



# International Journal of Advanced Academic Studies

E-ISSN: 2706-8927

P-ISSN: 2706-8919

IJAAS 2019; 1(2): 85-88

Received: 13-08-2019

Accepted: 18-09-2019

**Sandeepi**

Assistant Professor,  
Department of Computer  
Science, Govt. P.G. College,  
Safidon, Haryana, India

## The difference between free and open-source software

**Sandeepi**

### Abstract

Whether we use 'open source software' or 'free software', the fact is that both terms refer to particularly the same set of licenses and software, and at the same time, each of the term implies different underlying values. Although there are different parameters for free software licenses and open source licenses. In other sense, both these terms are related to and suggest the same set of licenses; they arrive at that set through different routes. In case of if the licenses are identical, a person's choice of terminology may imply different emphasis in values. Open software emphasis on the practical results because of these licenses. Various scholars and researchers find the two terms describing almost the same category of software, but both stand for views based on fundamentally different values.

**Keywords:** Open, free, software, licenses, differentiate

### Introduction

One aspect of software development that many people tend to forget about is how the software should be licensed. A software license dictates how the code can be used and distributed by licensees (the end users), which can make a significant impact on how widely the technology gets adopted. Most modern software is sold under a proprietary license which allows the publisher or creator to retain the intellectual property rights of the software.

However, there's an alternative viewpoint which contends that this puts an unnecessary level of control into the hands of software publishers. By preventing licensees from copying and changing software's source code, the idea holds, proprietary software publishers stifle innovation and hold back the potential growth of new technologies. This stance inspired the creation of licenses which grant users the rights to study, change, and share the software's source code to their liking. Software licensed in such a way is usually known by one of two names: "free software" or "open-source software."

Broadly speaking, both terms refer to the same thing: software with few restrictions on how it can be used. From the perspective of their proponents, both free and open-source software are safer, more efficient, and work more reliably than their proprietary counterparts. Why, though, do we have two labels for the same thing? The answer involves a bit of history, and an understanding of the nuances that form two separate but closely related movements.

The idea that an individual working with a piece of software should be allowed to view, edits, and share its source code without legal consequence is nothing new. Before 1970s, software was typically distributed along with its source code, among the main reason behind that was software was usually hardware-specific and end users would have to modify it to run on their particular machine or to add special functionalities.

Most people who interacted with computers around this time did so in a strictly academic or research setting which meant that computing resources were often shared, and changing software to create more efficient workflows or more reliable solutions was widely encouraged. For example, UC Berkeley's Project Genie developed the Berkeley Timesharing System-a time-sharing operating system built from scratch-by hacking the source code of the lab's SDS 930 computer.

As software became more complex and expensive to produce, though, software companies sought ways to halt the unbridled sharing of source code in order to protect their revenue streams and deny competitors access to their implementation. They began putting legal restrictions on their products, including copyrights and leasing contracts, and also started distributing their products under proprietary licenses. By the end of the 1970s, most software companies had stopped shipping software with the source code included. This led many longtime computer users to vocalize their dissatisfaction, and their ethos would eventually form the foundation of the Free Software Movement.

**Corresponding Author:**

**Sandeepi**

Assistant Professor,  
Department of Computer  
Science, Govt. P.G. College,  
Safidon, Haryana, India

The Free Software Movement was largely the brainchild of Richard Stallman. Stallman began his studies in computer science in the early 1970s before the rise of proprietary software licenses, and he worked as a researcher at the MIT Artificial Intelligence Laboratory through the early 1980s. Having been a member of the academic hacker community for over a decade, he had grown frustrated by the spread of proprietary software and came to see it as a violation of people's rights to innovate and improve existing software.

This, according to Stallman, puts an unfair burden on users and developers who would otherwise be able to change the code to suit their own needs or alter it to serve a new function. Thus, the GNU Project can be seen as both a response to the rise of proprietary software as well as a call back to the previous era of freely shared source code and collaborative software development.

In 1985, Stallman built on the GNU Project by founding the Free Software Foundation (FSF), a nonprofit organization dedicated to promoting the concept of free software to the wider public. Stallman would also later develop the GNU General Public License, a software license which guarantees the rights of end-users to run, view, and share source code freely.

According to the FSF, for a piece of software to be considered truly "free," its license must guarantee four essential freedoms to its users:

- The freedom to run the program as you wish, for any purpose.
- The freedom to study how the program works and change it so it does your computing as you wish. Access to the source code is a precondition for this.
- The freedom to redistribute copies so you can help your neighbor.
- The freedom to distribute copies of your modified versions to others. By doing this you can give the whole community a chance to benefit from your changes. Access to the source code is a precondition for this.

The FSF considers any software that fails to meet each one of these criteria as non-free and, therefore, unethical.

### The growth of open source

Stallman had chosen the label "free software" to relate the idea that users would be free to change and share source code as they saw fit. This has created some confusion over the years, as many people assume "free software" refers to any software that can be obtained for zero cost (which would be more accurately labeled as "freeware" or "shareware"). The FSF famously explains the name choice with the line, "think of free as in free speech, not as in free beer."

In the second half of the twentieth century, though, there was a growing worry among some GNU and Linux enthusiasts that this dual meaning would cause a large share of users to miss the philosophy behind free software and its advantages over proprietary code. The FSF had also become known for its hard-line ethical stance against proprietary software of all kinds. There was concern among some free software advocates that this approach was too unfriendly to business interests, and would ultimately hamper the spread of the Free Software Movement.

### The cathedral and the bazaar

In 1997 Eric S. Raymond, then free software advocate and developer, wrote *The Cathedral and the Bazaar*, a widely-cited essay which compares two different development models used in various free software projects. "The Cathedral" refers to a top-down development model where an exclusive group of developers produce the code, exemplified by the development of GNU Emacs. "The Bazaar," on the other hand, refers to a method in which code is developed publicly over the internet, as was the case for the development of the Linux kernel.

The essay's central argument is that the Bazaar model is inherently more effective at finding and resolving software bugs, as more people are able to view and experiment with the source code. Thus, Raymond argued, utilizing a community-driven, bottom-up development process results in safer, more reliable software.

Partially in response to the ideas presented in *The Cathedral and the Bazaar*, Netscape released the source code of its Communicator web browser as free software in early 1998. (The Communicator source code would later form the basis of Mozilla Firefox 1.0). Inspired by the commercial potential that Netscape saw in this source code release, a group of free software enthusiasts (including Raymond, Linus Torvalds, Philip Zimmerman, and many others) sought to rebrand the Free Software Movement and shift its focus away from ethical or philosophical motives. The group chose "open source" as its label for freely shareable software in the hope that it would better reflect the business value of a collaborative, community-driven development model.

Shortly thereafter, the Open Source Initiative (OSI) was founded by Raymond and Bruce Perens to encourage both the use of the new term as well as the spread of open-source principles. OSI also developed the Open Source Definition—a list of ten principles which a software's license must adhere to for it to be considered open-source:

1. **Free Redistribution** - The license shall not restrict any party from selling or giving away the software as a component of a larger software distribution containing programs from multiple sources.
2. **Source Code** - The program must include source code, and must allow distribution in source code as well as compiled form.
3. **Derived Works** - The license must allow modifications and derived works, and must allow them to be distributed under the same terms as the license of the original software.
4. **Integrity of the author's source code** - The license may restrict source-code from being distributed in modified form only if the license allows the distribution of "patch files" with the source code for the purpose of modifying the program at build time.
5. **No Discrimination against persons or groups** - The license must not discriminate against any person or group of persons.
6. **No discrimination against fields of endeavor** - The license must not restrict anyone from making use of the program in a specific field of endeavor.
7. **Distribution of license** - The rights attached to the program must apply to all to whom the program is redistributed without the need for execution of an additional license by those parties.

8. **License must not be specific to a product** - The rights attached to the program must not depend on the program's being part of a particular software distribution.
9. **License must not restrict other software** - The license must not place restrictions on other software that is distributed along with the licensed software.
10. **License must be technology-neutral** - No provision of the license may be predicated on any individual technology or style of interface.

### The differences between free and open-source software

As far as most people are concerned, the difference in meaning between "free software" and "open-source software" is negligible, and comes from a slight difference in approach or philosophy. As the Open Source Initiative sees it, both terms mean the same thing, and they can be used interchangeably in just about any context. They simply prefer the "open source" label because they believe it provides a clearer description of the software and its creators' intent for how it should be used.

For the "free software" camp, though, "open source" doesn't fully convey the importance of the movement and the potential long-term social problems caused by proprietary software. The Free Software Foundation sees OSI as being too concerned with promoting the practical benefits of non-proprietary software (including its profitability and the efficiency of a community-driven development model), and not concerned enough with the ethical issue of restricting users' rights to change and improve code on their own terms.

To decide whether or not a given piece of software is free or open-source depends on which license it's distributed under and whether that license is approved by the Open Source Initiative, the Free Software Foundation, or both. There is a good deal of overlap between which licenses are approved by which organization, but there are a few exceptions. For example, the NASA Open Source Agreement is an OSI-approved license which the FSF views as too restrictive. Thus, the FSF discourages anyone from using software distributed under that license. Generally speaking, though, there's a good chance that if it can be described as free software, it will fit the definition of open-source software as well.

### Alternative names

For the last few years, several other names for this kind of software have been used to put an end to this debate. "Free and open-source software"-oftentimes shortened to "FOSS"-is one of the most widely used, and is considered to be a safe neutral between the two. The term "libre software" ("libre" is a word which is derived from several Romance languages that roughly means "the state of liberty") has gained a following of its own, so much so that the acronym "FLOSS" (means "free/libre and open-source software) has also become fairly common.

It should be pointed out that both free and open-source software are entirely different from software in the public domain. Free and open-source software defines its freedoms through its licensing, while public domain software may adhere to some of the same virtues, but does so by falling outside the licensing system. An important distinction between both free and open-source software is that works based on free or open-source source code must also be

distributed with a FOSS license. Software released into the public domain does not have this requirement.

Another aspect with public domain software stems from the fact that not every country in the world recognizes non-copyrighted content. This makes it impossible to make a globally recognized statement that a piece of software is in the public domain. Thus, neither the FSF nor the OSI encourages developers to release software into the public domain.

The terms "free software" and "open-source software" can be used in place of each other for most contexts, and whether someone prefers one over the other usually comes down to a matter of semantics or their philosophical outlook. However, for many programmers that are looking to develop software and get it out to the public or for activists hoping to change the way people see and interact with technology, the difference can be an important one. Thus, when releasing new software, it's essential to carefully weigh the pros and cons of different licenses-including proprietary licenses-and choose the one those best suits your particular needs

### Conclusion

It goes without saying that lots of organizations as well as service providers such as IBM, Hewlett Packard are already doing business with all the open source software at the present and it is likely to distribute to many other organizations as well. For These Reasons, the day is not far enough when the software industry will likely experience a great transformation in its application of free software in major as contrasted to that of the proprietary software. In our viewpoint the reality that there exists an option of source codes being exposed to the users has the potential to conquer their guidelines set by the proprietary software and creates its very own specialized niche within the software ecosystem. Thus, concluding by International Journal of Computer Applications (0975 – 8887) Volume 114 – No. 18, March 2015 31 expressing there has been a switch over through the Proprietary Software model to the open source software model and it is right here to stay but one major question which arises now is –whether the proprietary software manufacturers should be able to cope up with the obstacles positioned in opposition to them by the OSS and will they have the ability to contend with the OSS?

### References

1. Coppola C, Neelley E. Coppola C, Neelley E. Williams Roy. 2nd European Conference on E-Learning Glasgow Caledonian University, Glasgow, 6-7 November 2003. Academic Conferences Limited.) Scott Wilson, Open source in higher education: how far have we come? (The guardian, 28 March 2013), 2003-2004.
2. Anderson Chris. Free: The Future of a Radical Price, 1st paperback edition, Hyperion. Bae, Sang Hoo, Jay Pil Choi, A Model of Piracy, Information Economics and Policy, 2006-2009, 18.
3. Besley Timothy, Torsten Persson, Daniel M. Sturm. Political Competition, Policy and Growth: Theory and Evidence from the United States, Review of Economic Studies, 2010, 77(4).
4. Bitzer Jürgen. Commercial versus Open Source Software: The Role of Product Heterogeneity in Competition, Economic Systems, Business Software

- Alliance (BSA), Eight Annual BSA Global Software 2010 Piracy Study. 2004-2011; 28(4):369-381.
5. Casadesus-Masanell Ramon, Pankaj Ghemawat. Dynamic Mixed Duopoly: A Model Motivated by Linux vs. Windows, *Management Science*, 2006, 52(7).
  6. Himanen P. *The hacker ethic and the spirit of the information age*. New York: Random House, 2001.
  7. Kuhn B, Stallman R. Freedom or power? Retrieved February 20, 2006, from <http://www.gnu.org/philosophy/freedom-or-power.html>
  8. Levy S. *Hackers: Heroes of the computer revolution*. London: Penguin. Locke, J, *Second treatise of government*. Indianapolis: Hackett, 1690-1984.
  9. Mikkonen T, Vainio N, Vadén T. Survey on four OSS communities: Description, analysis and typology. In N. Helander & M, 2006.
  10. Mäntymäki (Eds.), *Empirical insights on open source business*. Tampere: Tampere University of Technology and University of Tampere. Retrieved June 27, 2006, from <http://ossi.coss>.
  11. Moglen E. Enforcing the GNU GPL. Retrieved February 20, 2001-2006, from <http://www.gnu.org/philosophy/enforcing-gpl.html>
  12. Raymond ES. Shut up and show them the code. Retrieved February 20, 1999-2006, from <http://www.catb.org/esr/writings/shut-up-and-show-them.html>
  13. Stallman R. New UNIX Implementation. Post on the newsgroup net. Unix-wizards. Retrieved February 20, 1983-2006, from <http://groups.google.com/groups?selm=771%40mit-eddie.UUCP>
  14. Stallman R. GNU General Public License version 1. Retrieved February 20, 1989-2006, from <http://www.gnu.org/copyleft/copying-1.0.html> Stallman, R. (2002a). *Free software, free society: Selected essays of Richard M. Stallman* (pp. 41-43). Boston: GNU Press.
  15. Gay J (Ed.), *free software, free society: Selected essays of Richard M. Stallman* (pp. 41-43). Boston: GNU Press.
  16. Stallman R. GNU General Public License version 2. In J. Gay (Ed.), *free software, free society: Selected essays of Richard M. Stallman*, 2002b.