

Towards the Next Generation of Open Source Software A Survey of the State-of-the-Art and Possible Alternatives

C. Raja Kumar¹; M. Rajinikannan²

Assistant Professor, Department of Computer Science and Engineering,
SBM College of Engineering and Technology, Dindigul, Tamil Nadu, India¹;

Associate Professor, Department of Computer Applications,
PSNA College of Engineering and Technology, Dindigul, Tamil Nadu, India²

ABSTRACT

The Software developers have probably used a lot of software over the years for the purposes of development, testing, building and many more. Like the same way an end use have used a lot more software for their business purpose. From late 1070s, open source software began and steadily evolved into a sophisticated movement which has produced some of the most stable and widely used software packages ever produced. In this paper, we would like to trace those open source software along with its alternative commercial software. This will help the new budding IT industry to choose right software for their needs.

General Terms

Open Source Software.

Keywords

Open Source, OSS, FOSS, Closed Source, Commercial.

1. INTRODUCTION

There are different types of free software. They are generally categorized as freeware, shareware, or open source. Freeware usually refers to software that can be downloaded and used for free but with little or no support and no access to its source code. Shareware can be tried for free but we are expected to pay for it if we continue to use it after a trial period. The shareware developer has full control of the intellectual property and is responsible for providing support and enhancements. Source code access is restricted. Open source software is released with its source code and is free to use, modify and redistribute. It is supported by an online community of developers and users.

Under the closed-source model source code is not released to the public. The closed-source software is maintained by a team who produces their product in a compiled-executable state, which is what the market is allowed access to. Microsoft, the owner and developer of Windows and Microsoft Office, along with other major software companies, have long been proponents of this business model.

The reason to go for the free software is varies from user to user. Basic reasons are freedom, flexibility, cost and customizable. This following Figure 1 by Chao-Kuei [1]

explains the different categories of software.

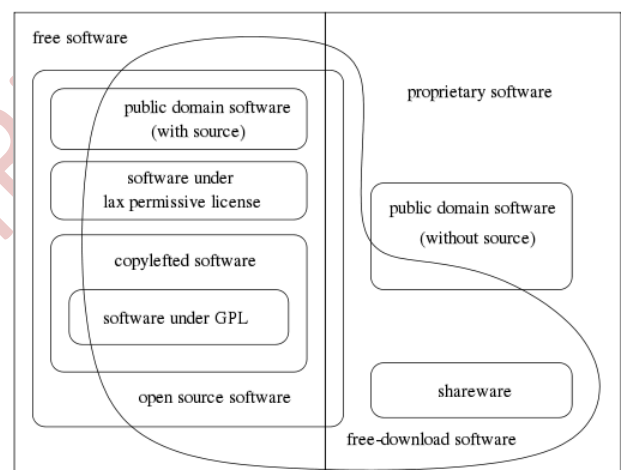


Fig 1: Different Categories of Software

Free software is software that comes with permission for anyone to use, copy, and/or distribute, either verbatim or with modifications, either gratis or for a fee. In particular, this means that source code must be available. "If it's not source, it's not software." This is a simplified description; see also the full definition.

The term "open source" software is used by some people to mean more or less the same category as free software. It is not exactly the same class of software: they accept some licenses that we consider too restrictive, and there are free software licenses they have not accepted. However, the differences in extension of the category are small: nearly all free software is open source, and nearly all open source software is free.

We prefer the term "free software" because it refers to freedom—something that the term "open source" does not do.

Proprietary software is another name for non-free software. In the past we subdivided non-free software into "semi-free

software", which could be modified and redistributed non-commercially, and "proprietary software", which could not be. But we have dropped that distinction and now use "proprietary software" as synonymous with non-free software. "Commercial" and "proprietary" are not the same! Commercial software is software developed by a business as part of its business. Most commercial software is proprietary, but there is commercial free software, and there is noncommercial non-free software.

2. OPEN SOURCE SOFTWARE

There are two statements named Linus' Law [3].

- Eric S. Raymond concerning software bug detection by a community.
- Linus Torvalds about the motivations of programmers.

Linus's Law as described by Raymond is a claim about software development, named in honor of Linus Torvalds and formulated by Raymond in his essay and book "The Cathedral and the Bazaar" (1999). The law states that "given enough eyeballs, all bugs are shallow" What that means is that the more people who can see and test a set of code, the more likely any flaws will be caught and fixed quickly. It's essentially the polar opposite of the "security through obscurity" argument used so often to justify the use of expensive proprietary products, in other words.

2.1 Advantages to having an open model

There Larger Developer Support – As its open source anyone can develop for the platform, giving it a much larger support for developers and gives them more of a feeling of ownership as they can alter whatever they like.

Customizable – In a closed source scenario, developers are given only options to change what the original developer chooses, but having full control lets them customize the look feel and even features of the software giving it much more possibilities

More Secure – Open source is much more transparent than closed, this means that anyone can look over the code, by having thousands of people reading through our code, bugs and vulnerabilities are located much quicker and submitted for fixing, it also lets us know if the bug has been fixed as we can check the code after each release.

Extended Community Support – As a product ages the original developer might move on and stop developing leaving the product to age with no new fixes or features, but if it's open then usually the community takes over and continues working on it allowing the usable life of the product to be extended well beyond what the original developer intended.

3. COMMERCIAL SOFTWARE

Commercial software, or sometimes payware, is computer software that is produced for sale or that serves commercial purposes, licensed under exclusive legal right of the copyright holder with the intent that the licensee is given the right to use the software only under certain conditions, and restricted from other uses, such as modification, sharing,

studying, redistribution, or reverse engineering. Usually the source code of proprietary software is not made available.

3.1 Advantages to having an closed model

Less confusion for customers – Many people aren't sure what the advantages of the different versions of android, By not allowing people to alter core features it makes it easier to pick the product we want, there is only one version of iOS that people need to learn about

Unified experience – Because its close the developer can choose the layout, the features, the options, the colors, and pretty much anything they choose. This seems restrictive, but at least guarantees that every user will have a similar experience and gives it a standard that can't be changed.

More profitable – A closed model can be greatly profitable as we can charge money for developers to use our API, and can lock out competition by not making design available for them to see.

4. BACKGROUND STUDY

Open Source Software (OSS) can be used in any type of software, from word processing to cloud computing to EDI (electronic data interchange). Some of the most famous and ubiquitous pieces of software, such as Linux and Mozilla Firefox, are OSS, yet some people are still hesitant to use less well-known pieces of open-source software. If we're like most people, probably one of the following reasons [2] is preventing us from using open source software.

Reason 1: We need too much support

One of the biggest advantages of OSS is that as long as we have someone with the know-how, we can completely customize the software to our needs. Because of the monies saved on licensing fees, using OSS is beneficial even if we don't do any customization; however, if we do customize, we'll need someone on hand who knows how to use open source software.

In addition, most proprietary software includes free tech support from the company, while OSS companies charge for those services. But this fee is countered by the fact that the code is open source, making it easier for IT team to learn how to use it.

Another reason that people think they need too much support while using OSS is that this type of software has a reputation for being relatively inaccessible to the average user. Unlike proprietary software, which is tested extensively for user-friendliness, OSS software is often written by and for computer professionals. Choosing software wisely helps eliminate this potential issue.

Reason 2: It infringes on intellectual property

We can negotiate with proprietary software companies about indemnification for intellectual property (IP) infringement, but that's not possible with OSS companies. Plus, the rules that govern IP for OSS are complex. However, we can counter this issue by purchasing indemnification insurance through a third-party vendor like OpenLogic [4]. In either situation, reading the terms of the license helps mitigate the risk of committing IP infringement.

Reason 3: It isn't reliable

Often without strong central management, the OSS

community must identify and provide solutions for errors with the software. This leads some to worry that problems will not be fixed, as compared to traditional software, which has centralized management and a dedicated team of developers to fix any issues.

In reality, the opposite is true. Repeatedly, issues with OSS are quickly fixed thanks to the work of the community members, whereas with a proprietary software company, users need to wait for the release of the next software update to fix a bug.

Reason 4: It isn't secure

This is perhaps the biggest misgiving that people express regarding OSS: Since the code is open, any opportunist can identify and exploit the program through hacking and viruses. Proprietary software companies, on the other hand, have team members dedicated to ensuring the security of their software.

Some risk is associated with using any software, and the overall risk associated with OSS is not higher than with any other type of software. While it's true that anyone can look at and potentially exploit the code, it's also true that anyone can look at the code to identify potential causes of security breaches and address them immediately.

The concerns that people have about OSS are not completely unfounded, but each concern can be mitigated with an understanding of the software in question. In many cases, using this type of software helps companies save money while also getting a product that is better suited to their needs. Once companies learn how to use open source software - and how to mitigate some of the risks associated with it many others may reap great benefits.

5. FEATURES OF OPEN SOURCE SOFTWARE

With the many business and government organizations that now use open source software such as Linux, it's becoming increasingly clear that price is not the only advantage such software holds. If it were, companies that adopted it during the Great Recession would surely have switched back to the expensive proprietary stuff as soon as conditions began to ease, and that's clearly not the case. Rather, free and open source software (FOSS) holds numerous other compelling advantages for businesses, some of them even more valuable than the software's low price [5].

Security

"Security through obscurity" may be a catchy phrase. It's hard to think of a better testament to the superior security of open source software than the recent discovery by Coverity [6] of a number of defects in the Android kernel. This is an encouraging discovery. Bugs in open source software also tend to get fixed immediately, as in the case of the Linux kernel exploit uncovered not long ago.

Quality

In general, open source software gets closest to what users want because those users can have a hand in making it so. It's not a matter of the vendor giving users what it thinks they want--users and developers make what they want and they make it well.

Customizability

Along similar lines, business users can take a piece of open source software and tweak it to suit their needs. Since the code is open, it's simply a matter of modifying it to add the functionality they want.

Freedom

When businesses turn to open source software, they free themselves from the severe vendor lock-in that can afflict users of proprietary packages. With FOSS, on the other hand, users are in control to make their own decisions and to do what they want with the software. They also have a worldwide community of developers and users at their disposal for help with that.

Flexibility

Open source software is typically much less resource-intensive, meaning that we can run it well even on older hardware. It's up to us--not some vendor--to decide when it's time to upgrade.

Interoperability

Open source software is much better at adhering to open standards than proprietary software is. If we value interoperability with other businesses, computers and users, and don't want to be limited by proprietary data formats, open source software is definitely the way to go.

Auditability

The visibility of the code behind open source software, however, means we can see for ourselves and be confident.

Support Options

Open source software is generally free, and so is a world of support through the vibrant communities surrounding each piece of software. Most every Linux distribution, for instance, has an online community with excellent documentation, forums, mailing lists, forges, wikis, newsgroups and even live support chat.

For businesses that want extra assurance, there are now paid support options on most open source packages at prices that still fall far below what most proprietary vendors will charge. Providers of commercial support for open source software tend to be more responsive, too, since support is where their revenue is focused.

Cost

Between the purchase price of the software itself, the exorbitant cost of mandatory virus protection, support charges, ongoing upgrade expenses and the costs associated with being locked in, proprietary software takes more out of our business than we probably even realize. We can get better quality at a fraction of the price.

Try Before We Buy

If we're considering using open source software, it will typically cost us nothing to try it out first. This is partly due to the software's free price, and partly due to the existence of LiveCDs and Live USBs for many Linux distributions.

5.1 Comparative study on open source and commercial software

Table 1 described below is a collaborative list of open-source alternatives to many popular enterprise software needs.

Table 1. Collaborative list of open-source alternatives

S. No	Category	Open Source	Commercial
1	Office Applications	Libre Office / OpenOffice	Microsoft Office
2	Web Browser	Firefox, Chrome	Microsoft Internet Explorer
3	Programming & IDE	Eclipse, KDevelop, Notepad++	Microsoft Visual Studio
4	Diagramming	Dia	Microsoft Visio
5	OS Server	RedHat Enterprise Linux, Canonical Ubuntu Server, CentOS Linux, Novell SUSE Linux	Microsoft Windows Server, UNIX - Sun Solaris, IBM AIX, HP UX
6	OS Desktop	RedHat Desktop / Workstation Linux, Canonical Ubuntu, CentOS Linux, Novell SUSE Linux	Microsoft Windows XP, Vista, 7, 8
7	Enterprise Resource Planning	penERP, Openbravo	SAP, Oracle, Microsoft Dynamics, PeopleSoft, SAGE, SAP
8	Web Application Development	PHP, Zend Framework	Microsoft ASP.Net
9	Relational Databases	MySQL	Microsoft SQL Server, Oracle DB, IBM DB2

10	Cloud Infrastructure	Open Stack	Amazon Web Services, Microsoft Azure, VMWare, Citrix
11	Content Management System (CMS)	Drupal, WordPress, Joomla, Plone	Morello, Vignette/Openext, Interwoven/Autonomy
12	Web Server	Apache web server	Microsoft IIS
13	Middleware - Message Bus	RabbitMQ, JBOSS	IBM MQ, Oracle Weblogic
14	Application Servers	RedHat JBoss, Sun / Oracle Glassfish, Apache Tomcat, Apache Geronimo	IBM Websphere, Oracle / BEA Weblogic
15	Programming Language	Python, Perl, Java	COBOL

6. CONCLUSION

Before making a decision to go for one tool or another based on cost, look at the total cost of implementation, including license costs, set-up costs, training, and any customization. Experience of open source tools is that they tend to provide good depth of functionality but often at the expense of ease of use. This can lead to higher training costs that wipe out the savings made on the licensing. Outside of cost, we might also want to consider value. Commercial software laid most of the foundation for computing since from the beginning, open source has proven to be just as useful and valuable as commercial software has been. With advancements in communication technologies emerging everyday that allow us to communicate and share ideas with one another a lot easier and faster the environment that proprietary software has created for open source allows open source to become very powerful and continue to advance and contribute to computing and software development.

7. REFERENCES

- [1] [online] <http://www.gnu.org/philosophy/categories.html>
- [2] [online] <http://www.openlogic.com/blog/bid/306009/>
- [3] Shane Greenstein, "The Range of Linus' Law", IEEE Computer Society, 2012.
- [4] [online] <http://www.openlogic.com/products/indemnification/>
- [5] Katherine Noyes, "10 Reasons Open Source Is Good for Business", PCWorld, 2010
- [6] "Coverity Scan: 2010 Open Source Integrity Report", <http://www.coverity.com/>