

A Study of Quality Assurance in Open Source Softwares

Navneet Chanalia

Abstract— Open Source Software(OSS) is a software that is developed by volunteer developers for free. It also includes its source code along with the executable file of the software. In this way OSS are free to use and modifiable. But there are some issues and concerns regarding the quality of the open source softwares. So the following literature survey focuses on the various factors affecting the quality of the OSS. And we have also proposed a way to assure quality in the OSS development.

Keywords— open source software, open source software licenses, software quality assurance.

1 INTRODUCTION

Closed Source Softwares are the softwares those are developed by developers working under the rules and regulations of their employers. These employers are various national and multinational software development companies. All these companies have their own predefined software development methodology. So all the software developers working under such companies have to follow a certain set of Software Development Life Cycle(SDLC).

But the same is not true in case of Open Source Software Development(OSSD). There are no predefined SDLC in case of OSSD. Also there is no quality assurance methodology in OSSD. These are the major reasons behind the lack of quality in OSSD.

Most of the software engineering books that one studies during graduation are written in the context of Closed Source Softwares(CSS) and not in the context of Open Source Software Development. So we strongly believe that there should be a study to improve the quality of OSS because they save a lot of money of the end users. A report by the Standish Group states that adoption of open-source software models has resulted in savings of about \$60 billion per year to consumers[1].

2 OPEN SOURCE VERSUS CLOSED SOURCE SOFTWARE

The following table(TABLE 1) depicts some of the major differences between Closed Source Softwares(CSS) and Open Source Software (OSS).

1. Closed source softwares are copyrighted and are developed to gain profit after selling them. On the other hand open source softwares are developed by volunteer developers for free.
2. The best feature of the open source softwares is that they comes with their source code. So that we can view and modify the source code according to our requirements. In this way user can enhance the functionality of an open source software. But same is not true in case of closed source softwares.

3. Another property of open source software is that their new versions are released very frequently. But in case of closed source softwares the end user have to wait longer to receive a working software.

Table 1
Comparison of Open and Closed Source Softwares[2].

PROPERTIES OF CLOSED SOURCE VERSUS OPEN SOURCE		
Type Property	Closed Source	Open Source
Copyright	Yes	No
Sold	Yes	No
Available Source Code	No	Yes
Free Software	No	Yes
Modify Code by Users	No	Yes
Systematic Development	Yes	No
Frequent Release	No	Yes
Openness	No	Yes
Parallel Process	No	Yes
Knowledge Sharing	No	Yes
Private License	Yes	No
Gaining Profit by Company	Yes	No

3 MORE ABOUT OSSD

3.1 List of some popular and successful open source softwares

1. Android.
2. Apache web server.
3. Open Office.
4. Mozilla Firefox.
5. MySql.
6. LINUX.

• Navneet Chanalia is currently pursuing masters degree program in Computer science and applications in Thapar University, India.

3.2 Team structure of open source software development

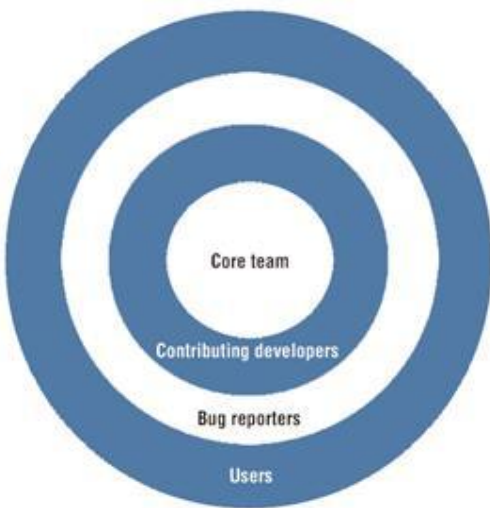


Fig. 1 Onion model for OSSD[3].

Figure 3.1 depicts the basic team structure for any open source software.

Core team: This is the group of OSS developers those take the initiative for developing. The core team is not only responsible for the coding part of the software but also for other project management activities. However, this group is relatively smaller than the group of 'contributing developers' but this group performs majority of the project related tasks. For instance while developing the Apache server 80 percent of the coding was done by a 15 member core development team[4].

Contributing developers: They come in existence after the first version of the software is released in public. After the first release contributing developers joins the development team. Obviously these developers will join the developing team depending on their area of interest in development.

Bug reporters: This is a set of users those not only uses the software but also reports the bug that the encounter while using the software. However this group does not perform any kind of coding but they are very helpful in improving the functionalities of the software.

In other words we can say that bug reporters are the users those perform the beta-testing for the open source softwares. That is why we believe that bug reporters have a major role in the success of open source software development.

Users: The last and the outermost layer of onion model consist of the end users all around the globe. They are the ones who finally uses the open source softwares like- Mozilla Firefox (which is one of the most popular web browser), LINUX (which is considered as one of the secured operating systems), Android (which is an operating system for smartphones and tablet computers), Apache server (which is a well known web server).

Some of these users eventually enter the group of bug reporters and a few of them also joins the development team.

3.3 Open Source Software Licenses

One may wonder what is the need of license for an open source software when it is free and open?

It is true that a software can be made OSS by releasing it uncopyrighted and there after the user of this OSS can view, modify and release it again. But problem will occur if any of the new user of this program releases it as a copyrighted version because in that scenario the new version of the actual software is not open source any more.

To avoid this scenario open source licenses use the concept of 'copyleft' which states that anyone who redistributes the software must pass on the freedom to further copy and change it. So this is one of the reasons why licenses are used with open source softwares.

If a software is released under a OSS license then the software must be used and distributed as specified by the license.

Some of the OSS licenses[5]

Berkeley Software Distribution (BSD)- It was introduced in 1970 and was originally written at the University of California Berkeley. It is used to allow commercial use of the software.

GNU Public License (GPL)- It was introduced in 1991 and it was written by Richard Stallman Source code of OSS under this license can be used, viewed and modified but can not be marketed. It also forbids programs to link to the exclusive rights.

Lesser General Public License (LGPL)- It was published in 1991 by free software foundation. Unlike GPL it allows links to a library or non LGPL/GPL program.

Massachusetts Institute of Technology (MIT)- It was written at the Massachusetts Institute of Technology. MIT permits OSS to be used under exclusive rights.

Mozilla Public License (MPL)- MPL was introduced at the Netscape Communications Corporation and is used for Mozilla Firefox, Mozilla Application Suite, and Mozilla Thunderbird. MPL is a combination of the BSD and GPL licenses.

4 SOFTWARE QUALITY ASSURANCE(SQA)

Roger Pressman[6] states that "Quality assurance consists of the auditing and reporting functions of management. The goal of quality assurance is to provide management with the data necessary to be informed about product quality, thereby gaining insight and confidence that product quality is meeting its goals. Of course, if the data provided through quality assurance identify problems, it is management's responsibility to address the problems and apply the necessary resources to resolve quality issues."

Firstly we have to choose quality standards those are required for our software. Then we have to plan how to achieve these standards. Then we have to frequently access the software to ensure that the software will meet the selected quality standards.

Quality in terms of software means that the software must satisfy all the requirements stated by the customer. So we can say that a quality software is one which meets both requirements and standards.

5 PROBLEMS IN OSSD

Following are the obstacles in the process of achieving quality in OSSD-

There is no fixed team: In case of Closed Source Softwares Development(CSSD) there is a fixed and dedicated team for the project undertaken. But in case of OSSD there is no one hired and paid like in the case of CSS. When there is a fixed team then the project manager can control and lead the team to achieve the desired quality. But in case of OSS team members are not assigned work, instead they choose work. And this thing is obvious in OSS because programmers joins the team by their own wish and interest.

There are no pre-specified product requirements: Unlike CSSD, there are no fixed requirements in case of OSSD. In case of CSSD a customer approaches the development team. Then the requirements of the customer are discussed and finalized. Then a team is assigned to analyze, design, code, test and integrate the final product. But in case of OSSD there is no such formal customer.

However, there is a large number of end users those are finally going to use the OSS, but there is no agreement or bound between the OSS developers and the end users. In case of OSSD programmers defines the requirements. Reason for this thing is - mostly programmers are attracted toward OSSD. Due to this the final product of OSSD exhibit the functionalities from developer's point of view and not from the end user's point of view. So there should be a way to cover this gap between the requirements set by the OSS developers and the actual requirements of the end user. If the OSS developers get ready to adopt some technique to meet actual requirements of the end users then OSS will become more reliable and will exhibit higher quality.

Little project documentation: Project documentation is the process to note down various information about the project during its life cycle. Project documentation not only serves as a record but also helps the new development team members to properly understand the project. It also helps in writing the product manual, which helps the end user in completely understanding the system.

But in case of OSSD there a little project documentation. That is why problems are faced by new developers in the OSS development team and the end users in understanding the OSS. So there should be some team members in the OSS development team those are responsible for documenting the various information about the OSS.

Project often go straight to programming:

In case of closed source software development first of all requirements are gathered and analyzed. After finalizing the requirements the project is designed. Then this design is converted into code.

But in case of open source software development most of the projects go straight to the programming part. So unlike the CSSD, the traditional steps for developing a software are not followed in OSSD. Due to this the overall quality of the OSS is compromised.

The developers in OSSD are the volunteers and they are not get paid for their contribution in the development of an OSS. But some steps must be taken to improve the quality of the

products developed under open source software development.

6 CONCLUSION AND FUTURE WORK

This literature survey concludes that open source development society has got a very powerful tool in the form of a huge base of volunteer developers. And the softwares developed by open source development society are not limited to any set of requirements. And the development team does not stop coding even after releasing the software. They keep on adding the functionalities to the softwares based on the user reviews. Also Open Source Softwares saves around \$60 billion of software users every year[1]. But there are some flaws in open source softwares in terms of quality.

As we know that quality in computer softwares is achieved by fulfilling the customer requirement and quality standards. But in case of OSS there is no formal customer and there are no standards defined for development. So we propose a system to collect requirements from the end users in advance. This system most probably would be a web portal that would be managed by a moderator. The moderator will receive requirements from the end users in a non technical language. Moderator will convert these requirements into technical form and forward them to volunteer developers. The developers will submit there code against these requirements.

The moderator will check the quality using a pre agreed quality tool. The output of the tool will be sent to the end users. This will be called as - quality with proof. Also, only those codes from developers will be accepted which will be above a particular quality level.

So in this manner both requirements and standards will be fulfilled.

REFERENCES

- [1] Rothwell, Richard (2008-08-05). "Creating wealth with free software". Free Software Magazine. Retrieved 2008-09-08.
- [2] Atieh Khanjani, Riza Sulaiman Schools of Computer Science & Industrial Computing Faculty of Information Science and Technology University Kebangsaan Malaysia, 43600 UKM Bangi, Selangor Darul Ehsan, Malaysia.
- [3] Achieving Quality in Open Source Software by Mark Aberdour, opensource-testing.org.
- [4] A. Mockus, R. Fielding, and J. Herbsleb, "A Case Study of Open Source Software Development: The Apache Server," Proc. 22nd Int'l Conf. Software Eng. (ICSE 00), IEEE CS Press, 2000, pp. 263-272.
- [5] A. Beard, H. Kim, "A survey on open source software licenses", Journal of Computing Sciences in Colleges, vol. 22 no. 4, April 2007.
- [6] Software Engineering, A Practitioner's Approach, Roger Pressman, ISBN 0-07-365578-3 (2001).