

Generations of OSS in Evolutionary Paths: Toward an Understanding of where OSS is Heading

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) is becoming increasingly visible throughout the software industry. The entire social and commercial fabric supporting the creation of software is changing. OSS is a global distributed community-oriented model of software development with an OSS-oriented license. This characteristic of OSS has created a wide range of diversity and heterogeneity in OSS. With less shared common ground, the number of OSS projects continues to grow and involve an increasing number of people. The author compares different arguments regarding different generations of OSS and presents the implications of the multiple views on the evolution of OSS.

Keywords — open source software; evolution analysis; generations; evolutionary patterns; management of OSS

I. INTRODUCTION

Open source software (OSS) has deeply penetrated the software industry. It has extended its use in academic, industrial, and particularly in domains such as embedded software. Advances in IT infrastructure have facilitated a wide acceptance and deployment of OSS.

The autonomous nature of OSS has increased its diversity in use, development, and commercial contexts. The community-based nature of OSS shows the wide variety of communities involved in OSS. Updegrave joked in his plenary address that there are three different opinions when there are two people talking about OSS [1]. It has become increasingly difficult to identify the basic building blocks of OSS and to develop mutually-shared views of OSS. With increasingly less common ground, OSS projects are growing worldwide everyday.

As the stacks implemented in OSS have come to a complete stack, the use of OSS has further penetrated into many application and industrial domains because OSS provides an almost turn-key solution. OSS has come to provide a wide variety of applications and infrastructures to a wide range of industrial domains. This penetration of OSS was completed over more than a decade. The span of time taken for the proliferation of OSS has exposed many courses of evolution, which represent different aspects of OSS.

The author examines multiple evolutionary patterns of OSS in order to understand the diversity of OSS with a viewpoint toward its evolution over a span of time.

II. PURPOSE AND RELATED WORKS

A. Purpose of Research

The aim of this paper is to identify the possible future directions of OSS by analyzing multiple different stage views of evolution of OSS.

B. Related Works

OSS was separated from free software in the late 1990's in order to revisit the commercial issues of using OSS. It is a paradoxical to publish source code, the core competence of the software industry, so openly. Fitzgerald discussed the contradictions, paradoxes and tensions of OSS in [2].

Raymond discussed open source from the business model perspective in this famous open source work series [3].

OSS has continued to evolve. Watson presented the second generation of OSS, or professional OSS [4] 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 [5] from a perspective of its organizational structure.

The long-term factors of OSS have also attracted the attention of researchers. Subramaniam discussed success factors using longitudinal data on OSS projects [6] and presented the impacts of different license types. Yu discussed time series analysis techniques to study the time dependence of open-source software activities using mailing lists, bug reports, and revision history [7] and presented diversity in cyclic-ness and in seasonal dependency.

As the size of OSS software has grown, organizational governance has emerged. Examples include the Eclipse Foundation [8] and Apache Software Foundation [9]. There are new industrial organizations emerging for industry-specific software: for example, mobile handset software-related foundations including the LiMo Foundation [10], the OHA [11], and the Symbian Foundation [12].

The originality of this paper lies in its identification of the driving factors in different evolutionary paths of OSS.

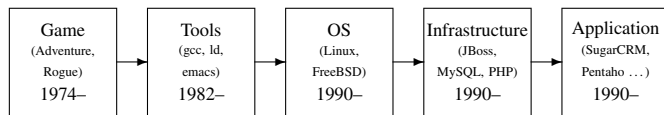


Fig. 1. Augustin's 5 generation-view

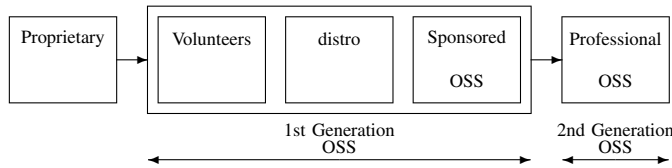


Fig. 2. Watson's 2 generation-view

III. VIEWS ON GENERATIONS OF OSS IN LITERATURE

A. Augustin's 5 Generation-view

Augustin discussed the generations of OSS dating back to 1974, from a viewpoint of stacks of OSS. Augustin presented the 5-generation view as depicted in Fig. 1. The first generation of OSS consists of games distributed by mailing lists. The second generation consists of tools for development environments. The third generation is the OS (operating system). The fourth generation consists of infrastructures such as database and web server scripting language. The fifth generation is applications. It is a generation analysis by domains or fulfilling computing stacks.

B. Watson's Two Generation-view

Watson presented a view in terms of a business model. Watson discussed the emergence of professional OSS, as the second generation of OSS. The two generations of OSS are depicted in Fig. 2. The emerging second generation is professional OSS, in which a company contributes its assets to open source and explores a wide range of business models based around that. Full-time employees are engaged in OSS to leverage their business models.

The author has a sense that this increased interaction between OSS and business was similar to that between the Internet and business in the first half of the 1990s. In the case of the Internet, it was gradually recognized that a fusion with business is the way that leads the Internet to its full potential. The author considers that a similar conclusion will be drawn in the case of OSS.

C. Letellier's Three Generation-view

From Letellier's description, the first generation of OSS was in the early days, when it was mainly a story of loosely connected individuals (roughly before 1990), the second generation appeared with the worldwide web and was the time of large-scale communities federated by associations of individuals (e.g. the Apache Software Foundation), and the third generation is the gathering of legal entities. The third generation goes beyond the level of single projects with cross-project collaboration. In the third generation, the parties involved are legal entities, not individuals, where a new governance model

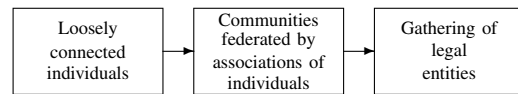


Fig. 3. Letellier's 3-generation view

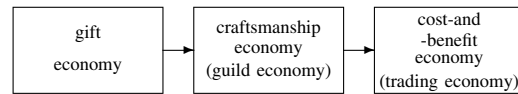


Fig. 4. 3-generation view of base economy types of OSS.

is needed. Letellier described three generations of OSS, as depicted in Fig. 3.

IV. EMERGING VIEWS ON GENERATIONS OF OSS

A. Base Economy Types of OSS

The shift in base economy types of OSS is depicted in Fig. 4. In the early days of OSS, the basic economy type is the gift economy. They give, so we gave. Gifts in turns are the basic constructs of the economy. This is similar to the primitive economy in early civilization. There are neither economic rules nor quantitative measures to be used in trades. It is the starting point of an economy.

Then, if a project and the community for it persists, then social norms, values, and social ties are developed. Internal rules and guidelines are developed for a community. In this stage, an economy of Guilds is used. A guild is an association of craftsmen in a particular trade. Confraternities of workers were organized in a manner that was something between a trade union, a cartel and a secret society. Community is the core part of this economy.

Then, OSS collides with the real world economy, the basic economy type is shifted to a cost-benefit economy. This is the common trading economy, used in the modern world. There are two types of economy. One is the OSS-centered economy. When Red Hat started copy-left, many people thought that copy-left could not be a sustainable business. Red Hat still persists and has proven that the radical copy-left is still viable when a related business is successfully built. The other is the professional OSS economy. Dual license is one example. Full-time employees with fully crafted business models enable this type of economy, which is a common business model with business model engineering. This is a regular economy with the leverage of OSS.

B. Generations of OSS Licenses

Licenses are important aspects of OSS. Even so, there is a large number of licenses in OSS, the author identifies a kind of evolution in licenses, as depicted in Fig. 5. Early examples of OSS include the GPL (GNU Public License) and BSD licenses. GPL is an important license which is based on free software, and pursues the freedom of software. It is unique in that it represents a philosophy rather than software development practices.

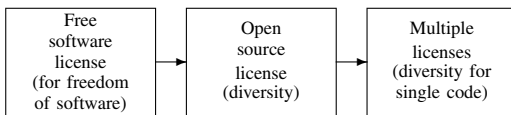


Fig. 5. 3-generation view of licenses.

Then the next generation is open source software licenses. The term OSS was coined when the community discussed licensing with the publication of Netscape software in 1998. There was some misunderstanding of licenses and many projects started to coin new licenses for their source code, which lead to a significant number of OSS licenses. The OSS licenses represent the diversity of OSS projects.

As people continued to learn and explore OSS licenses, it was recognized that there is no reason that one piece of software should have only one license. Software code can have as many licenses as needed. The dual license in OSS is a departure from a rigid and fixed licensing system. It allows a certain flexibility of business development to a company, as long as that company has a copyright for the entire code. This is the basis for professional OSS.

C. Evolution of Large-scale OSS

As features are extended and communities grow with the enhancements in IT infrastructure, the volume of code simply continues to grow. This exposes OSS projects to the challenges of large-scale software development. In order to cope with this, the following generations are observed during the evolution of large-scale OSS, as depicted in Fig. 6.

At the first stage, a community grows and extends a number of small projects into the community. The IT infrastructure enables management of a larger-scale of software. Accumulation of community experience, project management experience, and increased capabilities in development environments enable the development of large-scale software.

At the second stage, the idea of developing an entire software package in the community is abandoned. The architecture and ecosystem to enable further software development in relation to third-party software and corporations are developed. One example is the separation of platform and plug-in components. The shared platform is developed and maintained by the community. Each corporation can develop their own plug-in for its purposes, including business purposes. Examples include Eclipse.

At the third stage, a foundation is established to govern a large number of projects that are loosely connected. Each project is isolated from the viewpoint of functions and project management. The foundation has a higher level of orchestration. Examples include the Apache software foundation and the GNOME foundation.

D. Evolution of Corporate Engagement

As OSS has improved its presence and visibility in the software industry, it has increased opportunities for corporate involvement.

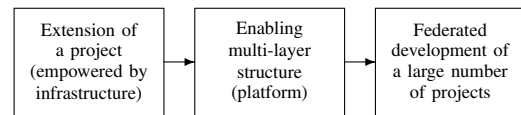


Fig. 6. 3-generation view of large-scale software development.

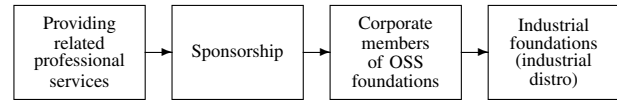


Fig. 7. 4-generation view of corporate engagement.

At the first stage, companies provide related professional services. Examples include distro companies. A distro is a software package which is a bundle of a specific software, or even an entire operating system with included software, already compiled and configured. It is generally the closest thing to a turn key form of OSS. Each OSS component continues to evolve in beta releases. Therefore, it is necessary for some companies to provide professional skills to ensure a turn key OSS solution.

At the next stage, companies provide sponsorship. When Linux became mature, many large enterprises discovered the value of a free version of Unix and invested their full-time employees in the maintenance of OSS. Eclipse was coined by IBM, but IBM opened this asset for the shared maintenance of OSS, because the IDE itself was not considered a core competency of their business.

At the third stage, OSS communities recognized the values of corporate involvement. Some foundations created corporate memberships, such as Advisory boards. This involvement created a revenue stream for OSS foundations for sustained growth. It enables many large-scale corporation secure a strategic position in OSS foundations.

At the fourth stage, corporate involvement leads to a foundation for specific industrial purposes. MeeGo and the Linux Foundation is an example. This is something like a distro for a mobile handset middleware platform, including 100 millions lines of OSS code.

The generations of corporate involvement are depicted in Fig. 7.

E. Evolution of Diffusion using OSS

OSS facilitates the unrestricted distribution of software code. This is used to leverage diffusion. The generations of diffusion using OSS are depicted in Fig. 8.

The first stage accommodates the diffusion of open standards. The implementation of open standards in OSS leverages the acceptance of a standard. This also fits OSS because open standards provide clear requirements, which eliminates overhead in OSS development with less ambiguity compared to other types of software.

The second stage leverages the diffusion of shared platforms. Splitting software into a shared platform and plug-ins provides efficient development of software as long as

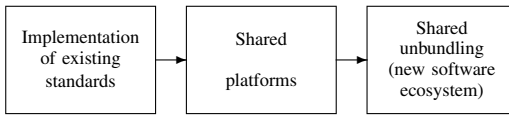


Fig. 8. 3-generation view of diffusion using OSS.

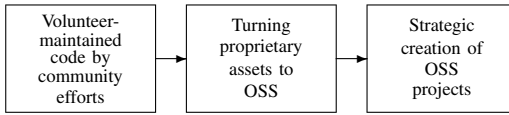


Fig. 9. 3-generation view of creation process of large-scale software.

the architecture is properly designed. Eclipse is one example. When the platform is based on OSS, it provides transparency of governance, neutrality of delivery control, and public participation. It also facilitates open distribution of technical information.

The third stage leverages unbundling and unlocking. The digital economy accelerates the unbundling of digital content. Examples include PHP, database infrastructure.

F. Emergence of New Generations of OSS

The shift in creation of large-scale software is depicted in Fig. 9.

The initial stage is code maintained by volunteers. The second stage is professional OSS, casting proprietary assets to OSS. The third stage is strategic involvement. An example is the Chrome OS by Google. It is unique in that, the project was established as OSS prior to the software release, even though it is a strategic project by a corporation. The publication of a project puts a certain burden on the corporate software development. However, if the deliverables are not for sale, it is more important to influence the whole industry using the open information structure of OSS.

V. DISCUSSION

A. Driving Factors of Evolution of OSS

With a review of stage views in the literature and emerging stage views, we can see the increased engagement of corporations and foundations. That is related to increasing confrontation with business models and society models. It is related to the increased variety of organizational governance and business models. And, it is also related to the diversity of licenses. We can see that the aggregated skills and experience of OSS further extends the diversity and complexity of OSS management.

Confrontation with business and business models has both intrinsic and extrinsic factors. The intrinsic factors include the completeness of OSS components and commercial quality of OSS components. These factors draw industry attentions. The amount of code has grown, therefore, more structured and organizational approaches are needed in OSS projects, in an intrinsic manner.

TABLE I
VISIBILITY OF COMPLEXITY AND STRATEGIC ENGAGEMENT OF OSS IN THE TIME-SERIES ANALYSIS

Type	Complexity	Strategic Engagement
Augustin's 5 generations	N.A.	N.A.
Watson's 2 generations	N.A.	Strong
Letellier's 3 generations	Strong	Weak
Economy types	N.A.	Fair
Licenses	Fair	(Indirectly) Fair
Large-scale software	Strong	N.A.
Corporate engagement	(Indirectly) Fair	Strong
Diffusion	Fair	N.A.
Creation process	N.A.	Strong

The extrinsic factors include the fact that the industrial players have come to realize that competition against OSS is not feasible in many business cases. This awareness drives much of business model development surrounding OSS components, due to shifts in the extrinsic business models.

B. Implications from Analysis of Evolution of OSS

The time-series analysis of the evolution of OSS reveals that the increased complexity and increased strategic engagement of OSS are becoming increasingly visible. The visibility of the complexity and strategic engagement of OSS in time-series analysis is depicted in Table I. Augustin's 5 generation-view is not aligned relative to these dimensions, partially because it is an argument for vertical application stack layers. The author marked conservatively in relation to diffusion. It is related to unbundling and platform engineering, which is indirectly linked to large-scale software and strategic engagement.

Multiple licenses have links to professional OSS and business model engineering. The author put (Indirectly) for licenses for the strategic engagement dimension to be conservative. The same is applicable to corporate engagement. Corporate engineering is related to business model and ecosystem engineering. The author put (Indirectly) to be conservative.

The author argues that OSS is getting into two new arenas as follows:

- large-scale software engineering, and
- strategic business model and ecosystem engineering

The bazaar-style OSS development placed emphasis on the number of eyeballs in the past. This was probably appropriate at a size of thousands of lines of code. The success of OSS over more than a decade has brought it to a scale of millions lines of code. Even with advances in IT infrastructure and experience of OSS-based quality assurance, it is a tough challenge to maintain millions of lines of code, whether it is OSS or proprietary. This has also brought the need for higher level governance, which inevitably invites strategic engagement of key stakeholders.

It implies that software engineering needs new skills in addition to the classic OSS skills and competencies. Newly required skills include business model, ecosystem, and large-scale software engineering.

C. Limitations

This is a descriptive study to identify the directions of movements in OSS. The stage views analyzed in this paper are qualitative. Quantitative verification of the evolution of OSS is beyond the scope of this paper. The comparisons are descriptive and lack any quantitative measures.

This is an exploratory work, therefore, the in-depth analysis of the evolution of OSS remains for further research.

VI. CONCLUSION

We have witnessed remarkable growth in software companies in the last three decades. Closed secrecy and copyrights are the main protections for this type of business. OSS has challenged these concepts and has achieved remarkable success.

With increased success, we have also witnessed increased diversity, heterogeneity, and complexity of OSS. There is no sign that any of these characteristics will decrease in the near future. In many cases, OSS-based development is more complicated than proprietary software development.

The author tries to decompose this complexity using a time-dimension analysis of the evolution of OSS. This attempt highlights the directions of movements in the management of OSS.

Parallel comparisons of multiple time-series analysis of the evolution of OSS reveals that increased complexity and strategic engagement are the two major trends in the current landscape of OSS. It implies that software engineering needs business model and ecosystem engineering, and large-scale software engineering, as well as the classic OSS skills and competencies.

OSS has covered a wide spectrum of software and has increased in its complexity, heterogeneity, and diversity. The author analyzed multiple evolutionary paths of OSS in order to identify the factors that drive the diversity.

The author identified two factors, complexity, in code and project management, and strategic engagement, as the major driving factors of these evolutionary paths.

The author highlighted these two factors in the analyzed evolutionary paths to confirm the importance of these factors over a wide range of different evolutions of OSS.

REFERENCES

- [1] A. Updegrove, "The future of F/OSS: Questions, challenges and opportunities," Plenary Address of OSS'10 (Notre Dame, IN, USA), June 2010.
- [2] B. Fitzgerald and P. J. Agerfalk, "The mysteries of open source software: Black and white and red all over?" in *HICSS '05: Proceedings of the 38th Annual Hawaii International Conference on System Sciences*. Washington, DC, USA: IEEE Computer Society, January 2005, p. 196.1.
- [3] E. S. Raymond, "The magic cauldron," <http://www.catb.org/~esr/writings/cathedral-bazaar/magic-cauldron/>, August 2000.
- [4] 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.
- [5] F. Letellier, "Open source software: the role of nonprofits 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.
- [6] C. Subramaniam, R. Sen, and M. L. Nelson, "Determinants of open source software project success: A longitudinal study," *Decis. Support Syst.*, vol. 46, no. 2, pp. 576–585, January 2009.
- [7] L. Yu, S. Ramaswamy, R. B. Lenin, and V. L. Narasimhan, "Time series analysis of open-source software projects," in *ACM-SE 47: Proceedings of the 47th Annual Southeast Regional Conference*. New York, NY, USA: ACM, March 2009, pp. 1–6.
- [8] The Eclipse Foundation, "The Eclipse Foundation web page," available at: <http://www.eclipse.org/>, 2004.
- [9] The Apache Software Foundation, "The Apache Software Foundation web page," available at: <http://www.apache.org/>, 1999.
- [10] LiMo Foundation, "LiMo Foundation Home page," <http://www.limofoundation.org/>, January 2007.
- [11] Open Handset Alliance, "Open Handset Alliance web page," available at: <http://www.openhandsetalliance.com/>, 2007.
- [12] Symbian Foundation, "Symbian Foundation web page," available at: <http://www.symbianfoundation.org/>, 2008.