

Available online at [www.sciencedirect.com](http://www.sciencedirect.com)**ScienceDirect**

Procedia Computer Science 35 (2014) 1711 – 1717

**Procedia**  
Computer Science18<sup>th</sup> International Conference on Knowledge-Based and Intelligent  
Information & Engineering Systems - KES2014

## A study on maturity model of open source software community to estimate the quality of products

Yoshitaka Kuwata<sup>a\*</sup>, Kentaro Takeda<sup>b</sup>, and Hiroshi Miura<sup>c</sup><sup>a</sup> *Muroran Institute of Technology, Japan*<sup>b</sup> *NTT DATA, Inc., USA*<sup>c</sup> *NTT DATA CORPORATION, Japan*

---

### Abstract

Open Source Software (OSS) is a new paradigm to develop software by community, in which groups of developer collaborating each other. The method becomes very popular because the products of OSS projects, such as source code, documents, results of tests, are published by open-license. One of noticeable features of OSS is openness of project. Anyone can access the products of OSS projects. It is expected that the quality of products could be higher than those developed by conventional methods, as more people access OSS products and the chance to find defect could be larger. However, there are no established methods for the evaluation of OSS, neither actual terms of the evaluation of OSS. We propose an evaluation method, which is based on the maturity model of OSS development community.

© 2014 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/3.0/>).

Peer-review under responsibility of KES International.

*Keywords:* Evaluation of Software; Maturity Model of Open Source Software; Open Source Software Community; Collaborative Software Development

---

### 1. Introduction

Open Source Software (OSS) is a new paradigm to develop software by community, in which groups of people collaborate each other to develop software. These activities are often called OSS projects. The products of OSS projects include source code, documents of software, and test results. In many OSS projects, their products are published under open-license. Anyone can freely access these OSS products.

The definition of OSS is described in reference [1]. The following is a list of important feature of OSS in practice.

- A) The process of software development, as well as organization of the project, is open. Anyone can join the project to discuss the specification of software, develop source code, test the code, write documents, and so forth.
- B) The products, such as source code, documents of software, test results, are open and available on repositories in the Internet.

---

\* Corresponding author. Tel.: +81-143-46-5893; fax: +81-143-46-5899.  
*E-mail address:* [kuwata@mmm.muroran-it.ac.jp](mailto:kuwata@mmm.muroran-it.ac.jp).

- C) Wide variety of OSS projects exists, which include Operating Systems, Libraries & Frameworks, and Application Programs. The size of the software also varies.
- D) The size of OSS project varies from one-person project to several thousands.
- E) The status of OSS project is different; i.e. one project is just started, another is in commercially used, and another is obsolete.
- F) Projects have their own license policy. One project releases their products under Public Domain (PD). Another project adopts Gnu Public License (GPL) [2]. Note that some combinations of license can be incompatible with each other. The software with these licenses cannot be merged in the same conditions.
- G) The quality of products is different. One project releases programs with commercial quality and another project produces poor quality.

As the products from OSS projects are open and the source code have chance to be examined by many people, it is likely that the defect of the code is corrected. As the result, it is expected that the quality of OSS becomes higher than closed source software. In general, license fee is not required for use of OSS products. Therefore it is cost effective to adopt OSS for integrations into computer systems.

On the other hand, management policy of OSS, such as testing, quality assurance, and validation, depends on the project. Therefore, it is very difficult to evaluate the quality of OSS in one criteria. Even if the quality of an OSS is known, risks still exist that the software violates IPR (Intellectual Property Right) of third parties. Furthermore, as software needs to be continuously maintained while they are in use, the continuity of the OSS development is an important issue.

In this paper, the authors assume the cases that one needs to choose OSS for commercial systems. Especially we believe the continuity of OSS is very important for the decision. In this paper, we propose a method to evaluate OSS that is based on maturity of OSS community.

## 2. Goal of our study

The goal of our study is to estimate the quality of products produced by OSS community. The result is used to make a decision on adopting the software or not. We start defining a set of metrics for the evaluation, followed by a framework for the evaluation of quality. As the functions of software depends on the purpose in use, there would be no common criteria exist for evaluation. Thus, we focus on non-function requirement.

The evaluation of the following three items is our goal.

- Quality of software
- Continuity of software development and maintenance
- Expandability of software, which is measured by how modular the software components are designed.

## 3. Previous research on OSS community

Because OSS is open, it is easy to access their development processes, record of development, documents, source code of software, and record of communication. Many studies exist in the field of software engineering and science.

One example of analysis on OSS communities is found in the study of Tai<sup>[3]</sup>. They tried to estimate release timing of software based on the records of the source code management system.

Ohkura<sup>[4]</sup> tried to evaluate the activities of OSS project based on the record of mailing list used to develop OSS. Tamura<sup>[5]</sup> used neural network technique to learn the importance of software components in a system and compare them to the faults recorded in issue tracking system.

In previous research shown above, the focus of the study is the products of OSS project. They don't consider the activities of OSS projects. But recent services such as RepOSS<sup>[6]</sup>, Ohloh<sup>[7]</sup>, are gathering activities of OSS projects indexed by products. They are targeted to provide measure of OSS maturity level and developer level to enterprises who want to use OSS and/or hire OSS developers.

In this study we mainly focus on the process and activities of OSS community to estimate the quality of their products. We propose model of OSS community for the evaluation of their products.

## 4. Our approach

### 4.1. Relationship models of OSS community

We analyzed structures of OSS community. We propose a relationship model of OSS shown as Fig. 1.

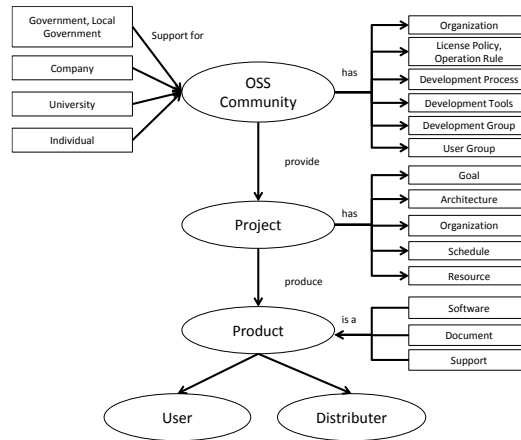


Fig. 1. Relationship model of OSS

In this model, an OSS community provides a project. The project has its own goal, architecture, organization, and schedule. Products such as software, document, and maintenance are produced by the project. We focused on developer groups, who have roles in production quality and maturity in this model, even though an OSS community also consists of user groups, Several variations of this model are possible, which are shown as Fig.2.

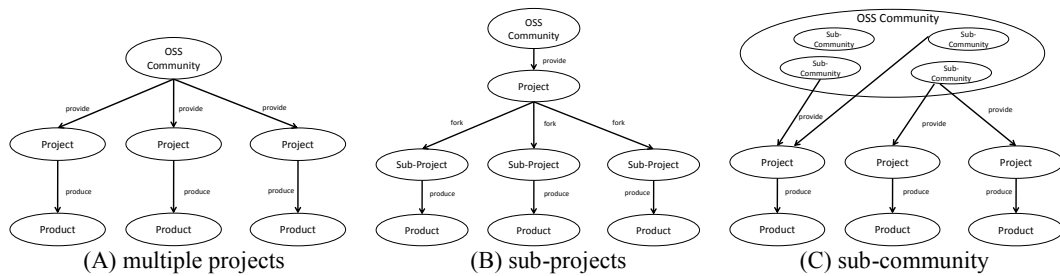


Fig.2 Variances of relationship model of OSS

In case (A) shown on the left of Fig.2, OSS Community provides three projects. Case (B) on the center of Fig.2 represents that a project consists of sub-projects. Case (C) on the right of Fig.2 shows that OSS community includes several sub-communities. One example of sub-community is plugin developer community in Mozilla developer network.

4.2. Maturity model of OSS community

We propose a new maturity model of OSS community based on capability maturity model (CMM) described in [8]. The following table shows the model for OSS community.

Table 1. Maturity model of OSS Community

Level	Status	Definition
1	Initial	A project is just started. Community is not formed at this stage.
2	Managed	Development is carried out by a project. At this stage, products are released. Although a community is organized, there is no explicit rule to manage projects and products.
3	Defined	A community defines development processes and rules to manage their projects and products. According to the definitions, projects are carried out. The projects continuously release products.
4	Quantitatively Managed	A community quantitatively measures the status of projects and products. For example, the community determines the release of products by checking the quality of them.
5	Optimizing	A community quantitatively measures the status of projects and it continuously improves the process

## 5. Examples analysis of major OSS community

### 5.1. A case of The Linux Foundation

Linux is one of the most popular OSS operating system, which is developed by The Linux Foundation. The following table shows an overview of the community based on [9].

Table 2. Overview of Linux Foundation

Organization	The Linux Foundation
Start date	1991
Number of projects	The system has about 100 subsystems
Numbers of developers	more than 5000
Software	1.5M Line of source code. Git is used for repository. GPL2
Management	System is divided into 100 subsystems. Each subsystem has a group of maintainers, who are responsible for the release. Staging release policy.

Note that Linux is very large software, and Linux community is also large. More than 5000 developers are involved to the project. Linux is developed by 100 sub-projects which is corresponds to variant (B) in Fig.2. Each sub-project has a group of maintainers, who are responsible for the management of their part as well as the source code. The maintainers decide how to change the source code and when to release them.

Linux community quantitatively measures the status of the projects by using test suites, and which corresponds to level 4 in our maturity model of OSS community. The community keeps improving their way of management. For example, the Linux Foundation has Technical Advisory Board, which role is described in [10].

*“The Technical Advisory Board (TAB) provides the Linux kernel community a direct voice into The Linux Foundation’s activities and fosters bi-directional interaction with application developers, end users, and Linux companies.”*

The activity corresponds to level 5.

### 5.2. A case of Free Software Foundation

Free Software Foundation (FSF), originator of the Free Software, is often recognized as one of the kinds of the oldest OSS communities. FSF develops software tools such as editors, compilers, and fundamental libraries. The following table shows an overview of the community based on [11].

Table 3. Overview of Free Software Foundation

Organization	Free Software Foundation
Start date	1985
Numbers of project	364 (Number of GNU package)
Numbers of developers	unknown
Software	Depend on package
Management	Each package has a group of maintainers, who manage source code. Procedure to manage package is defined as “Information for Maintainers of GNU Software.” Source code is maintained in repositories built by Git and Savannah. Hydra is used to check portability of the code. Volunteers run test process. There is no description of release condition of software.

FSF supports 364 sub-projects. Each sub-project develops a set of software called “GNU package.” The structure of FSF is corresponds to (A) in Fig.2.

FSF leaves the management of GNU package to maintainers, who are assigned for each package. As means and processes of management is documented, the community corresponds to level 3 in our maturity model. As the actual management of GNU package is left for maintainers, the level varies by project. A repository system is used for the management of source code, which makes it possible for maintainers to quantitatively measure the status of code. However it is not clear how these results are

used for the management of the projects, as the management is left for maintainers. We regard the level of community as level 3 or level 4, depending on the project. We found management of some project is less than level 3.

### 5.3. A case of the Apache Software Foundation

Apache Software Foundation (ASF) is also one of the oldest OSS communities. ASF started a project to develop a web server and it is supporting about 200 projects now.

The following table shows an overview of the community based on <sup>[12][13]</sup>.

Table 4. Overview of Apache Software Foundation

Organization	The Apache Software Foundation
Start date	1985
Numbers of project	199
Numbers of developers	unknown
Software	Depend on projects
Management	Project Management Committee (PMC) manages each project. Each project controls their code and release schedule. The following URL represents the example of Apache Ant Project; <a href="http://ant.apache.org/bylaws.html">http://ant.apache.org/bylaws.html</a> SVN is used for repository. Issue tracking system is used.

Just like FSF, the structure of ASF corresponds to (A) in Fig. 2. In the management of ASF, project management committee (PMC) gives a grant for the start of projects. ASF leaves the management of their projects to the manager of the project. The project may decide the management policies, rules, and processes by themselves. This mechanism gives maximum flexibility for the projects but not all of the projects can make the best use of them.

From the observations described above, we regard the level of community as level 2 or level 3, depending on the project in ASF.

### 5.4. A case of the OpenStack Foundation

OpenStack Foundation (OSF) was started in 2009 to develop a set of software for infrastructures of cloud computing. Compare to the other communities described above, OSF is new community.

The following table shows an overview of the community based on <sup>[14]</sup>.

Table 5. Overview of OpenStack Foundation

Organization	The OpenStack Foundation
Start date	2009
Number of Projects	Major 5 projects. Many projects are proposed.
Numbers of developers	5600
Software	Depend on projects
Management	The board members of the foundation make important decision about projects. Technical committee maintains source code. Periodic release of software. (period of six month) Source code is maintained by projects. Staging release policy (Beta, Release Candidate, Final, etc.) Github is used for repository. Launchpad is used as an issue tracking system.

One of the most noticeable characteristics is that OSF projects focus on commercial use of the products. Many IT companies contribute to the projects both in financial and human resources. Board members from fund companies make decisions on the organization and the projects. The release cycle of software is 6 month. There are strict rules and processes for the release of software.

As the management rules are documented, that fulfills the requirement of level 3 in our maturity model. The status of projects are quantitatively measured, thus we regard OSF as level 4.

## 6. Issues and evaluation

There are several types of organizations which support or host OSS community. It becomes an issue whether assessment of a maturity level of organization is necessary or not. We analyze how a difference was contributed to maturity level using an example of Linux Foundation and OpenStack Foundation.

### 6.1. Linux Foundation:

Linux Foundation is known as sponsor of Linux creator Linus Torvalds. It is defined as “The Linux Foundation promotes, protects and standardizes Linux by providing unified resources and services needed for open source to successfully compete with closed platforms.”<sup>[15]</sup> It means that Linux Foundation is a “promotion organization” of Linux and support developers, businesses and users to provide funds, conference, law program and standard body. There is no contract with Linux Foundation to join Linux project, and participants to community is free from foundation by laws. Every process is archived by project maintainers who are almost outside of the foundation except for creator Linus.

### 6.2. OpenStack Foundation:

OpenStack Foundation is recognized as base of OpenStack community. They defined themselves as “The OpenStack Foundation promotes the development, distribution and adoption of the OpenStack cloud operating system.”<sup>[14]</sup> It means that OpenStack Foundation is a “community host” of OpenStack communities. All developers are required to contract with the foundation to contribute code to OpenStack project and strongly ruled how to act in the community.

### 6.3. Difference analysis

Communities with each organization are measured as level 5 for Linux Foundation, and level 4 for OpenStack Foundation. The difference of organization role does not make a major contribution to a metric of maturity in our measurement. It is quite important for traditional measurement that a maturity level of organization capability. But we found that an organization that has strong role is not always measured higher level in OSS community maturity.

Here is an issue how to define a scope of OSS community. We found we could ignore the difference of community organization role and characteristics in our approach. We can use same metrics throughout different types of OSS community and organizations. It indicates the approach to evaluate OSS community is a stable way.

## 7. Conclusions and future works

We proposed a maturity model of OSS community to evaluate the quality of products. We tried to apply our model to actual OSS communities to evaluate the model. We found that communities with level 4 or greater had their own management systems to quantitatively evaluate software. Based on their systems, it will be easy for us to evaluate their software before we make a decision on adaptation of the software. For communities less than level 3, we need to rely on other methods such as histories of their activities on mailing lists, wiki, and source code itself.

Based on this study, we are planning to build evaluation method for each level of maturity.

## References

1. The Open Source Initiative, 2014, The Open Source Definition Version 1.9, webpage at <http://opensource.org/osd>
2. Free Software Foundation, 2014, GNU GENERAL PUBLIC LICENSE, webpage at <http://www.gnu.org/licenses/gpl.html>
3. Tai K., Kaiya H., and Kajiri K., 2011, Analysis and evaluation of open source project evolution based on the automatic classification of SCM data, IEICE Technical Report. Software Science 110(458), 79-84, 2011-02-28
4. Ohkura K., Onishi Y., Kawaguchi S., Ohira M., Iida H., and Matsumoto K., 2007, A Method for Activity Prediction using Cluster Analysis on Email Threads, IEICE Technical Report. Software Science 107(275), 41-46, 2007-10-15
5. Tamura Y., Yamada S., 2006, A Method of Reliability Assessment for Open Source System Development Based on Genetic Algorithms, IEICE Technical Report Software Science 106(120), 31-36, 2006-06-15
6. Northeast Asia OSS Promotion Forum WG3, 2011, RepOSS - a practical OSS reference in a flexible repository for assessing OSS, webpage at <http://www.repos.org/webapp/repos/app/osslist>
7. Black Duck Software, 2014, Ohloh - public directory of Free and Open Source Software and the contributors who create and maintain it, webpage at <https://www.ohloh.net/>
8. Paulk, Mark C., et al. .1993, Capability maturity model, version 1.1. Software, IEEE 10.4, pp18-27
9. The Linux Foundation, 2014, About The Linux Foundation, webpage at <https://wiki.linuxfoundation.org/en/About>
10. The Linux Foundation, 2014, Technical Advisory Board (TAB), webpage at [https://wiki.linuxfoundation.org/en/Technical\\_Advisory\\_Board\\_%28TAB%29](https://wiki.linuxfoundation.org/en/Technical_Advisory_Board_%28TAB%29)
11. Free Software Foundation, 2014, Information for Maintainers of GNU Software, webpage at <http://www.gnu.org/prep/maintain/maintain.html>
12. Apache Software Foundation, 2014, Apache Aries Project, webpage at <http://aries.apache.org/>
13. Apache Software Foundation, 2014, Index: Project Listing, webpage at <http://projects.apache.org/indexes/quick.html>

14. The OpenStack Foundation, 2014, Foundation » OpenStack Open Source Cloud Computing Software:, webpage at <http://www.openstack.org/foundation/>
15. The OpenStack Foundation, 2014, OpenStack Compute (Nova) in launchpad, webpage at <https://launchpad.net/nova>
16. Corbet J., Kroah-Hartman G., McPherson A., 2013, Linux Kernel Development :How Fast It is Going, Who is Doing It, What They are Doing, and Who is Sponsoring It, webpage at <http://www.linuxfoundation.org/publications/linux-foundation/who-writes-linux-2013>