

An Exploratory Long-Term Open Source Activity Analysis: Implications from Empirical Findings on Activity Statistics

Toshihiko Yamakami

► **To cite this version:**

Toshihiko Yamakami. An Exploratory Long-Term Open Source Activity Analysis: Implications from Empirical Findings on Activity Statistics. Pär Ågerfalk; Cornelia Boldyreff; Jesús M. González-Barahona; Gregory R. Madey; John Noll. 6th International IFIP WG 2.13 Conference on Open Source Systems,(OSS), May 2010, Notre Dame, United States. Springer, IFIP Advances in Information and Communication Technology, AICT-319, pp.395-400, 2010, Open Source Software: New Horizons. <10.1007/978-3-642-13244-5_37>. <hal-01056035>

HAL Id: hal-01056035

<https://hal.inria.fr/hal-01056035>

Submitted on 14 Aug 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



An Exploratory Long-term Open Source Activity Analysis: Implications from Empirical Findings on Activity Statistics

Toshihiko Yamakami

ACCESS

Toshihiko.Yamakami@access-company.com

Abstract. Open source software (OSS) activities are diverse and difficult to capture. The author attempts a web service-based observation of OSS activities. Small community factor is discussed from a social viewpoint.

1 Introduction

OSS is a multi-faceted process including code, license, community, tools, development process, innovation methodology, philosophy, and best practices. The author attempts an exploratory analysis of project mining of publicly available open source activities. The author performs an analysis of open source activity pattern in a chorological dimension. The author provides a perspective for long-term observation of open source software project activities and implications for social aspect of OSS.

2 Purpose and Related Works

The purpose of this research is to identify the patterns of open source software activities and its implications for social aspect of OSS.

Raymond discussed open source from the business model perspective in this famous open source work series [4].

Ducheneaut discussed the social analysis on a particular open source project from a dynamism viewpoint, how to retain and reproduce a community [3].

Bird analyzed the source code repository and mailing list archive for Postgress [1]. Bird also analyzed community structures of known successful open source projects [2] with the autonomous subcommittee formation.

3 Patterns and Chasms

The patterns are described in Table 1. It should be noted that the many open source projects do not reach the active state. In many projects, they even fail

Table 1. Patterns

Stopped	Never launched	Initiated, but no real open source activity took place.
	Launched and dead	Initiated, and no team was formed.
	Launched, high activity and dead (or complete)	An active team was formed and disbanded.
Active	Launched and one-person	Initiated, and one person keeps the activity.
	Launched and one-company with multiple persons	Multiple person team was formed and the activity continues to be active.
	Launched and multiple-company (or individuals)	Multiple company committed and the activity continues to be active.

to launch the project, therefore, no source code is available to public. It is an interesting research topic how these different states of project have been derived. The different states are identified with different chasms. The chasms patterns are depicted in Table 2.

Table 2. Patterns

Type	Description	Contributing Factors
Launch Chasm	Some projects cannot reach successful launch	Lack of experience, enthusiasm, completeness. Competitive components.
One-person Chasm	Some projects may be never maintained by more than one person	Lack of universality, documentation, use cases.
Use Chasm	Some components may not obtain any real users	Quality, lack of applicability, application use cases.
One-company Chasm	One company committed, but no other organization committed.	Lack of ecosystem. No industry-wide support.

4 Long-term Observation

In order to further examine the current states of open source projects, the author uses web services for long-term open source activity archives. There is a site “ohloh.net” which provides web services interfaces for long-term monthly OSS activity observation.

Observation methodology is as follows:

- code size
- contributing member size

The author analyzed the 77 open source projects automatically stored in the ohlor (<http://www.ohlor.net>). They include Action Script to XSL Transformation. The ohlor started tracking open source codes dated back to 1989. The mean results are shown in Table 3. 1996.3 in year means $1996 + \frac{3}{10}$ (year).

	regular	irregular
Patterns	Synchronized release patterns	No patterns
Code size	steady growth	frequently flat

Table 3. Statistics of open source projects.

Description	Average
Project	3755.7
Contributors per Project	2.2
Year	1996.3
Source lines per Contributor	8755.8
Commitments per Contributor	44.6
Source lines per Commitments	254.1

Contributors per project are depicted in Fig. 1 (a). 53.3 % of projects show $1 \leq x < 2$. 32.5% of projects show in $2 \leq x < 3$. With further detailed analysis, $1.4 \leq x < 2.4$ includes 74.0 % of projects.

This follows the following two observations:

- The bazaar-style open source project follows the basic structure that the one contributor per one project, and
- The bazaar-style open source project follows a main contributor with a sub contributor.

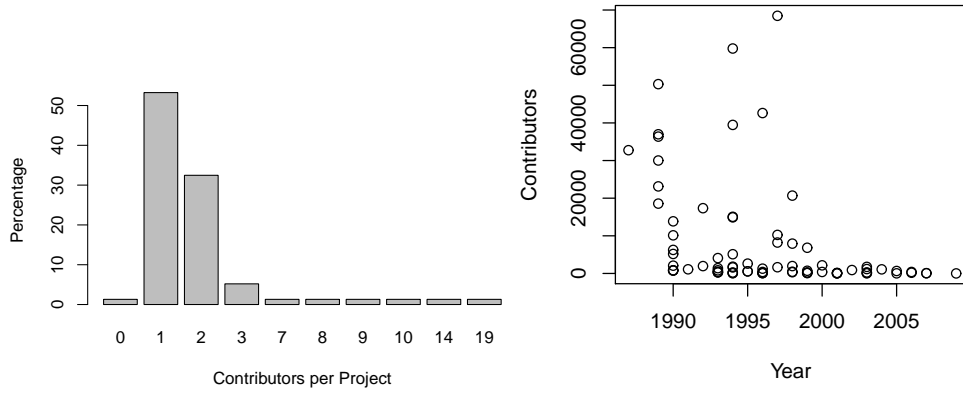
This reveals the following social structure of open source projects:

- In the bazaar-style open source project, it is difficult to coordinate the collaborative software development. Therefore, each contributor consists of a separate project. When a project consists of only one contributor, collaboration is simple.
- Even though to minimize the collaboration effort, people cannot perform the software development in an isolated manner. Social respect and attention is needed. This is reflected in the average $1 + \alpha$ contributors per project.

The year x contributors is plot in Fig. 1 (b). Many fresh projects have small number of contributors. With the older projects, we can see a large fluctuations among projects. The longer life does not directly contribute to the large number of contributors.

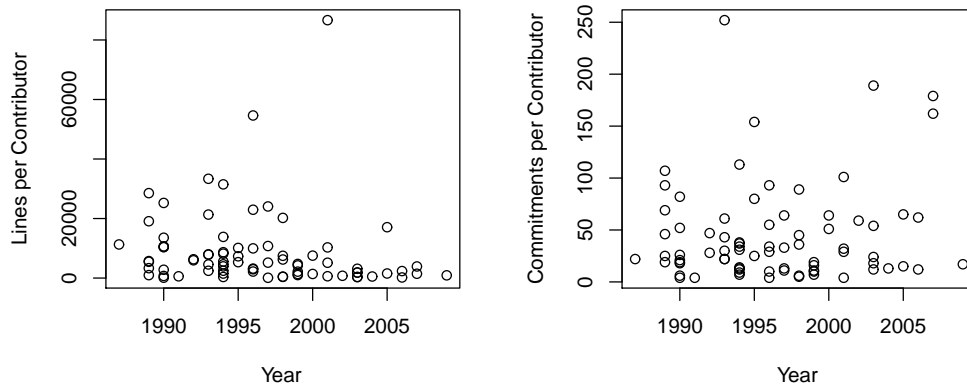
The year x Lines per contributor is plot in Fig. 2 (a). The same results are obtained. The younger projects have less lines, which is natural. With the older projects, we can see diversity. The longer life does not contribute to the large number of codes. The adoption of code and increase activity in code maintenance is a social process to overcome the social chasms.

The year x Commitments per contributor is plot in Fig. 2 (b). The commitment per contributor does not have the significant implication to the project age. The survival of long years does not have a significant impact from a statistical viewpoint, which is natural.



(a) Contributors per project. (b) Year x contributors

Fig. 1. Contributors per project.



(a) Year x Lines per Contributor (b) Commitments per Contributor

Fig. 2. Lines and Commitments per Contributor per project.

5 Discussion

5.1 Skill to leverage open source projects

The skills to leverage open source projects are illustrated in Fig. 3. Most projects are considered to remain in the unskilled leadership domain. It needs a system-

atic management and skill development to foster a productive OSS project. Considering one-man and plus alpha status of many OSS projects, it is considered to be useful to provide case studies of growing projects and skills to harness project growth.

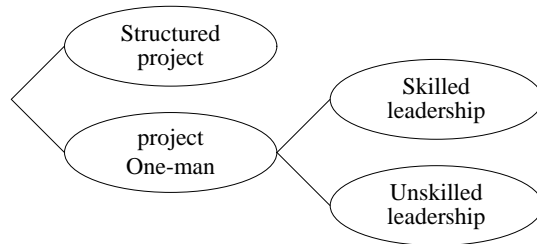


Fig. 3. Skills to leverage open source projects.

5.2 Different types of software for adoption

It should be noted that the type of open source software impacts the adoption. It is influenced by the nature of the source code, whether it is an end user product or a platform product. The success patterns depending on the two types of software are illustrated in Table 4. From the current observation, these two types are not distinguished, but further studies need to examine them.

Table 4. Success Patterns

Application Usage	Completeness	Feature completeness.
	Stability	Stably usable.
	Feature Advantage.	Usability of features.
Middleware Usage	Combinatorial usage.	Middleware applicability in combination of other components or applications.
	Upstream open source application popularity.	Component dependence on popular applications.

5.3 Limitations

This study is based on the external observation of open source project archives. It exhibits an interesting implication for awareness of small-sized fine-grained community in the open source projects, leveraging social respect and self satisfaction. This study is not based on any interviews or detailed motivation analysis, which is a limitation of this study. This study is descriptive, and needs further quantitative analysis on micro-level social adoption.

6 Conclusion

The influence of open source software on the software industry continues to increase. In order to identify the open source activity patterns, the author performs a long-term analysis of open source activities in a chorological dimension.

The author obtains the activity data from the publicly available web services to the open source activity archive. The author presents some exploratory results from project mining of publicly available data. Many open source projects consist of a small group with a small number of human relations to maintain a certain level of social respects. The study reveals the minimum level of social ties in many of the open source projects.

It is useful to leverage awareness of this small-society factor in many open source activities. In order to harness many successful open source projects, it is important to raise awareness for learning from successful projects and skills to depart from one-man projects.

References

1. Bird, C., Gourley, A., Devanbu, P., Gertz, M., Swaminatha, A.: Mining email social networks in postgres. In: MSR'06, pp. 185–186. ACM (2006)
2. Bird, C., Pattison, D., D'Souza, R., Filkov, V., Devanbu, P.: Latent social structure in open source projects. In: SIGSOFT '08/FSE-16, pp. 24–35. ACM (2008)
3. Ducheneaut, N.: Socialization in an open source software community: A socio-technical analysis. *Computers in Human Behavior* **14**(4), 323–368 (2005)
4. Raymond, E.S.: The magic cauldron. <http://www.catb.org/esr/writings/cathedral-bazaar/magic-cauldron/> (2000)