

The Third Generation of OSS: A Three-Stage Evolution from Gift to Commerce-Economy

Toshihiko Yamakami

ACCESS, CTO Office,
1-10-2 Nakase, Mihama-ku, Chiba-shi, Chiba-ken, JAPAN 261-0023
<http://www.access-company.com>

Abstract. Linux is penetrating into mobile software as the basis for a mobile middleware platform. It is accelerating the increasing visibility of open source software (OSS) components in mobile middleware platforms. Considering the 10-million lines of code of OSS-based industrial platforms such as a mobile middleware platform, engagement in foundations is inevitable for large-scale packages of OSS for industrial solutions. The author discusses the driving factors toward a foundation-based OSS and the transition of the underlying economy types to analyze the transitions to the third-generation OSS.

Keywords: Open source software (OSS), evolution of OSS, industrial OSS, foundation-based OSS.

1 Introduction

Linux has penetrated into a wide range of digital appliances, e.g. mobile handsets, digital TVs, game consoles, and HD recorders. It facilitates the reuse of PC-based rich user experience data service software with the high speed network capabilities of an embedded software environment. As Linux-based software is widely adopted for digital appliances, the original weak points of Linux in an embedded environment have been addressed, namely its real time processing and battery life capabilities. The author reviews the patterns of evolution in the past literature, and proposes the concept of third-generation OSS, a foundation-based OSS. The author discusses the driving factors toward a foundation-based OSS and the transition of the underlying economy types to analyze the transitions to the third-generation OSS.

2 Purpose and Related Work

2.1 Purpose of Research

The purpose of this research is to identify and understand the transitions in OSS with regard to base economy types.

2.2 Related Work

OSS was separated from the concept of free software in the late 1990's in order to revisit the commercial issues of using OSS. It is paradoxical to publish source code, the core competence of the software industry, so openly. Fitzgerald et al discussed the contradictions, paradoxes and tensions of OSS in [4]. Fitzgerald named OSS 2.0, the open source phenomenon has undergone a significant transformation from its free software origins to a more mainstream, commercially viable form [3]. This paper deals with a follow-up to OSS 2.0.

Raymond discussed open source from the business model perspective in this famous open source work series [8].

OSS has continued to evolve. Watson presented the second generation of OSS, or professional OSS [13] in contrast to the three types of first generation OSS: community OSS, sponsored OSS, and corporate distribution.

Letellier discussed the third generation of OSS [5] from the 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 [9] 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 [15] 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 [12] and Apache Software Foundation [11]. There are new industrial organizations emerging for industry-specific software: for example, mobile handset software-related foundations including the LiMo Foundation [6], the OHA [7], and the Symbian Foundation [10].

Capra et al analyzed analyses the impact of firms' participation on popularity and internal software design quality for 643 SourceForge.net projects [2].

Yamakami presented multiple views of the generations of OSS in order to understand the diverse evolutions of OSS [14].

The originality of this paper lies in its examinations of the transitions of the base economy types in order to identify and understand the evolution of OSS.

3 Landscape of OSS

3.1 Generational Views of OSS

OSS has been successful at penetrating into the world, including the software industry. It is no longer a question any more whether we should use OSS, because the current industrial best practices adopt the OSS-based development in many industrial domains. Examples include mobile multimedia software for mobile handsets.

OSS has become a reasonably complete stack, therefore, many user applications can be built using only OSS. This user application covers many industrial applications and enterprise applications.

Augustin discussed generations of OSS dating back to 1974 from the viewpoint of stacks in OSS. Augustin presented a 5-generation view as depicted in Fig. 1 [1].

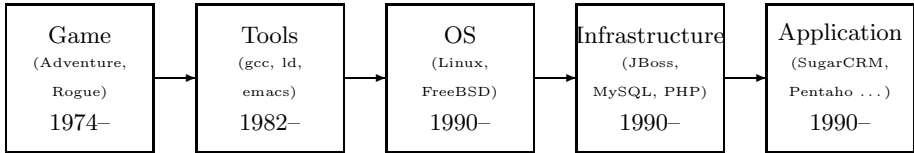


Fig. 1. Augustin’s 5 generation-view

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 infrastructure elements such as databases and web server scripting languages. The fifth generation is applications. It is a generational analysis by domains or completion of computing stacks.

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.

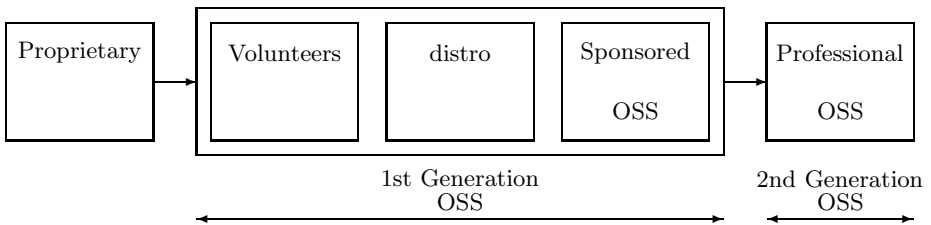


Fig. 2. Watson’s 2 generation-view

The second generation that emerges is professional OSS, in which companies contribute their assets to open source and explore 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 was the way that would lead the Internet to its full potential. The author believes that a similar conclusion will be drawn in the case of OSS.

3.2 Emerging New Aspects of OSS: Foundation Dimension

The completeness of OSS components for industrial solutions allows for a complete platform consisting of OSS modules. Large-scale OSS platforms have emerged these days, such as Symbian, and Android. Such industry-scale OSS solutions require the following, depicted in Table 1.

Table 1. Requirements for industry-scale OSS solutions

Item	Description
Governance	Governance to manage quality and performance of the entire platform.
Neutrality	Neutrality to serve as an industrial platform that can be supported by a wide range of stakeholders.
OSS community-friendliness	Friendliness toward upstream OSS communities to enable coordination and collaboration with upstream communities.
Ecosystem considerations	Ecosystem to enable different industrial stakeholders to participate, enabling both the maintenance of common parts and customized differentiation for each stake holder.

Considering the 10-million lines of code of OSS-based industrial platforms such as a mobile middleware platform, engagement in foundations is inevitable for large-scale packages of OSS for industrial solutions.

3.3 Perspective for Three-Stage Evolution of Large-Scale OSS

The author believes that the shift towards foundation-based OSS is a natural consequence of the shifts in base economy types of OSS.

The shift in base economy types of OSS is depicted in Fig. 3.

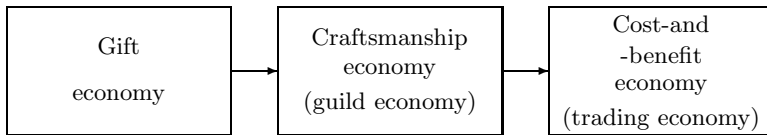


Fig. 3. 3-generation view of the base economy types of OSS

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 of early civilization. There are neither economic rules nor quantitative measures to be used for trades. It is the starting point of an economy.

If a project and the community surrounding it persist, 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. The 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 a business based on giving copy-left software away and providing expert services for a fee, many people thought that copy-left would not be a sustainable business. However, 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. The dual license is one example. It is allowed to provide multiple licenses including OSS license and commercial license. The dual license is to provide codes with an OSS license and a commercial license. Full-time employees with fully crafted business models enable this type of economy, which is a common business model with business model engineering. Another example is proprietary add-ons.

The misfits and fits with the current industrial landscape are illustrated in Table 2.

3.4 Implications from the Transitions of Base Economy Types

These transitions in base economy types can explain the evolution of licenses.

Table 2. Misfits and fits with the current industrial landscape

Economy type	Fits	Misfits
Gift	Universally applicable	Gift economy is important when there is a scarcity of code, however, once a large mass of code has been contributed, a value-weight of a gift decreases.
Guild	Strong social ties and norms help in the management of a large mass of code	Person-dependence may create obstacles to consistent handling of large-scale packages of OSS code for industrial solutions.
Commerce	Large-scale OSS projects require alignment to roadmaps, structured governance, skilled project management, coordination among different stakeholders, and industrial support.	Not all OSS communities accept these commerce-driven activities due for the historical reason and the volunteer-oriented nature of maintenance.

Licensing is an important aspect of OSS. Even so, there are a large number of licenses in OSS, the author identifies a kind of evolution in licenses, depicted in Fig. 4.

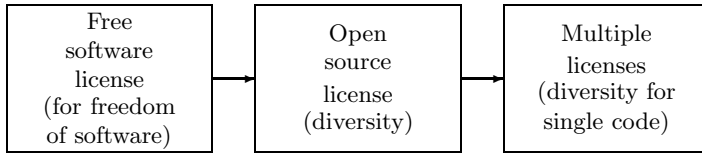


Fig. 4. 3-generation view of licenses

Early examples of OSS licenses include the GPL (GNU Public License) and BSD licenses. GPL is an important license that 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.

Then next generation consists of 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 create new licenses for their source code, which lead to a significant number of OSS licenses. The OSS licenses represented the diversity of OSS projects.

As people continued to learn and explore OSS licensing issues, 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.

As features are extended and communities grow with enhancements in the 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 understand these challenges, the following generations are observed during the evolution of large-scale OSS, depicted in Fig. 5.

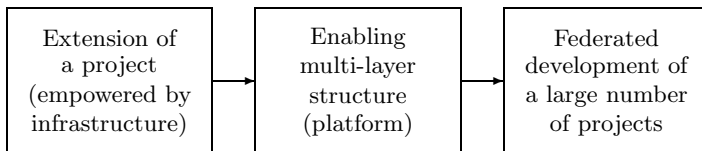


Fig. 5. 3-generation view of large-scale software development

In the first generation, a community grows and extends a number of small projects in the community. The IT infrastructure enables the management of

larger-scale software projects. Accumulation of community experience, project management experience, and the increased capabilities of development environments enable the development of large-scale software.

In the second generation, the idea of developing an entire software package within 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.

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

The generations of diffusion using OSS are depicted in Fig. 6.

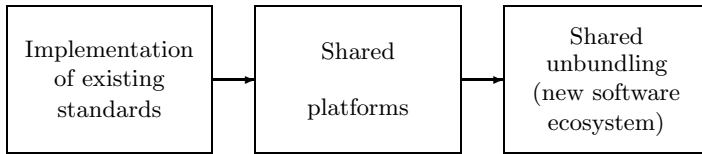


Fig. 6. 3-generation view of diffusion using OSS

The first generation 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 generation leverages the diffusion of shared platforms. Splitting software into a shared platform and plug-ins provides efficient development of software as long as the architecture is properly designed. Eclipse is one example. When a 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 generation leverages unbundling and unlocking. The foundation-based OSS provides a base for industrial ecosystem such as application stores and white-brand SDKs. The white-brand SDK in the embedded software engineering enables unbundling third-party applications that replace the second-party applications in the early days.

The transitions of corporate engagement are depicted in Fig. 7.

In the first stage, enterprises are engaged in users of OSS.

In the second stage, enterprises place resources for OSS projects, where stable development and delivered quality of code influences their businesses. One example is Eclipse. IBM hosted Eclipse in its early stage of and worked in the Eclipse

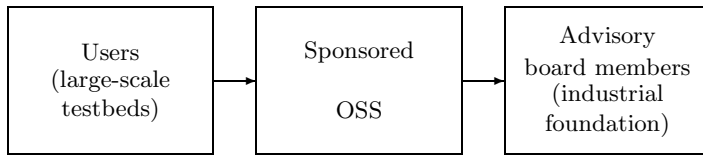


Fig. 7. 3-generation view of corporate engagement in OSS

consortium. Then, a non-profit foundation that owned all the code would work much better. IBM donated the entirety of the Eclipse code to the Eclipse Foundation in order to form an industrial framework to support Eclipse development. Another example is Linux. IBM, HP, SGI, Intel, and other industrial players provided human resources toward the improvement of Linux to commercial-grade quality.

In the third stage, many leading industrial players play an important role in industry-backed OSS-based foundations. They pay extra fees for an Advisory Board to manage the industrial governance of large-scale OSS projects. The extra fees can be used to reduce the fees of other regular members to host a wide range of support for diverse stakeholders.

4 Discussion

4.1 Advantages of the Proposed Model

The proposed transition model addresses the shifts in underlying schemes during the evolution of OSS.

Over the decades, OSS has gained experience, and has achieved a high level of quality, performance, and completeness. It has also increased its code size, diversity and heterogeneity. This has brought about changes in the landscape of OSS.

In the early stage of OSS, there was some anti-proprietary feeling in many OSS communities. After professional OSS emerged as well as a dual licensing scheme, an increasing number of OSS projects accepted their co-existence with enterprise involvement. The scale of current OSS projects requires some management and governance skills from large organizations such as global industrial players.

The proposed model highlights this underlying scheme change over decades of OSS evolution.

4.2 Implications

OSS is a multi-faceted phenomenon, as shown in Fig. 8.

One aspect of OSS is philosophy and social movement. From a dimension such as this, there is still hesitation to accept the commerce-based economy that underlies OSS.

Although there is some anti-proprietary feeling in many OSS communities, with respect for the volunteer attitudes in the communities, large organizations,

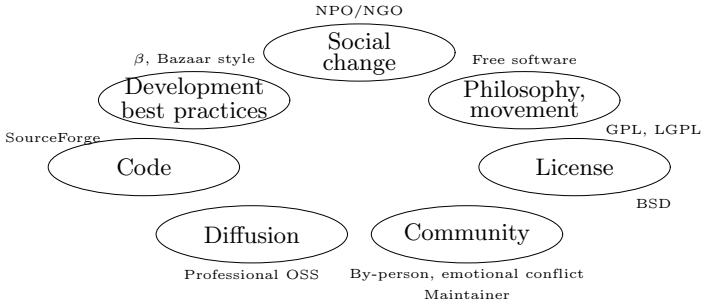


Fig. 8. Seven views of OSS

Table 3. Driving forces of commerce-based economy in OSS communities

Item	Description
Code scale	Large-scale code requires highly structured management.
Industry as user	Coverage of the completed software stacks lead the entire industry to become a customer of OSS component packages, and must have representatives from industry in OSS communities.
Organizational governance	Large-scale OSS solutions require highly organized governance to represent neutrality and compose reasonable roadmaps for stakeholders.
Global awareness	OSS is increasing in width and diversity, therefore, global awareness, usually provided by a global industry leading company, is important to OSS management.

mainly consisting of global industrial players are increasing their presence in OSS communities. The driving forces of this trend are depicted in Table 3.

4.3 Limitations

This research is descriptive and qualitative without any quantitative measures. In particular, this paper lacks any quantifiable measures for the transitions between economy types. Those quantitative approaches are beyond the scope of this paper.

This paper focuses on large-scale and industrial OSS solutions. Although, they are increasingly visible, however, they are a part of an OSS world of diversity and heterogeneity.

OSS projects are diverse and this paper lacks the in-depth analysis of each OSS project.

The impacts on social norms and trusts in OSS communities from these transitions are not covered in this paper.

The detailed business ecosystem engineering driving the proposed transitions is not addressed in this paper.

5 Conclusion

Over the decades, OSS has accomplished many great achievements and penetrated into the entire software industry. OSS has come to occupy the mainstream of the software industry. The success of OSS includes quality, performance, coverage and completeness as well as many established OSS-based foundations with global governance.

This success also highlights the necessity for revisiting the underlying economy types.

The author proposes a transition model from free economy to commerce-based economy. The transition is driven by the completeness of OSS for industrial solutions and the engagement of industry in OSS projects.

This trend does not fit with the free-economy-oriented-ness inherited from the early stage of OSS. However, multiple driving forces are visible to support the transition toward a commerce-based economy as the underlying principle in OSS projects.

The awareness of stages raised by the proposed model will help build mutual understanding between OSS communities; and industrial engagement, which has lead to productive evolution in many OSS projects.

The author presents multiple generation views that fit the proposed transition model in order to provide supplemental support for the transition discussed.

The transition has become visible over a long span of time, even decades. And it happens due to the maturity of OSS projects, and due to a mutual understanding between OSS communities and enterprises. The resulting institutional characteristics such as global awareness and harmonization with industrial solutions can contribute to further productive collaboration between communities and industries in OSS projects.

References

1. Augustin, L.: Why now is the time for open source applications. In: Plenary Speech of OSS 2010, Notre Dame, IN, USA (May 2010)
2. Capra, E., Francalanci, C., Merlo, F., Rossi-Lamastra, C.: Firms' involvement in open source projects: A trade-off between software structural quality and popularity. *J. Syst. Softw.* 84, 144–161 (2011), <http://dx.doi.org/10.1016/j.jss.2010.09.004>
3. Fitzgerald, B.: The transformation of opensource software. *MIS Quarterly* 30, 587–598 (2006)
4. Fitzgerald, B., Agerfalk, P.J.: The mysteries of open source software: Black and white and red all over? In: HICSS 2005: Proceedings of the Proceedings of the 38th Annual Hawaii International Conference on System Sciences, p. 196.1. IEEE Computer Society, Washington, DC (2005)
5. Letellier, F.: Open source software: the role of nonprofits in federating business and innovation ecosystems (January 2008) (a submission for AFME 2008), <http://flet.netcipia.net/xwiki/bin/download/Main/publications-fr/GEM2008-FLetellier-SubmittedPaper.pdf>

6. LiMo Foundation: LiMo Foundation Home page (January 2007), <http://www.limofoundation.org/>
7. Open Handset Alliance: Open Handset Alliance web page (2007), <http://www.openhandsetalliance.com/>
8. Raymond, E.S.: The magic cauldron (August 2000), <http://www.catb.org/~esr/writings/cathedral-bazaar/magic-cauldron/>
9. Subramaniam, C., Sen, R., Nelson, M.L.: Determinants of open source software project success: A longitudinal study. *Decis. Support Syst.* 46(2), 576–585 (2009)
10. Symbian Foundation: Symbian Foundation web page (2008), <http://www.symbianfoundation.org/>
11. The Apache Software Foundation: The Apache Software Foundation web page (1999), <http://www.apache.org/>
12. The Eclipse Foundation: The Eclipse Foundation web page (2004), <http://www.eclipse.org/>
13. Watson, R.T., Boudreau, M.C., York, P.T., Greiner, E., Donald Wynn, J.: The business of open source. *CACM* 51(4), 41–46 (2008)
14. Yamakami, T.: Generations of oss in evolutionary paths: Toward an understanding of where oss is heading. In: *IEEE ICACT 2011*, pp. 1599–1603. IEEE, Los Alamitos (2011)
15. Yu, L., Ramaswamy, S., Lenin, R.B., Narasimhan, V.L.: Time series analysis of open-source software projects. In: *ACM-SE 47: Proceedings of the 47th Annual Southeast Regional Conference*, pp. 1–6. ACM, New York (2009)