OPEN SOURCE LICENSING

101

FROM CONCEPTS TO
PRACTICAL IMPLICATIONS

GUSTAVO PINTO

Open Source Licensing 101
From Concepts to Practical Implications

Gustavo Pinto

First edition: 30 October 2020
Last edition: 30 October 2020

Preface

WITHOUT open source licenses, there wouldn't be open source. Open source licenses create a legal framework that plays key role in the widespread adoption of open source projects. Any open source project must be released under one (or more) open source license. Without a license, any source code available on the internet could not be openly (re)distributed.

As a natural consequence, open source licenses is one of the most important non-executable parts of any open source software project. Open source licenses not only drive how one can use an OSS but also ensure to what extent others can reuse it.

However, due to its non-technical nature, developers often misuse or misunderstand open source licenses. I believe there are at least *three* reasons that may justify this fact.

- First, the vocabulary used in open source licenses is not part of the developer's daily vocabulary; it then may require developers to gain additional knowledge to understand software licensing.

- Second, there are hundreds of open source licenses readily available for developers use. Indeed, many of these licenses are quite similar, differing by just one or two minor changes in the license text. Since developers are always busy, they might have minimal time to catch and understand these corner cases.

- Third, many of the textbooks about open source licensing devote a good proportion of the document to explain the concepts, ideas, and differences between the existing open source licenses. However, developers might also have difficulty relating how these abstractions could make any sense in their daily activities.

In this book I argue that the lack of a clear understanding of software licensing could challenge the existence of any open source software project. Let me give the reader one example.

The case of the Lerna project

In 2018, one of the Lerna project core maintainers decided to make some small changes in the license definition. The maintainer made a pull request that changed the used MIT license to include *restrictive* and *retroactive* clauses.[1] Not only its userbase was greatly impacted with this pull-request, but also (and intentionally) tech companies such as Microsoft, Amazon, Dell, Xerox, Canon, LinkedIn, and few other were not allowed to use Lerna to support their software development activities anymore.

According to the maintainer, the motivation to perform this change was because these tech companies were collaborating with the US Immigration and Customs Enforcement (“ICE"). On the maintainer’s views, “these companies care only about the millions of dollars that ICE is paying them and are willing to ignore all the horrible things that ICE does." Although one might think that there are good intentions behind these changes, there are also two tremendous problems hidden on it.

- First, by restricting the use of the software to certain groups of users, the Lerna would actually be discriminating against them.

[1] https://github.com/lerna/lerna/pull/1616

However, non-discrimination is a core value of open source. According to the fifth clause of the Open Source Definition,[2] the "license must not discriminate against any person or group of persons".

- Second, there is no such thing as a retroactive license change. The Free Software definition is unambiguous when it says that: "if the developer of the software has the power to revoke the license, or *retroactively add restrictions to its terms*, without your doing anything wrong to give cause, the software is not free"[3]. Yes, you can change the license, but it would only work for new users using the latest release of your open source software. You cannot change a license expecting the existing users would be affected. These current users could keep using the old version of your software (with the previous license), and this is just fine.

That single pull-request had several repercussions.

Eric Raymond, the author of the seminal "The Cathedral and the Bazaar" book [14] and one of the Open Source Initiative creators, also shared his views about this issue.[4] Among other things, he said that:

> "The Lerna project's choice is, moreover, destructive of one of the deep norms that keep the open-source community functional – keeping politics separated from our work".

Open source developers, when not coding, are free to do whatever they want. It does not matter for the code if the developer supports ICE or a far right-wing party. By mixing politics with open source, through the (mis)use of its license, the Lerna project's whole chain of users was impacted.

Long story short: the pull-request was reverted, the maintainer was kicked out of the project, and the open source community around Lerna

[2] https://opensource.org/osd-annotated

[3] https://www.gnu.org/philosophy/free-sw.en.html

[4] http://esr.ibiblio.org/?p=8106

was hurted. How? By using Lerna to advocate on his own behalf, the maintainer also indicated that the project Lerna shared his political views. This, in turn, could contribute negatively to attract new developers that may not share the same political view—all of this due to 22 lines added in the license file.

Why I decided to write this book?

Whether consuming or contributing, anyone interested in open source needs to know and understand basic open source licensing requirements.

Although there are many books and websites out there doing a great job explaining what open source licenses are all about and how they differ. However, I think that these mediums are lacking a more practical orientation for developers. For instance, it is not always clear the implications that developers may face when they misuse open source licenses.

With this book, I intend to cover practical and existing scenarios regarding licensing uses and misuses. My ultimate goal is to make licensing issues more exciting and appealing to developers. I would expect that non-lawyers would readily understand this book.

Who should read this book?

Short answer: anyone interested in open source software. Long answer: several stakeholders, including (but not limited to):

- If you are a developer and have to use open source software daily, **this book is for you!**

- If you are a project maintainer and still struggle to understand what license you should use, **this book is for you!**

- If you are a researcher interested in studying open source licensing issues, **this book is for you!**

- If you are a professor teaching open source or related topics, **this book is for you!**
- If you care about open source in general, **this book is for you!**

What this Book is not About?

In this book, we do not intend to discuss the ideas behind free/open source software in general. For this matter, there are many other good books out there, which I recommend the reader to take a careful look at, perhaps even before reading this book.

About the Terminology

A more experienced reader would rapidly notice that we are somehow mixing the terms “free software” and “open source software” throughout this book. This document uses “open source" for consistency but could have used “free software" just as easily. There are two reasons that support this our decision to stick with “open source”:

1. There is a common misconception that “free" refers only to copyleft licenses and “open source" refers only to non-copyleft (so-called “permissive") licenses.[5] However, this interpretation is rejected by the majority of the license-reviewing bodies in the field, including the Free Software Foundation and the Open Source Initiative. Just as an example, a license can be both qualified as a free license (by the Free Software Foundation) and as an open source license (by the Open Source Initiative). One perhaps interesting example is the APSL license, designed by Apple to license the Darwin operating system (which is the basis of many Apple components including macOS, iOS, watchOS, tvOS, and iPadOS). Even not being a true

[5] http://developers.slashdot.org/story/12/04/22/0211251/open-source-project-licenses-trending-toward-open-rather-than-free

copyleft license, FSF said that "it is OK to use and improve software which other people release under this license."[6]

2. the commercial vain of open source, discussed by the Open Source Initiative, has gained much more traction than the altruistic view promoted by the Free Software Foundation. Consequently, the term "open source" became more frequently employed not only by developers but also in social media.

Still, as we shall mention in Chapter 2, there is rich literature about the philosophical debate of free vs. open source. However, in this book, we do not intend to contribute to this debate, but we will leave the reader to exciting pointers throughout the book.

Conventions Used in this book

The following typographical conventions are used in this book:

Plain text: Indicates ordinary content

Italic: New terms and emphasis.

Boxes

> This is a tip box.

> This is a warning box.

[6] https://www.gnu.org/philosophy/apsl.en.html

Disclaimer

This book was written by Gustavo Pinto, a professor of software engineering. The observations made throughout the book are mostly based on my own experience while researching open source software and communities. However, when it comes to open source software (and, obviously, open source licensing), my views are somehow aligned with the Open Source Definition (OSD)[7], curated by Open Source Initiative (OSI). As a consequence, the reader might often see several links to OSI and OSD to support some discussions. I am not anyhow affiliated with OSI, though.

Moreover, since I am not a lawyer, this book does not constitute legal advice. Here I provide a high-level, highly-simplified, highly-generalized overview of my own understanding of software licensing. In fact, many readers could also notice that my views could be biased or even not entirely accurate. And that's OK. Open source software licensing is a rather complex landscape with *many* corner cases. For educational purposes, I think it is acceptable to over-simplifying things over completeness.

In my understanding, this book is far from being a final, definitive guide on open source licensing. Instead, this book aims to help software developers, students, researchers, instructors, etc. by providing a basic understanding about open source software licensing. I wholeheartedly believe that, after understanding the basics, the reader would easily get acquainted with many of the corner cases, if needed. That being said...

> ❗ Anybody who incorporates open source components into proprietary programs must look for legal advice!

About the author

Gustavo Pinto is a software engineering researcher and professor at the Federal University of Pará, Brazil. Gustavo Pinto got his Ph.D. from the Federal University of Pernambuco, Brazil, in 2015. Since then, he became

[7] `https://opensource.org/osd`

a fairly active researcher, studying topics related to open source software and communities. Know more online at `www.gustavopinto.org` or follow him on Twitter (`@gustavopinto`).

Contents