

Open Source Software Maintenance Process Framework

Timo Koponen
Department of Computer Science
University of Kuopio
Box 163, 70211 Kuopio, Finland
+358-17-162388
timo.koponen@uku.fi

Virpi Hotti
Department of Computer Science
University of Kuopio
Box 163, 70211 Kuopio, Finland
+358-17-162105
virpi.hotti@uku.fi

ABSTRACT

To identify the Open Source maintenance process two well known Open Source projects Apache HTTP server and Mozilla web browser were studied. The Open Source software maintenance process is formal even anyone can submit modifications or defect reports to Open Source software projects. We assume that the Open Source maintenance process is similar to the maintenance process defined by the ISO/IEC. In the case studies, four activities were found similar to the activities of the ISO/IEC Maintenance process. This paper presents the Open Source maintenance process framework. The framework is exemplified with the ISO/IEC Maintenance process framework.

Keywords

Open Source, Maintenance Process, Apache, Mozilla

1. INTRODUCTION

This paper is study of the Open source maintenance process from viewpoint of the ISO/IEC maintenance process. The objective was to identify the Open Source maintenance process and find correlation between the Open Source maintenance process and the ISO/IEC Maintenance process. Previously Open Source software engineering including maintenance is researched from the viewpoints of the development, project and social community [3, 8, 17]. The other studies have presented development models by specifying the roles [17]. Research has also showed lack of the maintainability [16]. The Open Source software processes are studied at viewpoint of the software development [17].

Two Open Source projects, Apache and Mozilla, were analyzed. They were selected because they are succeeded and well-known. Furthermore, they are used as sample cases in the other Open Source software studies [4, 8, 9]. The cases were analyzed using publicly available data such as projects' homepages, mailing lists, defect and version management systems.

Section 2 will present the ISO/IEC Maintenance process. Section 3 will present the case studies and Open Source maintenance process framework. Section 4 will exemplify the Open Source maintenance process framework with the ISO/IEC maintenance process.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Open Source Application Spaces: Fifth Workshop on Open Source Software Engineering (5-WOSSE) May 17, 2005, St Louis, MO, USA.

Copyright 2005 ACM 1-59593-127-9 ... \$5.00.

2. ISO/IEC MAINTENANCE PROCESS

The ISO/IEC Maintenance process model is presented in the SWEBOK, ISO/IEC 12207 and ISO/IEC 14764 standards [5, 6]. Alternative descriptions of the maintenance process are in the ISO/IEC 15288 and de facto standard ITIL (IT Infrastructure Library). Their viewpoints are for the system life cycle instead of the software life cycle [7, 13, 14].

The ISO/IEC Maintenance process provides required operations, their inputs and outputs. Most of the maintenance activities are similar to the software development activities. A request has to be tracked during the identification, analysis, implementation, review, and acceptance. The ISO/IEC divides maintenance tasks to six activities (Figure 1). [5]

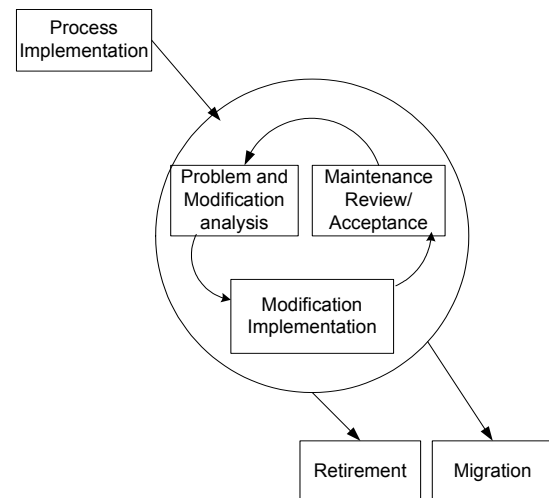


Figure 1. ISO/IEC Maintenance process activities [5]

The pre-delivery activities include planning for the post-delivery actions - process implementation. It also includes an implementation of the configuration or version and request management process.

The post-delivery activities include problem and modification analysis, software modification, maintenance review, acceptance, and training and operation of the help desk. It also includes retirement and migration. [5]

The activities are divided to the tasks that are shown in the table 1.

Table 1. Activities and tasks of the ISO/IEC maintenance process [5]

Activities / Tasks
Process implementation
Develop maintenance plans and procedures
Establish procedures for Modification Requests
Implement the Configuration Management Process
Problem and modification analysis
Perform initial analysis
Verify problem
Document the results
Develop options for implementing the modification
Obtain approval for modification option
Modification implementation
Perform detailed analysis
Develop, code, and test the modification
Maintenance review/acceptance
Conduct reviews
Obtain approval for modification
Migration
Develop a migration plan
Notify users of migration plans
Conduct parallel operations
Notify user that migration has started
Conduct a post-operation review
Ensure that old data is accessible
Retirement
Develop a retirement plan
Notify users of retirement plans
Conduct parallel operations
Notify user that retirement has started
Ensure that old data is accessible

For example, the activity Process implementation has three tasks. All the activities and tasks are taken in to considerations in section 4.

3. OPEN SOURCE CASE STUDIES

Open Source software is a software product that is licensed with designated Open Source licenses. Open Source licenses give to the users' source code and rights to modify and redistribute the software. Usually these are only rights of the copyright holder. [8] The Open Source licenses are approved by the Open Source Initiative (OSI). It has defined requirements of the Open Source license in the Open Source Definition (OSD) [16].

The Open Source project community has a hierarchical organization model. The members of community are divided to the three major groups: core members, developers and users.

The core members are initiators or major contributors of the project. They manage the project and their contribution is major part of the work. They may be managers of the modules or subprojects. The amount of core members is small - from 1 up to number of the modules.

The developers participate to the development of software. Participation can mean code contribution, documentation or other activity. The users group is larger than the group of the core members but significantly smaller than the group of the users. The users are users of software product of project. They are largest of the three groups. [2, 17]

Subsection 3.1 presents Apache and subsection 3.2 Mozilla. Subsection 3.3 concentrates to the maintenance of Apache and Mozilla. The main aim of subsection 3.4 is found out the tasks of the maintenance process.

3.1 Apache HTTP Server

Apache HTTP Server is well known Web server. It is licensed with the Open Source Initiatives approved license - Apache License 2.0. The first version of Apache was released at December 1995. Now it has over ten million users. The project is led by Apache Foundation that owns trademark to the product and the project has multiple associates such as IBM. The Apache project uses CVS as a version management system and Bugzilla as a defect management system. [2, 9]

According to Netcraft [12], Apache is used in over 70 percent of web servers and it has over near 40 million users. [8] In the Apache project over 3000 different people has submitted problem reports. Furthermore, nearly different 400 people submitted code to the project. The project has 50 core members that are specially mentioned. [9]

The plans of the Apache project are presented in the homepage of the project. The plans will present general guidelines of the project. The actual plans are discussed in the mailing lists of the project. The defect management and version management processes bases on use of the Bugzilla and CVS systems. [1]

3.2 Mozilla

Mozilla web browser project was launched at January 1998, when Netscape released the source code of the Netscape web browser. Now Mozilla has millions of users. Mozilla is licensed with the Mozilla Public License (MPL). The project is led by the Mozilla Foundation. It has multiple associates such as American Online (AOL). The project uses CVS as a version management system and Bugzilla as a defect management system. [8, 10]

According to the OneStat [15] over 7 percent of the web users, use Mozilla as a web browser. Almost 7000 people has submitted problem reports to the project. Furthermore, near 500 people has submitted code to the project. [9] The core developers are responsible for one or more modules as a module owner. There are about 60 different core developers in the project. [10]

The Mozilla project has a roadmap that shows planned maintenance releases [11]. The Mozilla project's version management and defect management processes bases on usage of the Bugzilla and CVS systems. [9]

3.3 Maintenance in Projects

In the Apache project anyone of the users can submit modifications or submit defect reports. Some of the problems are

assigned to the core members that have unofficial specialty area in the project. The developers perform tests before submitting the modification. A steering group reviews all the modifications before acceptance. The steering group is formed from the core members. A new version of the software is released, when the core member, who is assigned as a release manager, expresses a release request and the other core developers accepts release request. In the release, the software acquired and packaged from the version management system and released as installation files. [9]

In the Mozilla project anyone can submit modifications or defect reports to the defect management system, Bugzilla. Most of the problems are assigned to the developers. The developers perform the tests before submitting the modification to the project. Tests can be such as compiling and few test cases. The software is compiled daily and the "smoke tests" are done by the test teams. All changes are inspected before acceptance. The inspections are done by the module owner and the one of the super reviewers. Accepted modifications are taken to the version management system, CVS. The new version of the software is released when a milestone of the roadmap is reached. In the release, software is packaged from the version management system and released as installation files. [9]

3.4 Maintenance Activities and Tasks

Both projects, Mozilla and Apache, have two pre-delivery tasks Implement defect management system and procedure; and Implement version management system and procedure.

The post-delivery actions are planned as a form of the roadmap. The procedures for the modification requests were implemented as a form of the defect management system. The version management process was integrated to the version management system that was in use. The post-delivery tasks that were identified are presented in table 2. Furthermore, the bonds between the tasks were found and they are also presented in the table 2.

Table 2. Identified tasks post-delivery and correspondence matrix

Tasks	A	B	C	D	E	F	G	H	I	J	K	L	M
Problem initial analysis (A)	S	c	c	c									
Problem reporting (B)	c	S	c	c									
Problem verification (C)	c	c	S	c				c					
Problem documentation(D)	c	c	c	S				c					
Problem assign (E)					S	c		c					
Implementation (F)					c	S	c	e					
Submit modification (G)					c	c	S						
Discussion(H)								S	c				
Review (I)								c	S				
Acceptance (J)								c	c	S			
Pre-release testing (K)											S	c	
Packaging (L)											c	S	c
Announcement (M)												c	S

S - Strong correlation

c - correlation

The tasks A, B, C and D form the first activity Problem and modification analysis that includes problem analysis, reporting, verification and documentation. The tasks A, B, C and D correlate to each other because they may cause a circular or overlapping process and an information flow. For example, the Problem reporting (B) forms a problem ticket after the problem is verified in the Problem verification (C).

The tasks E, F and G form the second activity Implementation that includes problem assignment, implementation and its submission. The correlation is obvious because assigner and implementer can be same person and the implementation can be only sign of the assignment.

The tasks G, H, I and J form the third activity Modification review and acceptance that includes modification review, discussion and acceptance. The acceptance is related to the discussion and review, because it is done usually voting by the core group's members. The acceptance can be done by the module owner or subproject manager in the minor cases.

The tasks K, L and M form the fourth activity Release Management that includes pre-release testing, packaging and release announcement. These tasks are usually done together by the same person or group. They form a group of tasks that are done sequentially and all at the time.

Table 3. Activities and tasks of the ISO/IEC maintenance process [3]

Activities / tasks
Process Implementation
Implement Defect management system and procedure
Implement version management system and procedure
Problem and Modification Analysis
Initial analysis
Verification
Documentation
Implementation
Assign
Implement
Submit
Modification review and acceptance
Discuss
Review
Accept
Release Management
Pre-release testing
Packaging
Release announcement
Retirement

All thought, almost every task has correlations to discussion (H). It is grouped with the acceptance because it is in major role. It could as well be grouped to the problem and modification analysis or implementation.

Activities and tasks are presented in the table 3. They are explained with more details in the section 4

4. OPEN SOURCE SOFTWARE MAINTENANCE PROCESS FRAMEWORK

The Open Source software maintenance process framework structure is guide lined by the ISO/IEC maintenance process and it is result of the evaluation of the Mozilla and Apache case studies. The framework is presented in the table 4 parallel to the ISO/IEC maintenance process activities and tasks. The task guidelines are presented after the table.

The maintenance plans should be developed and updated. The plans can be informal, as a road map of the project. They should be updated during the projects' lifecycle.

The implement defect management system and procedure and establish procedures for modification requests activities are similar to the ISO/IEC model. In the Open Source projects, the procedures can be based on the defect management system, such as Bugzilla. The procedures should also include information about the review and acceptance of the modification. The defect management system gives procedures to report and assign the problem reports and modification requests. Furthermore, the defect management system provides flexible possibilities to the tracking and controlling because it logs reports and requests, and produce statistical information and analysis.

The implement version or configuration management system and procedure activity correspond to implement the configuration management process in the ISO/IEC model. In the Open Source projects, the procedures can be based on the configuration management system, such as CVS. They should also support distributed software development. [8]

The initial analysis, verification and documentation activity users detects problems or modification needs. The users should be required to do initial analysis of the problem before reporting the problem or request. The problem report should express classification of the report, such as a problem or an enhancement, reproducibility and instructions of the problem, the version of the software and the environment. The initial analysis produces initial documentation of the modification as an outcome.

Eric Raymond has presented [17] that Open Source software development has two basic implementation models, the bazaar and the cathedral. In the cathedral model projects, the implementation task is given to a developer who has developed that a part of the software. In the bazaar model projects, the user or the developer that reports the problem can implement the modification. Usually the projects are mixture of both models.

Before submitting modifications, the developer should test modification locally. After the local testing, the developer can submit modification to the project. Modification submissions should use the defect management system or the version management system depending of the status of the submitter.

The discussion, review and acceptance tasks precede applying to the project's source code. Approval for the modification is obtained from the core developers. Suitability and correctness of

Table 4. Open Source software maintenance and ISO/IEC maintenance processes

ISO/IEC Maintenance Process	Open Source software maintenance process
Activities / Tasks	Activities / tasks
Process Implementation	Process Implementation
Develop maintenance plans and procedures	
Establish procedures for Modification Requests	Implement Defect management system and procedure
Implement the Configuration Management Process	Implement version management system and procedure
Problem and Modification Analysis	Problem and Modification Analysis
Perform initial analysis	Initial analysis
Verify problem	Verification
Document the results	Documentation
Develop options for implementing the modification	
Obtain approval for modification option	
Modification Implementation	Implementation
Perform detailed analysis	
	Assign
Develop, code, and test the modification	Implement
	Submit
Maintenance Review/Acceptance	Modification review and acceptance
Conduct reviews	Discuss
Obtain approval for modification	Review
	Accept
Migration	Release management
Develop a migration plan	
Notify users of migration plans	Pre-release testing
Conduct parallel operations	Packaging
Notify user that migration has started	Release announcement
Conduct a post-operation review	
Ensure that old data is accessible	
Retirement	Retirement
Develop a retirement plan	
Notify users of retirement plans	
Conduct parallel operations	
Notify user that retirement has started	
Ensure that old data is accessible	

the modification could be discussed with the other developers. If the modifications to same problem are implemented concurrently, the project manager must choose.

The release management activity should include versioning, pre-release testing and packaging. The versioning should produce a unique version number to the each release. New versions are tested before releasing. The pre-release testing criteria should be established. The pre-release testing should include at least compiling tests if delivered into the source code, and a set of test cases. The announcement about the new version (migration) to the users should be done via project's homepages, mailing lists and news.

Open Source projects does not have a retirement procedure. The project should announce when the development of the software

version or the software is ending. The project should leave the source code of the software available that accessibility of data remains.

5. CONCLUSIONS

In this paper, the maintenance processes of the Apache and Mozilla were studied. Several tasks were found from the maintenance processes of the Apache and Mozilla. Found tasks were identified and grouped into the activities.

Activities were divided to pre- and post-delivery tasks as the ISO/IEC Maintenance standard proposes. We identified two pre-delivery and 13 post-delivery tasks. By using correspondence matrix the tasks were gathered to the five activities that were the process implementation, problem and modification analysis, implementation, modification review and acceptance and release management.

Furthermore, the Open Source maintenance process framework was created and compared with the ISO/IEC Maintenance process. The ISO/IEC Maintenance process has six activities. Four of them were similar to the activities that were found in the case study. The fifth activity, the release management, replaced the migration of the ISO/IEC process, because the new versions are only released not migrated. Furthermore, the sixth activity, the retirement that was presented in the ISO/IEC Maintenance process but it was not found from the Apache and Mozilla.

6. REFERENCES

- [1] Apache: *Apache HTTP Server Project Guidelines and Voting Rules*. Apache Foundation, 2004.
<http://httpd.apache.org/dev/> (10.11.2004)
- [2] Apache: *About the Apache HTTP Server Project*. Apache Foundation, 2004.
http://httpd.apache.org/ABOUT_APACHE.html (11.11.2004)
- [3] Capiluppi A., Lago P., Morisio M.: Characterizing the OSS process. *Proceedings of International Conference on Software Engineering, 2nd Workshop on Open Source Software Engineering*, Orlando, Florida, May 2002.
- [4] Erenkranz, J.: Release Management Within Open Source Projects. *Proceedings of International Conference on Software Engineering, 3rd Workshop on Open Source Software Engineering*, Portland, Oregon, February 2003.
- [5] IEEE: *Guide to the Software Engineering Body of Knowledge (SWEBOK)*. IEEE Computer society 2001, Los Alamitos, California.
- [6] ISO/IEC: *ISO/IEC 12207:1995/Amd 2002: Software Engineering: Software life cycle processes*. ISO/IEC 2002.
- [7] ISO/IEC: *ISO/IEC 15288:2002: Software Engineering: Software life cycle processes*. ISO/IEC 2002.
- [8] Koponen T., Hotti V.: Evaluation Framework for Open Source Software. *Proceedings of The 2004 International MultiConference in Computer Science and Computer Engineering*, Las Vegas, Nevada, June 2004
- [9] Mockus A., Fielding R., Herbsleb J.: Two Case Studies of Open Source Software Development: Apache and Mozilla. *ACM Trans. Software Engineering and Methodology*, 11(3), 309-346, 2002.
- [10] Mozilla.org: *Homepage of the Firefox web browser, Thunderbird and Mozilla Suite*. Mozilla.org, 2004.
<http://www.mozilla.org/> (10.11.2004)
- [11] Mozilla.org: *Mozilla.org development tools*. Mozilla.org, 2004. <http://www.mozilla.org/tools.html> (10.11.2004)
- [12] Netcraft: *Netcraft*. Netcraft Ltd 2004.
<http://www.netcraft.com/> (25.11.2004)
- [13] Office of Government Commerce: *Infrastructure management*. ITIL, London, Stationary Office, 2002.
- [14] Office of Government Commerce: *Application management*. ITIL, London, Stationary Office, 2002.
- [15] OneStat: *Press Box*. OneStat 2004.
http://www.onestat.com/html/aboutus_pressbox34.html (25.11.2004)
- [16] Open Source Initiative: *The Open Source Definition*. Open Source Initiative 2003,
<http://www.opensource.org/docs/definition.php> (6.8.2003).
- [17] Raymond E. S.: *The cathedral and the bazaar*.
<http://www.catb.org/~esr/writings/cathedralbazaar/>, 2000.
- [18] Rothfuss, G.J.: *A Framework for Open Source Projects*. Master Thesis in Computer Science. Departement of Information Technology, Universität Zuurich, 2002.
- [19] Schach, S.R., Offutt, A. J.: On the Nonmaintainability of Open-Source Software. *Proceedings of International Conference on Software Engineering, 2nd Workshop on Open Source Software Engineering*, Orlando, Florida, May 2002.