

Open Source Technology : Design Principles, Methodology and Applications - A Literature Review

Srichandana Abbineni

Assistant Professor, IT Department
CVR College of Engineering, Hyderabad, India
e-mail: chandu.abb@gmail.com

Nayani Sateesh

Assistant Professor, IT Department
CVR College of Engineering, Hyderabad, India
e-mail: nayanisateesh@gmail.com

Abstract: Closed source has a great significance in monopoly of software industries. Software users have been witnessing this monopoly of various software companies for the past many years. As an alternative to these closed sources, the concept of Open Source is originated. The term “Open Source“, literally means a consortium of like-minded individuals or corporations who try to come up with products/services that conforms to standards to one specific vertical sector of software industry. In this paper, the major focus is given on various open source technology concepts, development methodology and its significance along with various categories of open source applications.

Index Terms - Open Source, OSI, GNU, FOSS

I. INTRODUCTION

Open source is a development method which harnesses the power of distributed peer review and transparency of process [1]. The main objective of the open source is better quality, higher reliability, more flexibility, lower cost, and an end to predatory vendor lock-in. The emphasis on “openness” in open source software has fostered the growth of a world-wide community of developers contributing to the evaluation and the improvement of various software programs for use in networked servers and desktop systems ranging from operating systems and web servers to various personal productivity tools like word processing, spreadsheets etc. This approach is being considered as a more democratic alternative to monolithic single vendor efforts. In this connection, Open Source Initiative (OSI) was founded in 1997 to promote the open source development activities.

Open development principally encourages an exchange and exploration of new ideas. By enabling and encouraging community development of the software, faster technological advancement is possible. The software also tends to become more reliable as more people can work on and fix the problem in the source code. With this model, it becomes easier for the standards to evolve as development is open, and agreed upon by the parties involved in the development. Interoperability with other software is another benefit of open development as access to the code enables it to be more easily integrated with other programs. The development model [2] shown in figure 1 illustrates how the application development and integration is done in the open environment. In this environment volunteer users are grouped together to form a community to exchange ideas and views towards the development of new software which are non-vendor lock-in. They also contribute the solutions to fix the software bugs identified.

A standard is said to be open when it comprises the following key characteristics [3]:

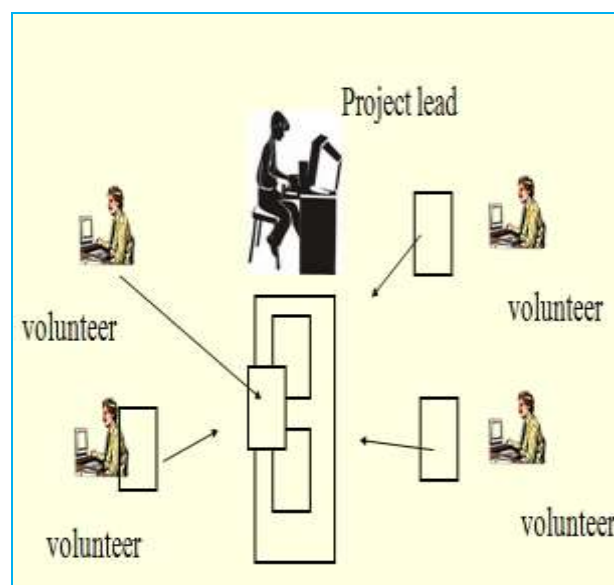


Figure 1: Open Source Development Model

Not controlled by a single vendor – No lock-in to a single vendor's implementation; helps in maximizing end-user choice and making the market more competitive

No Royalties – Free to implement without paying hefty licensing fees or royalties.

Available – Anybody can read and implement the standard.

An Open Technology Standard promotes competition in the marketplace, which benefits the consumer. It prevents a single vendor lock-in and better interoperability. Apart from ability to access the source code, the distribution terms of **open-source software** must comply with the following criterion [3,4,5]:

- **Free Redistribution** - License shall not restrict any party from selling or giving. No royalty to be paid for such cases.
- **Source Code** - Must be allowed for distribution in source code as well as compiled form.
- **Derived Works** - Modifications and derived works must be allowed under the same terms as the license of the original software.
- **Integrity of the Author's Source Code** - Distribution of software built from modified source code must be explicitly permitted by the license.
- **No Discrimination against Persons or Groups** - License should be free from discrimination against any person or group of persons.
- **No Discrimination against Fields of Endeavor** - License must not restrict anyone from making use of the program in a specific field of endeavor.
- **Distribution of License** - No need for execution of an additional license by the parties to whom the program is redistributed.
- **License Must Not Be Specific to a Product** – Supports for redistributed works with the same rights as those that are granted in conjunction with the original software distribution
- **License Must Not Restrict Other Software** - License must not place restrictions on other licensed software in distribution.
- **License Must Be Technology-Neutral** - The license may not be predicated on any individual technology or style of interface.

The following are the motivational factors [6] which are leading the open source technology as the big thing in technology transfer [7].

- Passion
- Educational
- Portfolio
- Status
- Networking
- Need & Influence
- Altruism/Knowledge sharing
- Quality
- Economical

II. OPENSOURCE DESIGN PRINCIPLES

The major design principles [8, 9] involved in the development of Open Source Technologies are:

- Intellectual Property
- Development Paradigm
- Resource Model

These principles are depicted in the following figure 2.

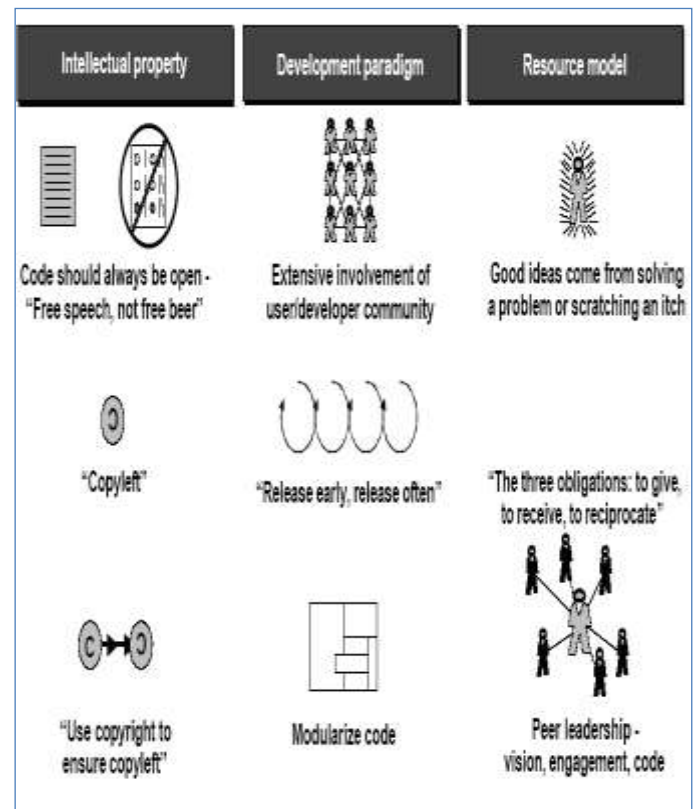


Figure 2: Open Source Design Principles

Users should be treated as co-developers: users should be treated as co-developers as they can access the source code and encourage additions to softwares, bug fixing and documentation etc. It increases the rate at which software evolves.

Early releases: Releasing the first version early helps in finding co-developers early.

Frequent integration: Integrating the new code as early as possible avoids the overhead of fixing many bugs at the end of the project life cycle.

High modularization: Software should be modular allowing for parallel development.

Dynamic decision making structure: There is a need for strategic decisions depending on changing user requirements and other factors.

III. OPEN SOURCE TECHNOLOGY MODEL

In the open source development environment [8], the peer groups or the community group people develop the software and then distribute to the users. The following figure 3 depicts the technology model.

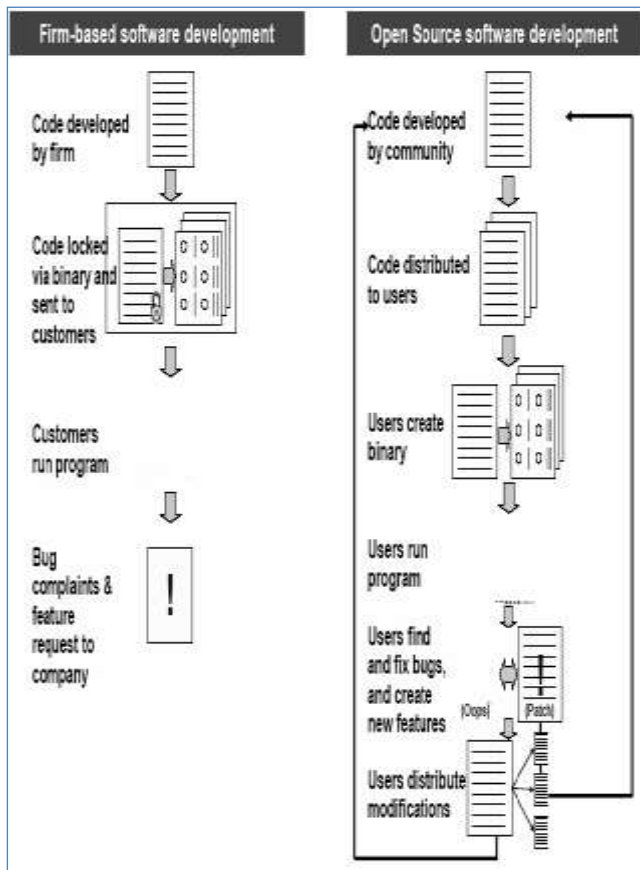


Figure 3: Open Source Technology Model

The users run the applications. If any new requirements are to be added or any bugs to be fixed, the users will work on them. Once these are set, the enhanced software will be redistributed to the community. Since the source code is available and visible to all, everyone can view the code and can make the updates as and when required. For every enhancement and update or bug fixing, the code is thoroughly tested and upgraded by many efficient developers or by the community. This type of development process differs from the firm based or traditional development methodology, where the code will be hidden and the software applications are shared in an executable form (vendor lock-in) or unreadable form. The users who use the program or application have to depend on the supplier whenever there are bugs to be fixed or enhancements to be made.

Open source development environment helps applications de-bugged for free. Releasing new applications to the community helps new generations of developers learn how to code. These new developers create their own applications which are fed back into the community and rapidly advance the growth of the whole system.

Here is an illustration of various Open Source Licenses, Compatibility with the derived works and softwares cover under these licenses [3, 10].

	GPL Compatible	Derivatives Work Licenses	Software under the License
MIT/X	Y	Any	X Window System (X11)
BSD	Y	Any	OpenBSD, FreeBSD
GPL	Y	GPL	LINUX, MYSQL
LGPL	Y	Any with restrictions	Delta3D, OpenOffice
Apache	N	Any	Apache Web Server, Jakarta Tomcat
MPL	N	Any with restrictions	Mozilla Web Browser, Firefox Web Browser
QPL	N	QPL	Old versions of Qt Toolkit and KDE Desktop Environment
Artistic License	N	Any with restrictions	Standard Perl

Table 1: License Compatibility

To understand best how open source technology is developed, we can compare it to traditional closed source software produced by Proprietary companies. Here are the few comparisons between the Open Source Software Development and Commercial Software Development [11]

Open Source	Commercial
No Vendor Lock-in	Vendor Lock-in
Developers are the Users	Developers are the seldom users
Development cost Low	Development cost more
Development is in open environment	Development is in closed Environment
User-developers determine functionality	Product managers determine functionality
Potentially redundant effort	Redundant effort avoided

Table 2: Open vs. Commercial Software development

IV. OPEN SOURCE APPLICATIONS

Nowadays open source technology is integrated into many domains like Artificial Intelligence, Big Data, and Business Intelligence. Numerous open source applications are available as alternatives to proprietary software. The category of the open source applications is broadly divided into the following major categories:

- Operating Systems: Linux, GNU etc.
- Office Productivity: OpenOffice , LibreOffice etc.
- Image Manipulation: GIMP, Pixmap etc.

The following are some of the popular open source projects [12] as on February 9, 2017

Project Name	Category	Sample of Related Companies
Linux	IT Operations	Red Hat, Ubuntu
Git	DevOps	GitHub, GitLab
MySQL	Data & Analytics	Oracle
Node.js	DevOps	NodeSource, Rising Stack
Docker	DevOps	Docker
Hadoop	Data & Analytics	Cloudera, Hortonworks
Elasticsearch	Data & Analytics	Elastic
Spark	Data & Analytics	Databricks
MongoDB	Data & Analytics	MongoDB
Selenium	DevOps	Sauce Labs, BrowserStack
NPM	DevOps	NPM
Redis	Data & Analytics	Redis Labs
Tomcat	IT Operations	NA
Jenkins	DevOps	CloudBees
Vagrant	IT Operations	HashiCorp
Postgres	Data & Analytics	EnterpriseDB
Gradle	DevOps	Gradle
Nginx	IT Operations	Nginx
Ansible	IT Operations	Ansible
Kafka	Data & Analytics	Confluent
GitLab	DevOps	GitLab
Hbase	Data & Analytics	Cloudera, Hortonworks
Chef	IT Operations	Chef*
TensorFlow	Data & Analytics	Google
Cassandra	Data & Analytics	DataStax

Table 3: Open source projects (source: <https://techcrunch.com/>)

V. ADVANTAGES & LIMITATIONS

The following section covers the wide variety of advantages and limitations of the open source development.

A. ADVANTAGES

Flexibility: Software can be used, or modified, to serve user’s desired purposes achieved via modifications or add-ons.

No restriction on use: No contractual restrictions on how the software is being used.

Low cost: There is no charge for the software itself. The major cost is local development; if the number of users is large, users share their efforts and each user’s cost is reduced.

Large Support of Community: Large number of user groups’ support allows problems to be easily solved and bugs to be quickly exposed and fixed.

Scalability and Robustness: A large community of highly skilled software developers is involved in creating open source solutions.

Avoiding lock-in: In vendor lock-in, switching to alternatives are prohibitively high. Vendors may then increase the price of product upgrades or support without greater risk of losing existing customers. It can be overcome in open source development.

Translation: With access to the source code it is easy to translate the language of the software interface.

Bug fixing: When a bug is spotted in the proprietary software, the only people who can fix it are the original developers, as only they have access to the source code. Open source software is different as large number of users can access and change the code; bugs tend to be more visible and more rapidly corrected.

B. DISADVANTAGES/ LIMITATIONS

- Open Source requires a certain learning curve to use and get accustomed.
- Incompatibility issues with 3rd party products may exist.
- Quality assurance process is not widely transparent.
- No financial incentives in this open source development.
- Quality of documentation.

VI. CONCLUSIONS

Open source technology is free for all. It is required to meet ever changing client needs, to avoid proprietary monopoly and to improve World Wide Web services. Open source technology is different from traditional technologies because it is free for all, it is without any proprietary rights and it is more open to all. There is no specific law for open source technology. Each client must keep his own safe guards to protect the technology from external hackers.

VII. FUTURE WORK

Future works to be conducted on Government initiatives for the support of Open Source and its impact on economical growth. Also studies to be conducted on scope and usage of open source applications in various government departments and their ease of use.

REFERENCES

- [1] Riehle, D, "The Economic Motivation of Open Source: Stakeholder Perspectives", IEEE Computer, vol. 40, no. 4, April 2007, pp. 25-32.
- [2] Jim DeGraw, Ray Zado, "Open Source & Research" - A Presentation, Feb 2005.
- [3] Dan Stoner, "Introduction to Open Technology Standards and Open Source Software" - A Presentation, University of Florida, 2003.
- [4] Chris DiBona, Sam Ockman, "Open Sources: Voices from the Open Source Revolution", O’Reilly Publishers, 1999.

-
- [5] Karl Fogell,” Producing Open Source Software How to Run a Successful Free Software Project “, O’Reilly Publishers, 2005.
 - [6] Margit osterloh, Sandra Rota, “Open source development – Just another case of collective invention”, CREMA, 2005.
 - [7] Jamil Alkhatib, Mohab Anis, Hamid Noori,” Open Source: The next big thing in Technology transfer to developing nations”, Proceedings of International Association for Management of Technology-IAMOT ,2008.
 - [8] Jim Herbsleb, A Presentation on “Why Open Source Works”, Carnegie Mellon University.
 - [9] Avijit Gupta, V. SaiSantosh, A Presentation on” Free/Open Source Softwares”, IIITM, Gwalior.
 - [10] Andrew M. St. Laurent, “Understanding Open Source and Free Software Licensing”, O’Reilly Publishers, 2004.
 - [11] Josh Lerner, Jean Tirole, “The Simple Economics of Open Source”, National Bureau of Economic Research, 2000.
 - [12] Dharmesh Thakker, Max Schireson, Dan Nguyen-Huu, "Tracking the explosive growth of open-source software", 7th Apr 2017.