Open Source Software Development Methodology - A Review

Mayank Joshi

***Abstract*— We are surrounded by computer software's and computer software has made our computing experience more better. Various categories of software are available in market, one of which is Open Source Software (OSS). As the name suggest Open Source Software are those Software whose source code is available for free. Most of the software's available today are open source software. Open Source Software is very popular among various software firms. 78% of software companies are dependent on Open Source Software's.[1] Licensing for Open Source Software are very easy and flexible. Thousands of licensing communities are available for licensing Open Source Software. The main reason for growth fo Open Source Software production is due to its features such as – customization, security and accountability, quality, etc. There is misconception among people that money cannot be made form, but this is completely wrong money can be made from Open Source Software and people are even making money form Open Source Software.**

The landscape of Open Source Software is very large and it is definitely going to increase. It can be easily estimated that in coming years all the software firms will completely relay on Open Source Software's.

***Index Terms*— Closed source software, Development, Licensing, Open Source Software.**

I. Introduction

Open source software is software whose source code is available freely to see, modify, enhance and share. "Source code" is the part of software which is visible only to programmers who created it or who have legal rights to view the code and only they can manipulate the code of software or change the behavior or way of working of the software. A normal software user can never view the source code of software. Programmers who have access to a computer program's source code can improve that program by adding features to it or fixing parts that don't always work correctly. The open-source model, or collaborative development from multiple independent sources, generates an increasingly more diverse scope of design perspective than any one company is capable of developing and sustaining long term. Many of the traditional software companies have tried to take advantage of the free software methodology, not just by using the source code (software), but also by creating advanced business models.

II. Difference between Open Source and Other Type of Software (Closed Source Software)

Some software has source code accessible only to those who has created it or maintains exclusive control over it or who can modify. These kinds of software are called "proprietary" or "closed source" software. Only the original programmer or owner of proprietary software can legally copy, inspect and alter that software, if anyone else found doing any kind of alternation with piece of software will be considered as an illegal act. In order to use closed source software's, the users must agree all their term and conditions (usually by accepting the license displayed at the time of installation of software) that the user will not do anything with the software that is not permitted by owner of software. Examples of proprietary software are – Microsoft Office, Intent Download Manager (IDM). Open source software is totally opposite of closed source software's. The owner of Open Source Software make its source code available free for viewing, copying, altering or sharing. Open source software's are generally safer than closed source software's (propriety software) because the software is open to public, creating a mass collaboration that results in the software being constantly updated, fixed, improved, and expanded on. Mozilla Firefox, Blender are the example of Open Source Software. According to a study Linux (Open Source Operating System) source code has 0.17 bugs per 1000 lines of code while proprietary software generally scores 20–30 bugs per 1000 lines [2]

III. When a Software is Called Open Source Software

Every software cannot fall under the category of open source software for a software to be open source, showing implementation details doesn't mean software is open source software. There are certain standards which the software have to follow to be called an open source software:

1) The implementation of the software must be available freely for viewing, modifications, sharing, and software should be shared in complied and source code form and there must be some source form where source code can be obtained for free. Most popular example is GitHub.

2) The software must be redistributed freely. If anyone sell the software as a part of other software (proprietary software) in that case license must not charge anything to the developer. The developer is free to use it in any way he want.

3) The software must be licensed under any of the various open source licensing community and license must not put any restriction on all other modified or derived work. For example it must not say that is original software is open source then the modified work must be open source or derived work must be licensed under same licensing community*(GNU GPL is an exception, which says derived work should also be licensed under GNU GPL and derived work must be open source)*.

4) The license must not restrict anyone from making use of the program for any other purpose. For example, it may not restrict the program for being used as commercial purpose. Developer can use the software or sell the software in the way he want.

IV. Open Source Development

The development model is adopted for the development for open source software (projects) is shown in fig.1.a. The

Mayank Joshi, Computer Science, Birla Institute of Applied Sciences , Bhimtal, India, 9410167522.

project leader is a person who has designed the foundation of software or a developer who has the initial idea and who implemented it as a software. He can be a single person or a group of person or any institution. He can design the software for personal use or for any institution or for any other purpose.

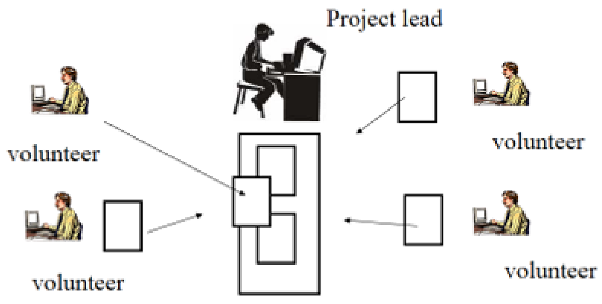

Fig. 1.a: Open Source Development Model [3]

As soon as project leader will release the software meanwhile he will also release the source code of software which will be accessible by other people. If any person want to do future development in software he/she can download the source code and can start development. The person who edits or alter the source code is known as Volunteer. The volunteer can use the source code for fixing further bugs or manufacturing completely new software, or transforming software for personal use etc. It is not compulsory that the new software developed after modification will be open source, it can be open source software or can be closed source or commercial software. It totally depend on volunteer how he choose to publish or release his new modified software.

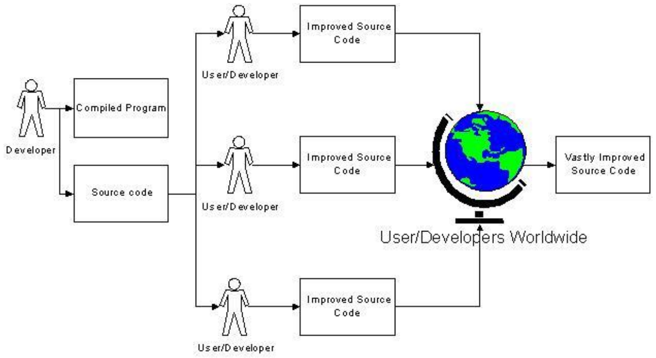

Fig. 1.b: Open Source Development Working [4]

Fig. 1.b: shows the detailed open source development process working. Source code of original software released by the developer is accessed by other user and the user made further changes to the software and release to World Wide Web (Internet) so that it can be available for modification for other user, and the cycle continues in same way. Now two version of software are available in market the original one and the modified one. Now other user who want to contribute has more option to choose between various versions of software.

V. Open Source Licensing and Copyrights

Licensing of open source software is generally done to make copying, modification, and sharing legal. Thousands of different licenses are available, developer can choose any license which he thinks is best suitable for him. Of all the licenses available some of most popular open source licensing communities are:

- Apache License 2.0[5]
- BSD 3-Clause "New" or "Revised" license[6]
- GNU General Public License, version 2 (GPL)[7]
- GNU Library or "Lesser" General Public License (LGPL)[8]
- MIT license[9]
- Mozilla Public License 2.0[10]
- Common Development and Distribution License[11]
- Eclipse Public License[12]

Some of these licenses are very easy to adopt as they say user can easily Some licenses are very easy to adopt because of the flexibility in licensing while some are very difficult to adopt because of certain restriction with licensing such as GNU GPL (or GPL) uses copyleft licensing which means the if original software is released under GNU GPL (or GPL) licensing then the modified or derived software must also be distributed under same licensing term. If derived work is published under any other licensing term then it will be considered as violation of licensing. Linux kernel is example of open source software which is licensed under
GNU GPL licensing.

VI. Rise in Open Source Projects

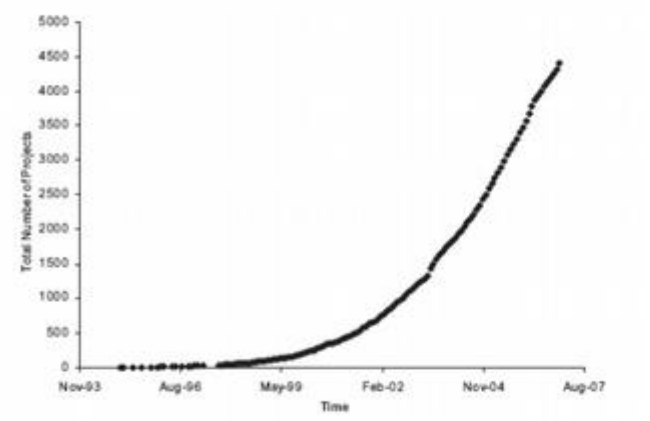

Fig. 2: Inflation in Open Source Projects, 1993 - 2007 [13]

Fig. 2 shows the growth in open source projects between years 1933 – 2007. Before 1999 the figure was very less, total number of open source projects were very less (not even 500) and before 1996 open source was not even very popular.. But after 2002 the number of open source project increased very rapidly and most of these projects were commercial projects. There were various reason for such a rapid increase some of which are listed below:

1) You do not have to re-invent the wheel which means if you are developing a software so you don't have to write the whole code again. You can use the source code of another open source software and modify the source code to fulfill your requirement. Hence it help to save time and decreased software development cost.

2) The quality of open source software is very high as lots of skilled developers are continuously working on it to

increase the quality of software. The software is also free form various vulnerabilities as bugs are continuously fixed by developers and bugs free, more stable versions are released on regular basic. This also make open source software more secure as if there if there is any security issue it can be fixed in further releases.

3) Licensing of open source software is very easy and flexible which can also be considered as the reason of increase in open source software. Unlike closed source or proprietary software, open source software licensing ensures the source code of software is to available for free and can be modified and altered by any developer having source code hence developer do not have to depend on owner for source code.

4) Open source software provide an unique feature known as customizability. It ensures that user can customize the software himself according to his need. If user would like add something or customize he can you can hire skilled software developer to do it for him. Hence open source have no limitation.

5) Some Open source software also provide the feature known try before you use which ensure that user do not have to install the software, he can try the software and if he finds the software is suitable for him than he can install the software. Example of try before you use is Linux which is an open source operating system. You can run Linux Os directly without installing it.

VII. Can money be made from Open source software

There is myth among people that money cannot be made from open source software. But it is totally hogwash, money can be made from open source software. Various ways of making money with open source software are as follow:

1) You can make money by adding specific features or extensions to a software**.** Some companies offer bounties to have specific features implemented in open source software that they use for business functions. Often there is no need for the feature to remain closed source, so significant code is contributed back to the community. In some cases the additional features are required to remain proprietary, but are based on open source software. Hence in both the cases developer have opportunity to earn money.

2) As a developer you can develop an open source commercial or non-commercial software. You have to make the software and source code available for free but you can charge user or institution for its maintenance.

3) You can make the basic version of a software available for free, but you can charge for additional 'premium' features. Cedega(previously known as WineX) provides a reimplementation of the Windows DirectX API under Linux and is released as a combination of free and proprietary code[14]. Gmail for organizations is one example of a service that offers both free and premium options or LinkedIn which also provide both free and premium version.

4) You can earn form advertisement. You can design an open source software, but you can add advertisements to your software, so that when user may use the software he or she may see the advertisement and you can charge money to advertising company. You can also charge the user to remove advertisement by purchasing premium versions (as discussed in 3rd point). This way of earning money is very popular now a days. Software's are available for free but they are asking you to pay to make your software advertisement free.

VIII. Conclusion

Open Source methodology has increased notably nowadays as compared to past and certainly going to increase in coming years due to its various advantages, such as the ability to reduce costs and development time, or to avoid being dependent on a single vendor and availability of source code. Open Source is definitely changing the world of software development. More number of large and small software companies are shifting towards open source. So contributing to open source and becoming an open source developer is a plus point to modern and traditional developers. Many large firms has started various programs to increase the contribution of developers towards open source. For example Google has started Google Summer of Code to encourage the student's developers to participate in open source projects. Hence we can easily conclude that it is worth being an open source developer.

Acknowledgment

I would like to express my sincere gratitude to my advisor Professor Dr. Yugal Joshi, Kumaun University for his support. His guidance help me a lot at the time of researching and writing thesis. I would also like to express thank faculty members form Birla Institute of Applied Sciences for their help. I would also like to express my gratitude towards my friends and family members for their help and support.

References

[1]https://opensource.com/business/15/5/report-future-open-source-survey
[2]http://knowledgebook.ru/articles/articles-33/open-source-software-vs-proprietary-software-16315/
[3]https://tamkeentech.files.wordpress.com/2015/03/picture1.png
[4]https://cippic.ca/sites/default/files/images/figure1.jpg
[5]http://www.apache.org/licenses/LICENSE-2.0
[6]https://spdx.org/licenses/BSD-3-Clause
[7]https://www.gnu.org/licenses/old-licenses/gpl-2.0.en.html
[8]https://www.gnu.org/licenses/lgpl-3.0.en.html
[9]https://en.wikipedia.org/wiki/MIT_License
[10] https://www.mozilla.org/en-US/MPL/2.0/
[11] https://en.wikipedia.org/wiki/Common_Development_and_Distribution_License
[12] https://www.eclipse.org/legal/epl-v10.html
[13] http://15809-presscdn-0-93.pagely.netdna-cdn.com/wp-content/uploads/media/MTIyMzAyNjQ3NzEyNjQ1NzM0.jpg
[14] https://en.wikipedia.org/wiki/Cedega_(software)

Mayank Joshi Computer Science, Birla Institute of Applied Sciences Bhimtal,