IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 3, March 2017

Improvement of the Open Source Software Development Process

Shaikh Shariful Habib

Assistant Professor, Computer Science & Engineering Department, International Islamic University Chittagong

Chittagong, Bangladesh

Abstract: Software development process is a very tightly steps maintaining process. We buy the software by making payment from the different enterprises. It may have different problems such as buying frequently, changing etc. To overcome such type of different problems we very often use open source software. But oss (open source software) has to maintain the proper development process. In this paper I have tried to give some ideas to improve the development process.

Keyword: Development process, peer review, parallel debugging, impede, derived works.

I. INTRODUCTION

Open source software (OSS) attracts the practitioner along with the business and the research communities. The source code of open source software may be freely modified and redistributed with few restrictions. It is produced by loosely organized, ad-hoc communities consisting of contributors from all over the world who seldom meet face-to-face, and who share a strong sense of commitment [1].

The basic principle for the OSS development process (OSSDP) is that by sharing source code, developers cooperate under a model of systematic peer-review, and take advantage of parallel debugging that leads to innovation and rapid advancement [2]. OSSDP (Open Source Software Development Process) can produce software of high quality and functionality. With the development of open source software the activities of software development has become fulfilled.

II. IMPORTANT NEEDS FOR OPEN SOURCE SOFTWARE

Recently, many organizations have started to look towards OSS and OSSDP as a way to minimize their development efforts by reusing open source code and to provide greater flexibility in their development practices [3]. Two factors may impede this growing interest in OSSDP. Firstly OSS (open source software) projects do not typically provide explicit process models, prescriptions, or schemes other than what may be implicit in the use of certain development tools for version control and source code compilation. Secondly, most studies that report on OSS projects like Apache and Mozilla [4, 5, 6] provide only informal narrative descriptions of the overall software development process. Developers who want to join an OSS project must discover its underlying development process by using public information sources on the Web. It could be argued that OSS projects are ingrained in the hacker culture and represents the antithesis of software engineering [7]. The OSS development practices have communities whose members operate with a high degree of autonomy. Process understanding and communication, process comparison, reuse & improvement and process enactment support are important measurement indicator.

The OSI definition [8] includes the following criteria:
− free redistribution: the license shall not restrict any party from selling or giving away the software as a component of an aggregate software distribution; no royalty or fee is required for such sale,
− source code: the program must include source code − derived works: modifications and derived works are allowed, not necessarily subject to the same license as the original work,
− integrity of author's source code: derived works must carry different names or version numbers than the original work,
− no discrimination against persons or groups,
− no discrimination against fields of endeavor,
− distribution of license: no need of any additional license,
− license must not be specific to a product,
− license must not restrict other software,
− license must be technology-neutral.

III. PROCESSES FOR THE OPEN SOURCE SOFTWARE

Two process [9] in the oss are as following:
1) Software development process and
2) Community process

In software development process the following structures can be maintained

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 3, March 2017

a) user (i. download a release, ii. end use, iii. contribute to the project, iv. contribute to the web site)
b) developer (i. develop code, ii. test code, iii. other contributions)
c) manager (i. manage the release, ii. manage the project web site)
d) infrastructure maintainer (i. manage the project code repository, ii. create a build, iii. package a release, iv. manage release accessibility.)
e) committer (i. review code, ii. commit code in the code repository, iii. choose new committers)

In community process the following structures can be maintained
a) Steering committee member (i. project steering, ii. create new projects, iii. manage existing projects)
b) web team member (i. manage the web portal)
c) foundation member(manage the foundation)

IV. TECHNIQUES OF OPEN SOURCE

Open source has unconventional principles such as the distribution of free source code and massive user participation. Open source has certainly introduced a new dimension in large-scale distributed software development.

The open source initiative and its followers propose a software development model that promotes free distribution and 3 complete access to source code [OpenSource, 2002; Wu, 2001].

[Wu, 2001] and [Cubranic, 1999] focus on cooperative work and configuration management to support distributed development. [Cubranic, 1999] discusses major issues of coordinating open source development projects, including collaborative communication mediums and configuration management tools.

The user participation level in open source projects was incredibly high, generating up to 20% of the changes for almost 50% of the projects, and discovering 20% to 40% of the faults in 20% of the projects. Almost 60% of the projects were started to meet the developer's personal needs, later migrating to the community (if that need was common to many users) and growing in size while trying to accommodate an increasing number of features[10].

The use of configuration and bug tracking tools to support collaborative and distributed software development reached approximately 75% of the projects. The open source processes and tools for change management employed by some of the projects definitely seemed to be at the cutting edge of large-scale collaborative software development [11]. In-house testing still takes a considerable amount of time, Mature testing techniques would seem recommendable.

V. IMPROVEMENT OF THE OPEN SOURCE SOFTWARE DEVELOPMENT

My suggestions are the followings:

1. The roles of the community process are very important. Steering committee member should remember that the project which is to be created should completely be new or the developers who are developing the project are hacker or not. Also the committers who review should verify the hacking activity i.e. special secured system should be applied.
2. The release process proposal should be very clear and unambiguous. By the release process developer, manager, infrastructure manager should be chosen in such a way that there has been no hacking created i.e. proper authentication should be maintained.
3. A trust relationship should be preserved between communication process and software development process. These relationship should be verified after a certain period of time to overcome the hacking.
4. There should maintain a ratio of the numbers of members between community process and software development process to maintain a balance and transparency between these two processes. If needed then the numbers should be increased or decreased to preserve the volunteerity.
5. To maintain the transparency of the activities of committers or discarding the hacking activity not only the proper authentication but also second reviewer is needed.

VI. FUTURE WORKS

In future, it should make a more valuable process for developing open source software more transparently and intelligently.

VII. CONCLUSION

Open source software development process is very important, sensitive, watchful eye maintaining issue. As the development process takes place through the world proper maintenance or management techniques should be kept to preserve the acceptability and scalability.

REFERENCES

[1] An Introduction to Open Source Communities, E.E. Kim, Blue Oxen Associates, Technical report, BOA-00007, 2003.
[2] Results from Software Engineering Research into Open Source Development Projects Using Public Data, S. Koch, and G. Schneider, Diskussionspapiere zum Tätigkeitsfeld Informations verarbeitungund Informations wirtschaft, Hansen, H.,R., and Janko, W., 22, Wirtschafts universität Wien, Austria, 2000.
[3] Reusing Open-Source Software and Practices: The Impact of Open-Source on
[4] Commercial Vendors, A. W. Brown, and G. Booch, ICSR-7, LNCS 2319, Springer-Verlag, 2002, pp. 123-136.

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 3, March 2017

[5] A Case Study of Open Source Software Development: The Apache Server, A., Mockus, R.T. Fielding, J. Herbsleb, 21st International Conference on Software Engineering (ICSE), Los Angeles, CA, 1999, pp 263-272.
[6] Two case studies of open source software development: Apache and Mozilla, A. Mockus, R.T. Fielding, and J. Herbsleb, 2002. ACM Transactions on Software Engineering and Methodology, 11(3), ACM, 2002, pp 309-346.
[7] An Overview of the Software Engineering Process and Tools in Mozilla Project, C.R. Reis, and R. Pontin de Mattos Fortes, Workshop on OSS Development, Newcastle upon Tyne, UK, 2002, 162-182.
[8] A Descriptive Process Model for Open-Source Software Development, Johnson, K., Master Thesis, Univ. Calgary, Alberta, 2001.
[9] The open source Definition Version 1.9 (on line), OSI, http://www.opensource.org, 2003.
[10] Chapter 1 OPEN SOURCE SOFTWARE DEVELOPMENT PROCESS MODELING Jacques LONCHAMP LORIA, BP 254, 54500 Vandoeuvre-les-Nancy, France (jloncham@loria.fr)
[11] Raymond, 1999] Raymond, E.S. 1999. Linux and Open-Source Success, IEEE Software, Vol. 16, No. 1, 85 - 89.
[12] Mozilla, 2002] Mozilla. 2002. Mozilla tinderbox framework, http://tinderbox.mozilla.org.

DOI 10.17148/IJARCCE.2017.6340