

A Two-dimensional Classification Model of OSS: Towards Successful Management of the Evolution of OSS

Toshihiko Yamakami
CTO Office, ACCESS
Nakase 1-10-2, Mihama-ku, Chiba-shi
Chiba, Japan 261-0023
Email: Toshihiko.Yamakami@access-company.com

Abstract—Open source software (OSS) has become mainstream in software development, utilizing a global information infrastructure. The term OSS was coined in 1998 during discussion of the public release of Netscape browser software source code. Since then, OSS has continued to increase in visibility in the software industry. Corporate members are showing increasing commitment to OSS projects.

OSS is a complicated social process to understand. OSS is a multi-faceted phenomenon including code, a licensing structure, a community, development best practices, a method of diffusion, and a philosophy.

The author discusses multiple evolutionary paths for OSS and presents a two-dimensional classification model for OSS. The proposed model facilitates evolutionary design for OSS projects.

Keywords — open source software; social evolution; evolution: social process; community

I. INTRODUCTION

OSS has become mainstream in software development in the last decade. The term OSS was coined in 1998, in the process of discussing the public source code release of Netscape browser software.

The autonomous nature of OSS has presented a wide variety of OSS realizations and evolutions. Linux, GNOME, Apache, Android, Eclipse, MeeGo, Symbian, and MySQL are all popular OSS projects. They have taken all kinds of different directions for their various evolutions.

OSS is not only a software development practice. It is becoming apparent that a wide range of software falls into a category that fits OSS-based development.

The quality of OSS has improved over the years. Some of the OSS-based components have gained the reputation of commercial-grade software. The community's attitude toward OSS has become flexible over the span of time, which has allowed in the number of OSS-based businesses to increase.

With the emerging trend of professional OSS, which utilizes techniques like dual licensing, it is time to revisit the original conflict model between public software and proprietary software. Past lessons indicate that successful corporate engagement and alliance with commercial partners with an awareness of business models are key success factors for OSS projects.

One of the visible trends in the industry is how OSS has extended its coverage of large-scale software. The Google Android Linux kernel consists of 7.7 million lines of code. Symbian has 49 million lines of code. MeeGo, depending upon how the size is counted, is reported to have 100 million lines of code. Although this trend can be accelerated, it is difficult to reverse.

Globally distributed large-scale software development is a real challenge. In order to cope with this challenge, it is important to understand the variety of evolution of OSS. The author presents different evolutionary patterns found in OSS projects. Then, the author proposes a two-dimensional classification model in order to capture the dynamics of OSS evolution.

II. PURPOSE AND RELATED WORKS

A. Purpose of Research

The aim of this paper is to identify the multi-faceted characteristics of OSS evolution from the viewpoint of corporate involvement.

B. Related Works

Open innovation around OSS was discussed by Alexy [1]. OSS represents a typical example of open and distributed innovation [2]. Chesbrough discussed the importance of open innovation in the digital ecosystem [3].

Web 2.0 has brought further digital content in a social dimension since O'Reilly coined the term in 2005 [4]. Karhu discussed Web 2.0 and a digital ecosystem for boosting user-driven service business [5]. Freemium is a business model topic in this era of the Internet [6], often bound to Web 2.0. This is closely related to OSS.

Watson presented the second generation of OSS, or professional OSS [7] in contrast to three types of first generation OSS, such as community OSS, sponsored OSS, and corporate distribution.

Letellier discussed the third generation of OSS [8]. From his description, the first generation came in the early days, in which it was mainly a story of loosely connected individuals (roughly before 1990), the second generation appeared with

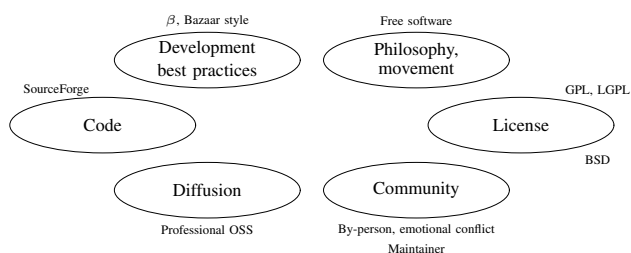


Fig. 1. Six views of OSS.

the worldwide web and was a time of large-scale communities federated by associations of individuals (e.g. the Apache Software Foundation), and the third generation goes the gathering of legal entities. The third generation is beyond the level of single projects in terms of cross-project collaboration. In the third generation, the parties involved are legal entities, not individuals, and requires a new model of governance.

Yamakami presented the importance of organizational commitment in his three-dimensional model of large-scale OSS development [9].

The originality of this paper is to present a “grand design” view of the evolutionary patterns of OSS from a viewpoint of organizational evolution.

III. OSS AND ITS EVOLUTIONARY PATTERNS

A. OSS views

OSS can be viewed from multiple perspectives, as illustrated in Fig. 1. It is a code base, a licensing structure, a philosophy, best practices, a diffusion method, and a community. These factors are interrelated and add complexity to the evolution of OSS.

It is important to identify the role of each aspect during the understanding of evolutionary patterns.

B. Evolutionary Patterns

The Linux kernel is based on a traditional mailing list based traditional project. When Linux was born, BSD had many IPR concerns, therefore, many free UNIX programmers joined Linux. Linus Torvalds started by creating a new operating system kernel in 1991 and announced this system in the Usenet newsgroup “comp.os.minix”. The Linux kernel has been marked by constant growth ever since. It turned into the GNU Public license in 1992 from its original license. The Debian project began calling its product Debian GNU/Linux in June 1994. In the past decade, many paid full-time paid company employees joined this effort, driving Linux to high performance and reliability. Linux can be installed on a wide variety of computer hardware, ranging from embedded devices such as mobile phones, smartphones and wristwatches, to mainframes and supercomputers.

The Free Software Foundation is a non-profit corporation, founded by Richard Stallman in October 1985, to support the free software movement, a copyleft-based movement that aims to promote the universal freedom to create, distribute and modify computer software. From its founding up to the mid-1990s,

FSF’s funds were mostly used to employ software developers to write free software for the GNU Project. Since the mid-1990s, FSF employees and volunteers have mostly worked on legal and structural issues for the free software movement and the free software community. The GNU project produced the GNU licenses, GPL (GNU General Public License) and LGPL (GNU Lesser General Public License), two of the most popular OSS licenses. The Foundation actively engages in the enforcement of the GPL.

Apache had its origin in the NCSA web server. It adopted CGI and, therefore, it became popular in the web community. Volunteers coined the Apache Group to re-energize the project in 1995, and to keep track of patches to the NCSA web server. The Apache Foundation was formed in 1999. The Apache Software Foundation is a non-profit corporation in the United States that supports Apache software projects, including the Apache HTTP Server. The Apache Software Foundation consists of a decentralized community of developers. The software they produce is distributed under the terms of the Apache License. Apache projects are characterized by a collaborative, consensus-based development process, and an open and pragmatic software license. Each project is managed by a self-selected team of technical experts who are active contributors to the project. The Foundation’s objectives are to provide legal protection to volunteers working on Apache projects, and to prevent the Apache brand name from being used by other organizations without permission. It also holds several ApacheCon conferences every year to highlight Apache projects, related technology, and to encourage Apache developers to gather together to share ideas.

The Eclipse project was started by IBM Canada in 1998. IBM made it open source in 2001. The Eclipse Consortium, an unofficial consortium of fifty software industry vendors[2], founded The Eclipse Foundation, a not-for-profit legal entity to lead and develop Eclipse in 2004. The Eclipse Foundation is a not-for-profit corporation that hosts open-source Eclipse projects and helps cultivate both, an open source community, and an ecosystem of complementary products and services. The Eclipse Foundation has a criteria of strategic members. Each strategic member has a representative on the Eclipse Foundation Board of Directors giving it direct influence over the strategic direction of Eclipse.

MySQL AB was established in 1995, and provides MySQL under dual licenses. At the initial stage, the concept of dual licensing was not widespread, therefore, MySQL used a proprietary licensing system that was criticized by some members of the community. Dual licensing allows for the combination of all the good aspects of a proprietary license and an open source license. An open source license provides opportunities for user feedback such as bug reports and suggestions of feature enhancements. MySQL AB was acquired by Sun Microsystems in 2008. It presents a successful business model story.

Snort is open source software that was started in 1998. Sourcefire was established by the original author of Snort to provide commercial enterprise versions in 2001. Sourcefire

completed an initial public offering in March 2007.

GNOME started in 1997. The GNOME Foundation was formed in 2000. GNOME is a GUI toolkit project, inspired by KDE. It provides the GNOME, desktop environment, which works on the X Window System. Originally, it provided a Windows-like GUI, however it gradually changed to a Mac OS-like interface starting with version 2.8. GNOME has a regular release every six months. The GNOME Foundation works to accomplish the goal of the GNOME project: to create a computing platform for use by the general public that is completely free software. The Foundation coordinates releases of GNOME and determines which projects become part of GNOME. It also acts as an official voice for the GNOME project in public communications. It provides educational materials and international technical conferences, such as GUADEC and the Boston Summit. It represents GNOME in related technical conferences sponsored by others, and helps create technical standards and promote the use of GNOME.

The Symbian Foundation is a non-profit organization that stewards the Symbian platform, an operating system for mobile phones, based on the Symbian OS, which was previously owned and licensed by Symbian Ltd.. The Symbian Foundation evangelizes, co-ordinates, ensures compatibility, and provides key services to its members and the community, such as collecting, building, and distributing Symbian source code. It was established in June 2008. The Foundation released Symbian^3 in February 2010 under the Eclipse Public License.

The Google Chrome OS OSS project was announced in July 2009. Google released the Chrome OS source code as the Chromium OS project in November 2009. It is an up-and-coming open source operating system designed by Google to work exclusively with web applications. It is set to have a publicly available stable release during the second half of 2010. From the initial phase, the Chromium OS project started as a project open to the public, and aimed at aggregating industry attention from the very beginning as a strategic industrial movement. The operating system is based on Linux and will run only on specifically designed hardware. The user interface resembles that of the Chrome web browser.

OW2 is a non-profit international consortium for OSS middleware, EAI, e-commerce, clustering, and grid computing. It was formed in January 2007 by merging the ObjectWeb Consortium with Orientware. The goal of OW2 is high quality development of OSS components for distributed computing. It is devoted to developing distributed infrastructure software able to interconnect systems that are heterogeneous by nature. For this purpose, the software developed by OW2 complies with open standards established by independent bodies such as JCP, OMG and OSGi. The aim of the consortium is to promote a dialogue between technology suppliers and user companies in order to foster the emergence of an offering of professional solutions based on open-source infrastructure software

A summary of the involvement of corporate members in OSS projects is depicted in Table I.

TABLE I
EXAMPLES OF CORPORATE INVOLVEMENT IN OSS PROJECTS.

Item	Description
OSS contribution	Contribute code as OSS licenses
Multi-license contribution	Contribute code as OSS licenses under multiple-licenses (as copyright holder)
Corporate assignment	Assign full-time employees to OSS projects
Recruiting	Hire active volunteers as employee
Corporate members	Join and pay special corporate fee and influence strategy (e.g. Advisory Board members)
Marketing	Include OSS and publicize the existence of OSS code
Services	Provide professional services on OSS to customers

TABLE II
CASES FOR FORMING FOUNDATIONS IN OSS.

Factor	Summary	Example
Platform software	Foundation members would like to secure neutrality toward the original authoring company. Or the original authoring company claims neutrality for stake holders.	Symbian, Eclipse, Android
Large-scale project	Projects grow and need large-scale project management	GNOME, Apache
Philosophy, movement	Social movement and socialization of philosophy are pursued.	GNU
Ecosystem construction	Creating balanced governance among authors and users	Eclipse
Inter-project alliance	Creating governance among multiple foundations	OW2

C. Driving Factors for OSS Evolution

Usually, OSS companies are formed when there is a large installed base, with professional services expected to create a sustained revenue stream.

Foundations are formed in the following four cases depicted in Table II.

Each pattern constructs the different driving power for evolution.

The factors that drive organizational development to form a foundation in an OSS project are depicted in Table III.

TABLE III
FACTORS DRIVING ORGANIZATIONAL DEVELOPMENT TO FORM A FOUNDATION.

Factor	Summary
Code management complexity	Assigns permanent staff for code management.
IT infrastructure costs	Accommodates good IT infrastructure and assigns technical staff for good technical support.
Complicated ecosystem	Coordinate players in ecosystem of OSS in an organized manner.
External communications	Coordinate good external communication regarding licenses, roadmap, and other technical and organizational staff. Recruit members in a systematic manner.

TABLE IV
CHARACTERISTICS OF THE TWO DIMENSIONS.

Dimension	Characteristics
Institution	Shows the degree of maturity/visibility of organizational structure governing development projects.
Industrial	Shows the degree of industrial solution goals. If this is high, it shows the industrial attention to the completeness of the package. If this is low, it shows the focus is the on development of each component.

TABLE V
CHARACTERISTICS OF THE FOUR DOMAINS CLASSIFIED BY TWO DIMENSIONS.

Institutional Dimension	Industrial Dimension	Characteristics
Low	Low	Autonomous community-based project focusing on individual or general-use component.
Low	High	Autonomous community-based project focusing on industry-specific component directly related to commercialization.
High	Low	Institutional project with high organizational governance focusing on individual or general-use component.
High	High	Institutional project with high organizational governance focusing on industrial interest directly related to commercialization.

D. Two-Dimensional Classification of OSS

An analysis of organizational and industrial dimensions of OSS is illustrated in Fig. 2. This is an attempt to clarify the positions of the various stages in OSS throughout its evolution. The entities were identified in past examples of OSS evolution. Two dimensions are used to identify the relative positions. The vertical dimension illustrates the transition from community project to institution. The horizontal dimension illustrates the transition from non-industrial project to industrial project. The characteristics of the two dimensions are summarized in Table IV.

The characteristics of the four domains identified by the two dimensions are illustrated in Table V.

These two transitions represent important progress that has become visible in the past decade of the growth of OSS worldwide.

The characteristics of each component in Fig. 2 are depicted in Table VI.

The classic OSS evolutionary path in the late 1990's is depicted in Fig. 3.

This classic pattern is no longer applicable to the modern patterns of OSS development.

Grass-roots efforts are based on the factors depicted in Table VII.

It is important to understand and maintain these different factors driving OSS evolution.

During development of communities, it is important to focus these self and social factors. Work-sharing fosters mutual trust and social commitment. It is a supporting factor for community development. Design and management of these driving factors

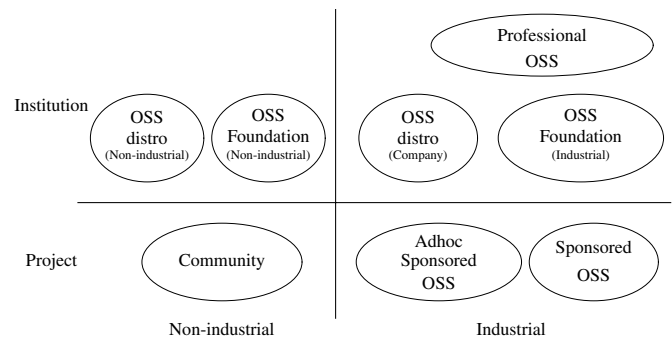


Fig. 2. Two-dimensional model of classification of OSS

TABLE VI
CHARACTERISTICS OF EACH ENTITY POSITIONED IN THE TWO DIMENSIONS

Entity	Description
Community	Autonomous distributed group gathered individually. Primitive Form of OSS. Many OSS projects start in this form.
Adhoc Sponsored OSS	Community with support of full-time corporate employees. No explicit governance from the company who employs them.
Sponsored OSS	Community with explicit funding from a specific company. That corporate support is publicly announced and recognized.
Professional OSS	Community with support from a commercial company. That company is generating major revenues from the software.
OSS distro (non-industrial)	Non-profit company or organization to generate a package of large number of OSS components for public distribution. It is general purpose, not an industry-specific.
OSS distro (company)	Public company to generate package of a large number of OSS components for public distribution. Distribution is the major activity related to the revenue generation of that company.
Foundation (non-industrial)	Non-profit organization to manage one or more OSS projects.
Foundation (industrial)	Non-profit organization to manage one or more OSS projects focusing on a particular industrial solution. Members from the industry take a specific advisory/governing role in the organization.

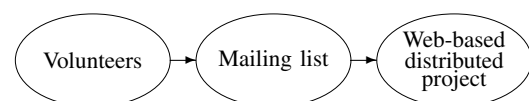


Fig. 3. Classic OSS evolutionary path in the late 1990's

TABLE VII
FACTORS DRIVING GRASS-ROOTS EFFORTS

Factor	Description
Self expression	Expression of one's own skills and passions
Social commitments	Perform social service as a good citizen of community.
Social respect	Exercise one's skill and gain social respect from the community.
Work sharing	Exercise assigned actions and gain shared benefits from an OSS project. Rewarding due to benefits obtained through use of OSS.

TABLE VIII
FACTORS FOR CORPORATE INVOLVEMENT

Item	Description	Example
Existing component	Useful components for corporate software	matchbox, PostgreSQL
Existing platform	Useful platform for cooperate products	Linux, MeeGo, Eclipse
Tool	Useful tool for corporate products and services	Eclipse
Strategic platform	Strategic platform for future business development and ecosystem development	Android, Symbian, Chromium OS
Supporting OSS projects	Encouraging OSS projects.	Mozilla

are a key to success of evolution of OSS.

In retrospect, many OSS projects have been empowered by the involvement of corporate members. The factors for corporate involvement are depicted in Table VIII. It has been derived from utilization of external resources by enterprises.

OSS is a combination of grass-roots efforts and corporate involvement. Emotional attitudes of OSS communities are very similar to those of Internet communities against commercialization of the Internet in the middle of 1990's. Corporate involvement sometimes creates bias in community-based development and management. It should be noted that healthy cooperation with commercialization players promotes wide adoption and increasing engagement in OSS projects. Awareness of multiple evolutionary paths, including corporate involvement, certainly adds strength and competence to the management of OSS projects.

Pure dualisms, such as OSS vs. proprietary software, do not address the current industrial landscape in a proper manner. The proposed model describes a spectrum-based view of the management of OSS projects.

IV. DISCUSSION

A. Advantages of Proposed Classification Model

The proposed dimensional analysis presents a perspective for capturing the reality of OSS projects. OSS is a complicated process and is influenced by multiple technological and economical landscapes. It is a complicated ecosystem involving a wide range of industrial stakeholders.

The proposed dimensional analysis provides a useful two-dimensional framework for capturing the current status of each OSS ecosystem entity and its possible evolutionary paths.

Rome was not built in one day. Large-scale software takes a long time before it matures. Awareness of business models, partner alliances, and organizational governance are of vital interest to OSS projects if they are aimed at sustained growth in reality. The proposed two-dimensional classification model provides insight into how a community can grow to accomplish the goal of persistent existence allowing software to mature.

B. Implications for successful management of evolution of OSS projects

OSS projects face many challenges in autonomous distributed development. For the persistent growth of a project, it is important to have awareness of the ecosystem. One of the vital factors for an ecosystem is corporate engagement.

The proposed two dimensional classification model can be used in designing the evolution of an OSS project in order to best utilize corporate engagement. The model provides understanding of the current status and possible evolutionary paths.

In the early stages, the stereotypical view of OSS was of a public service with shared burdens and non-profitable give-and-take. Lessons from the past decade demonstrate that many successful OSS projects make use of the vital interests of commercial partners and the institutional management of development with the long-term commitment of commercial partners. OSS is a kind of open innovation, however, the word *open* does not exclude any industrial or organizational commitments. Organizational roadmap design and project management, made possible by industrial commitments, enable the productive evolution of OSS projects.

C. Outcome of Evolution

There are many outcomes of the evolution of OSS. The following patterns are identified:

- Harmonization of Organizations
- Vertical Conflicts
- Horizontal Conflicts

In the most ideal case, organizational evolution brings harmony to users, communities, and the particular organization.

In reality, there are two types of conflicts and tensions in OSS communities. In the case of vertical conflict, communities and the evolving organization may have different interests. In the case of horizontal conflict, multiple different organizations may have different interests in a particular OSS project or a package of multiple OSS projects.

D. Limitations

This research is descriptive and there is little quantitative verification of the arguments in this paper. It provides a framework for positioning an OSS project and helps explore the possible further organizational growth of that project.

The two dimensions are qualitative, high and low, and detailed analysis of each dimension is beyond the scope of this paper.

The quantitative measures for the stages of evolution and in-depth investigation of a particular OSS organization are beyond the scope of this paper.

V. CONCLUSION

Since the term OSS was coined in 1998, the autonomous nature of OSS has engendered a wide variety of different evolutionary paths. Examples include Linux, GNOME, Apache, Android, Eclipse, MeeGo, Symbian, and MySQL. Although

this diversity may be increased in the future, it is difficult to expect that it will decrease.

The industrial trend of large-scale OSS is becoming increasingly visible. It adds a further challenge to OSS, in the form of fundamentally global distributed software development. It is becoming increasingly important to be aware of OSS evolution during its development.

The author first discusses the factors that drive the evolution of OSS. Then, the author proposes a two-dimensional model of the evolution of OSS, with a focus on a time-dimensional perspective.

There is no single path for the evolution of OSS, as a retrospective view of past OSS development clearly shows. It is important to understand the costs and benefits of the organizational management of OSS projects. And, it is important to increase awareness of evolutionary patterns in OSS in order to facilitate the healthy growth of OSS projects.

The emerging success of OSS projects sometimes obscures the challenges for evolution of OSS. As the size of OSS projects grows, it is more difficult to manage the healthy evolution of OSS projects. From this aspect, we are still at an early stage of capturing in-depth understanding of evolution of OSS. Evolution of OSS as a research topic is still an unexplored and challenging research field, which needs organizational and social science approaches and methods to address dynamic nature of social evolution.

REFERENCES

- [1] O. Alexy and J. Henkel, "Promoting the penguin: Intraorganizational implications of open innovation," Social Science Research Network, available at http://papers.ssrn.com/sol3/papers.cfm?abstract_id=988363, February 2009.
- [2] O. Alexy and G. George, "Placing a value on openness: A legitimacy perspective on open and distributed innovation," Social Science Research Network, available at http://papers.ssrn.com/sol3/papers.cfm?abstract_id=1019527, October 2009.
- [3] H. W. Chesbrough and A. R. Garman, "How open innovation can help you cope in lean times," The Magazine, Harvard Business Review, available at <http://hbr.org/2009/12/how-open-innovation-can-help-you-cope-in-lean-times/ar/1>, December 2009.
- [4] T. O'Reilly, "What is Web 2.0: Design Patterns and Business Models for the Next Generation of Software," available at <http://oreilly.com/web2/archive/what-is-web-20.html>, September 2005.
- [5] K. Karhu, A. Botero, S. Vihavainen, T. Tang, and M. Hämäläinen, "A digital ecosystem for boosting user-driven service business," in *MEDES '09: Proceedings of the International Conference on Management of Emergent Digital EcoSystems*. New York, NY, USA: ACM, 2009, pp. 246–253.
- [6] C. Anderson, *Free: The Future of a Radical Price*. Hyperion Books, July 2009.
- [7] R. T. Watson, M.-C. Boudreau, P. T. York, E. Greiner, and J. Donald Wynn, "The business of open source," *CACM*, vol. 51, no. 4, pp. 41–46, April 2008.
- [8] F. Letellier, "Open source software: the role of non-profits in federating business and innovation ecosystems," <http://flet.netcipia.net/xwiki/bin/download/Main/publications-fr/GEM2008-FLetellier-SubmittedPaper.pdf> (a submission for AFME 2008), January 2008.
- [9] T. Yamakami, "A three-dimensional view model of open source-aware software development for large-scale mobile software platforms," in *DEST 2010*. IEEE Computer Society, April 2010, pp. 131–135.