public and open nature of FOSS allows it to evolve at incredible speed. It can be difficult to keep up with the latest developments. Thankfully, your company doesn't usually need to stay on the cutting edge of software. It can choose the tools and technologies available at the time it has a problem to solve, whatever those tools may be.

This rapid evolution of software makes it impractical for me to make recommendations for tools or technologies in this book. I could provide suggestions for tools, but that will only frustrate you when you try to use them and discover that they're very out-of-date, are no longer supported, or don't solve your newer problems. Already, recommendations I'd have made at the start of writing the book have been superseded by other tools.

In cases where well-established FOSS options are available, I'll list them. For the most part though, you won't find many "use this!" suggestions. What you *will* find are the concepts and keywords you'll need to do effective searches using your favorite search engine.

One Size Doesn't Fit All

Now that you have an idea of what's in the book, how can you make the most of that information?

First of all, know that there are guidelines and best practices for how to approach FOSS in your company or organization, and you'll find those in abundance within these pages. However, the practical application of those tips will look different for each company, and sometimes also for different divisions or teams within a single company. That's because each company has its own constraints, needs, and business strategies to consider when approaching this or any business-related decision or policy.

While you may be able to learn some interesting tricks from seeing how other companies approach FOSS, you cannot take their approach, drop it into your organization, and expect it to work any more than you can expect to drop new tooling into your software development life cycle and expect it to work without integration. Sure, it would be a heck of a lot easier if we *could* do those things, but we can't. Instead, you need to take the long way around and figure out the business value your company does or would like to receive from using and/or releasing FOSS, and then build strategies, policies, and procedures to help realize that value.

In lieu of the mythical one-size-fits-all solution for FOSS in business, this book provides not only the background you need to understand the FOSS-in-business problem space but also recommendations for how to gauge the business risks and values that FOSS can bring to your organization and then how to make the most of them. Most chapters first give you a thorough discussion and education about the topic at hand. Take from this what you and your company require for its business and operating models, skipping matters that don't apply and adding others that make sense for you, your people, your FOSS supply chain, and your business needs.

It's tempting to take a shortcut on this path, but it's unlikely to end well. I've seen too many companies end up in a FOSS quagmire that way, throwing millions of dollars away on FOSS projects, products, and strategies that are doomed to failure due to lack of knowledge and preparation. You'll find the price of the book and the time required to read it far easier to manage than a million-dollar failure (and have a considerably higher ROI as well).

Commit to the Effort

While we're talking about avoiding shortcuts, here's another thing you should keep in mind. Like all worthy efforts and strategies, your company's strategic FOSS initiatives will take time to develop, implement, and bear fruit. While a lack of knowledge and preparation certainly scuttles a lot of well-intended FOSS plans, I've seen many well-planned initiatives killed off before they've been given a chance to work. Usually this is due to unrealistic expectations on the parts of the implementors and their superiors. Contrary to common belief, the superpower of free and open source software isn't the software; it's the people who *create and support* that software who make FOSS the powerful and strategic tool that it is. Like any effort that involves people—such as digital transformation and other efforts involving human culture—changes require time to settle in and take effect.

As you read the chapters that follow, keep this in mind. Certainly your company can gain benefits from FOSS in a quarter or less, but the *really* good stuff is further out on the time scale. Plan accordingly, but most especially set adequate expectations with all of the stakeholders. This concept isn't new to you; strategic efforts take time, and a FOSS initiative is no different.

On the Matter of Jargon

Let's talk for a moment about, well, talking—more specifically, about language and communication and how the former often defeats attempts at the latter.

All too often we put together a string of words that *we* understand without pausing to consider whether the recipient will as well. We assume, and often incorrectly, leading to a failure in basic communication requirements. The prize for the most common type of words to cause miscommunication goes to jargon, which quickly turns even your native language into a foreign one, depending on the audience. Every industry has its own dedicated vocabulary—its own jargon—that its practitioners create and share. This jargon frequently includes a metaphorical alphabet soup of acronyms, each one obscuring a Very Large Idea behind a very small set of letters. This language compression is invaluable for speeding up communication between industry peers. We all do it, and there's usually nothing wrong with it.

The subject of this book requires a lot of jargon. There's no way to escape that. First there's the language of business, with its TCOs and its KPIs; then the language of software and technology, with its APIs and CSPs, its PRs and CI/CDs; next up are the languages of intellectual property law and compliance, with CLAs and SBOMs; and of course there's the language of FOSS itself, with its GPLs and MITs.

It's highly unlikely that you're well-versed in all of these different languages, and that's OK. None of us knows everything. However, if you're going tackle the matter of FOSS in your company, then you need at the least to become familiar with the many different languages involved. For that reason this book won't avoid jargon, but it *will* go out of its way to explain it, no matter its source language. Readers with a more technical background will learn some business terms, and business folk will learn technical ones. By the time we reach the end, we'll all be able to communicate much more effectively with each other.

Two resources are included in the book to help with this process. The first is Appendix 1, Acronyms & Initialisms, on page ?. This does no more and no less than unpacking acronyms out to their constituent words, since often all we need is that little nudge to allow us to remember and understand. The second resource is Appendix 2, Glossary, on page ?, which collects and defines jargon from all of the languages involved with the Business of FOSS. You'll find a lot to learn here, and having a glossary at hand means you don't have to memorize a bunch of unfamiliar terms while you're also trying to learn how to make your company more successful through FOSS.

Eschew Obfuscation

Communication is difficult enough on its own. Why make life harder for everyone by using uncommon words where plain, straightforward ones will do? Complex linguistic constructions obscure intent and misrepresent unpretentious hypotheses as euphuistic intellections; which is to say, fancy words make it hard to communicate. It's also a cruel thing to subject your readers to, especially if English isn't their primary language (as is the case for many of you—hello!).

I'm a big fan of plain language, and that's what you'll get in this book. Sometimes sentences may sound a little old-fashioned, since ending a sentence with a preposition is something up with which I will not put. Aside from that, though, I'll do my best to use (never utilize) direct, clear writing.

Links and References

A lot of links and references appear in the chapters that follow, so you'll rarely (if ever) need to raise a *[citation needed]* flag. However, keeping track of all these links so you can find them again can be rather a bother. That's why you'll find them all collected in Appendix 3, Links from Chapters, on page ?, organized by chapter. Because everyone's mental models are different, you may remember a link from Chapter 1, Lay the Foundation, on page ?, while I remember the same link being mentioned in Chapter 3, Licenses: The Rules of IP Engagement, on page ?. Therefore you'll find links repeated between chapters in this appendix, and they'll be listed in the order in which they appear in their respective chapters. Hopefully this will make it a lot easier for you to find what you're looking for quickly.

Among the appendices you'll also find Appendix 5, Other Useful Resources, on page ?. Unsurprisingly, this lists other books that I recommend if you'd like to learn more about using, releasing, managing, and contributing to free and open source software. Since it's in a book, this list naturally is static. However, it largely mirrors the list available on https://fossbooks.com, which I will update as new recommended books become available.

Join the Online Community for More Resources

The Pragmatic Bookshelf has provided a forum dedicated to this book on the DevTalk platform.[2] You can join the community of readers in this forum, discussing topics related to the book and helping each other. You can also use the forum to report book errata and make suggestions that might end up in a future edition. Also, there's a dedicated book web page on the Pragmatic Bookshelf website.[3]

By this point you've figured out that this isn't going to be your standard stuffy business book or inscrutable tech tome. You're here because you have problems to solve and questions to answer, and you don't want to waste time wading through some stifling text to get that done. Aside from the unavoidable need for jargon, this book will keep it casual, readable, and above all *practical*. You and your company have goals to meet. This book will help.

[2] https://devtalk.com/books/business-success-with-open-source/
[3] https://pragprog.com/titles/vbfoss/business-success-with-open-source/

Part I

Fundamentals of FOSS

The first part of the book gives you a solid understanding of the concepts you'll need to be successful with open source.

You'll learn what open source and Free Software are, how they differ, and a great deal about intellectual property, copyright, and licensing.

After that, answer the "why does this matter?" question by learning of the risks and benefits of free and open source software (FOSS) to your business.

The part ends with an introduction to the beating heart of open source in your organization: the Open Source Program Office (OSPO).