

A Stage Model of Open Source Activities: An Exploratory Analysis on Open Source Repository

Toshihiko Yamakami

CTO Office

ACCESS

2-8-16 Sarugaku-cho, Chiyoda-ku, Tokyo, Japan 101-0064

Email: Toshihiko.Yamakami@access-company.com

Abstract—Open source has impacted software industry since its birth. The magnitude of its impact is increasing day by day in a global scale. It is critical for any software vendor to take an appropriate position in the open-source-involved software engineering. The software engineering metrics needs a social dimension to cope with open-source-involved context. After more than a decade of open source activities, open source is still on the stage of identifying its position in the software industry from the quality control, project management and business model perspectives. In this paper, the author attempts to identify the macro view of the open source activities from the publicly available activity logs.

I. INTRODUCTION

The wide-spread information communication technology enables a flexible form of collaboration. Open source software is one of the deliverables of this technology trend. It shakes the software industry when they found that open source software was valid in commercial use. As technologies advance, it has become clear that open source software occupies a significant portion of software engineering. It provides a path for many open source software to grow. The author provides a perspective for long-term observation of open source software project activities and implications for software quality and adoption strategies.

II. PURPOSE AND RELATED WORKS

A. Purpose of Research

The purpose of this research is to identify the patterns of open source software activities and their implications to quality evaluation and adoption strategy.

B. Related Works

There are emerging foundation engineering utilizing open source software in the mobile platform software [1] [2] [3].

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

As the open source software establishes its standpoint in the industry, the business models have been discussed in research and industries.

The success of Linux brought attention to the business models. Fink discussed the business and economics of Linux [5].

Baird discussed the heterogeneous world of proprietary and open-source software [6]. Watson discussed the second generation open source software (professional open source software) [7].

Bonaccorsi discussed the hybrid model [8].

The utilization of OSS in the industry environment based on ISO/IEC 9126 was discussed by Sohn [9].

Doroles discussed the open source acceptance from the attitude formation perspective [10].

Sacchi discussed the free/open source software development work practices [11].

Ducheneaut discussed the social analysis on a particular open source project from a dynamism viewpoint, how to retain and reproduce a community [12].

Bird analyzed the source code repository and mailing list archive for Postures [13]. Bird also analyzed community structures of known successful open source projects [14] with the autonomous subcommittee formation.

III. METHOD

A. Method

The author performs an exploratory analysis of open source activities.

The following analysis is performed:

- Registration year analysis
- Download pattern analysis
- Age pattern analysis

B. Data Set

For sampling analysis, the site SourceForge (<http://www.sourceforge.net>) is used for analysis. Source Forge is a popular site for open source repository, active since 1999. It hosted 156,077 projects on Sep 28 2009. There are more than million registered users for this site. More than 6,000 projects are added annually to this site. This means that the number of projects increases almost every hour. So the number of 156,077 was just an approximation at some

snapshot. SourceForge provides a tool to navigate the open source projects. Using this tool, more than 60,000 projects are shown not to have recent source registered on this site. This is confirmed by manual browsing with 10,000 interval probes.

The author created a simple crawler to get the first 156,000 high ranking project. The remains 77 projects with the lowest rank projects are ignored. It is because that the lowest rank shows the inactiveness of the projects.

The crawler was coded by R2.9.0. The simple crawler took approximately 2 hours to get the data.

Initial crawl obtained 66,218 data out of 100,000 projects. A curl-based off line crawler was tested, however, 30–35 % data were lost. This is caused not by the R implementation. The author considered it as a network issue. In this paper, the 65–70 % data are used because the data drop was occurred in a non-arbitrary way beyond the author control in this environment.

156,000 project crawls were performed and 93,181 project data were recorded as results.

IV. OBSERVATIONS

A. Registration Year Analysis

The registration year distribution in active 48,684 projects is depicted in Table I. The oldest registration started in 1999 in this site.

The registration year distribution among non-active 44,497 projects is depicted in Table II. This shows the distribution among the projects without any source code availability. It is considered that these projects had source in the past, but lost the downloadable source code over a span of time.

The earliest one dates back to 1999. Projects registered in 2009 have possibility that it will start active operations in the near future. It is a rough indication of even-not-launched project distribution from one type of samples.

B. Download Pattern Analysis

The download distributions among the projects with source code availability are shown in Table III. This is the result from the download logs of 48,684 projects. The project launch date varies from 1999 to 2009. Therefore, it is a rough snapshot of the current situation.

The two 100-millions downloads are Emule and Azureus, both of which are P2P file sharing client software. This distribution is a combination of one for developer use and one for end user use. When software becomes mature, there is a higher possibility that end user download outnumbers. Depending upon software category, some of the middleware components are for developer only. The detailed analysis of these factors is not performed in this paper.

The author assumes the following method following the law of large numbers conjecture:

Chasm identification using Gap Analysis to the Law of Large Numbers There are a large number of projects with different maturity and life cycle stages. In a certain range of downloads, the law of large

numbers is applied and the number of projects per download range is consistent. At some point, the download distribution departs from this statistically stable condition, where the chasm for activities exists.

The consistent status in the range of small download numbers is a non-verified conjecture. This paper follows this method following this conjecture.

The distribution in the more fine-grained analysis is given in the range of $n < 12$ to identify the peak out point. This is based on the download span of $2^{n-1} \leq \text{download} < 2^n$. It indicates a peak out point at 2^8 where downloads are between 128 and 255. This is an exponential table, therefore, each column has twice as many possible values as the left neighbor column. When the distribution is random, it leads to a double download value compared to the left neighbor one. From 1 to 7, the double trend continues. It can be considered to be similar to a random distribution. The table shows the non-random distribution from 8. The author considers that there is some new threshold around here. It is assumed to be an outcome of some social factors in open source community development, however, this macro level analysis does not expose any direct evidence to discuss it. We may consider some open source community incubation process around here. However, it should be noted that any reasonable discussion needs further gathering of empirical data.

The first chasm for open source projects is a successful release of real source code. Half of the projects failed to cross this chasm. The second chasm is social acknowledgement. Using gap analysis to the early consistent stage, the author assumes the second chasm around 2⁸ downloads.

C. Age Pattern Analysis

Code age is calculated from the number of month from the project registration to the most recent source code update available on the SourceForge site. The month age pattern among projects with less than one year is depicted in Table V. There are 30,568 projects with less than one year life with available source. Two thirds of them are less than 3 months in existence. It should be noted that the age means the length from registration date to the most recent update date. Therefore, project registered in May 2003 and most recent update was in June 2003, then, the length is one month.

The year level age distribution is shown in Table VI. Year 0 means the project history is less than 1 year. Year 1 means the project history is equal or larger than 1 year and less than 2 years. 86.6 % of projects have less than 3 year life. It should be noted that the young projects recently registered have naturally short life. There are more than 16,000 projects registered in 2008 and 2009 with active source code. The number of 30,568 of 0 year history shows that half of those projects were registered before 2008 and are not updated recently.

TABLE I
REGISTRATION YEAR DISTRIBUTION IN ACTIVE PROJECTS.

year	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009
projects	192	2,560	3,818	4,674	5,059	4,906	5,184	6,340	6,145	5,822	3,984

TABLE II
STARTING YEAR DISTRIBUTION IN NON-ACTIVE PROJECTS.

year	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009
projects	3	191	992	1,114	1,156	2,176	2,399	8,722	10,643	10,896	6,205

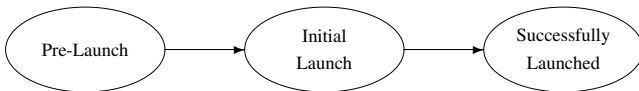
TABLE III
DOWNLOAD DISTRIBUTION IN ACTIVE PROJECTS ($10^{n-1} \leq \text{download} < 10^n$).

n	1	2	3	4	5	6	7	8	9
projects	706	6,516	19,914	14,607	5,333	1307	257	42	2
percentage	1.45	13.38	40.90	30.00	10.95	2.68	0.53	0.09	0.00

TABLE IV
DOWNLOAD DISTRIBUTION IN ACTIVE PROJECTS ($2^{n-1} \leq \text{download} < 2^n$) IN $n < 12$ RANGE.

n	1	2	3	4	5	6	7	8	9	10	11
projects	88	192	289	575	1,188	2,397	4,193	5,632	6,386	6,384	5,517
percentage	0.27	0.58	0.88	1.75	3.62	7.30	12.77	17.15	19.45	19.44	16.80

Fig. 1. A three-stage model of open source project activities.



V. DISCUSSION

A. Three-Stage Model

A three-stage model of open source project activities is depicted in Fig. 1. Even when a project is registered in SourceForge, 50 % of them do not reach the source availability stage. This state is described as “pre-launch”. After the source code publishing, 50 % of projects stay under 1,000 download. In the figure, this is marked as “initial launch”. The projects go in the “successful launch” when it exceeds the 1,000 download chasm. Among these projects, only 0.6 % goes to the “star projects” with more than 1,000,000 downloads. The successful launched project passes the second chasm of social acknowledgement.

B. Success Factors

After the three-stage model is identified, the critical issue is to identify success factors in the projects passing through the chasm and reach “successfully launched” status.

There are two conditions to be considered:

- project maturity, and
- activeness.

The download distribution in the projects starting by 2006 is depicted in Table VII. There are 15,062 projects. Source from more than 8,815 projects are not currently available on this source repository. 3,236 projects have most recent source update in 2006, 1,426 projects in 2007, and 818 projects in 2008.

The download distribution in the projects actively updating in 2009 is depicted in Table VIII.

The download distribution in the projects starting by 2006 and actively updating in 2009 is depicted in Table IX.

C. Advantages

Open source project is a social process with a certain level of culture and philosophy. This is unique in the field of software development. At the same time, many open source projects are based on people-dependence. Governance, roadmap and quality control are controlled by the maintainers, heavily depending personality and personal skills of them with some exceptions of large open source projects.

This leads to many emotional and political discussion on open source projects. With the proliferation of open source projects and increased penetration to the software business, it is important to establish quantitative measures to identify and assess open source projects.

This is an early attempt to understand the open source projects in macro-level and quantitative measures.

The open source software development is a combination of technical and social analysis. The further studies on this macro-level analysis and cross-project comparison are important to provide an objective clues to improve open source project management.

D. Applications

Open source activities are difficult to assess and control. Many businesses seriously consider open source software component adoption. In order to secure this adoption, quality control and quality assessment of open source activities are crucial.

The long-term analysis and cross-project comparison enables this control and assessment in order to leverage the

TABLE V
THE MONTH AGE DISTRIBUTION WITHIN THE LESS THAN ONE YEAR PROJECTS WITH AVAILABLE SOURCE.

months	0	1	2	3	4	5	6	7	8	9	10	11
projects	10,536	6,160	2,837	2,006	1,644	1,398	1,211	1,094	1,047	997	860	778
percentage	34.47	20.15	9.28	6.56	5.38	4.57	3.96	3.58	3.43	3.26	2.81	2.55

TABLE VI
THE YEAR AGE DISTRIBUTION IN THE ACTIVE PROJECTS.

year	0	1	2	3	4	5	6	7	8	9
projects	30,568	6,421	3,557	2,211	1,469	1,005	688	416	315	119
percentage	65.36	13.73	7.61	4.73	3.14	2.15	1.47	0.89	0.67	0.25

TABLE VII
DOWNLOAD DISTRIBUTION IN ACTIVE PROJECTS STARTING BY 2006 ($10^{n-1} \leq \text{download} < 10^n$).

n	—	1	2	3	4	5	6	7	8
projects	8,722	47	783	2,948	1,927	522	92	20	1
percentage	57.91	0.31	5.20	19.57	12.79	3.47	0.61	0.13	0.01

TABLE VIII
DOWNLOAD DISTRIBUTION IN ACTIVE PROJECTS UPDATING IN 2009 ($10^{n-1} \leq \text{download} < 10^n$).

n	—	1	2	3	4	5	6	7	8	9
projects	7	99	1,620	2,840	2,166	1,236	562	169	33	2
percentage	0.08	1.13	18.55	32.52	24.80	14.15	6.43	1.93	0.38	0.02

potentials of open source project. Visualization of cross-project comparison helps each open source activity improve their governance and project control.

E. Limitations

This paper is based on an exploratory observation of 1999–2009 Source Forge publicly open data. SourceForge is one of the most popular open source project hosting sites. However, it is just a snap shot of this particular site.

Also, due to the HTTP connection irregularity, not the complete dataset was analyzed in this paper. Two thirds of the total data are believed to be significant, however, further research with the exhaustive dataset can expose more accurate analysis.

It should be noted that the observation was performed as it is. There is some irregularity on the available data. For example, a small number of projects have older source code update date than registration date. When the source code availability was lost after some downloads, some projects show download numbers with "N/A" at the most recent update date. With the nature of this publicly available data observation, it is unavoidable to deal with this minor anomaly. The magnitude of impact due to these irregularities are estimated to be smaller than 0.1 %.

This research assumed simple measures to check the registration date and the most recent update date. These are the rough indicators for project maturity and activeness. The further in-depth analysis on the project maturity and degree of active updates may expose further findings on the open source projects.

This research does not take any consideration on the project size or code size. The consideration on these size factors may lead to the further analysis.

This research does not take any consideration of the quality of downloads. The download can be end-user downloads or source code developer downloads. There is no measure available for these different audience at the moment. Also the download is just a number in this research. The quality and characteristics of source code developer community or end user community are for further studies.

VI. CONCLUSION

There are multiple aspects of open source activities. From one perspective, it is a kind of philosophy on computer software. From another perspective, it is a new software development process or methodology using global IT infrastructure. There are a wide variety of characteristics on community development and use among open source software.

The author attempted to identify a stage-based view on the open source project evolution. The author performed observation of publicly available long-term open source project logs. From this exploratory analysis, it is indicated that a half of open source projects do not reach the source code publication status. When a project starts public source code publication, a half of them stay in the low level distribution. After a project reaches a certain active publication, only a very small number of projects reach a star status, with more than 1,000,000 downloads. It is only 0.6 % of all projects.

The author then performed an exploratory analysis on the impact of project maturity and project activeness on the download numbers. From this analysis, maturity and activeness are not important when the other factor is not present. With both maturity and activeness on code update are supported, the positive result on the increased distribution is shown.

The research reveals some of the dynamism of open source activities in a long span of time. This paper shows the two

TABLE IX
DOWNLOAD DISTRIBUTION IN ACTIVE PROJECTS STARTING BY 2006 AND UPDATING IN 2009 ($10^{n-1} \leq \text{download} < 10^n$).

n	1	2	3	4	5	6	7	8
projects	1	1	21	151	311	197	58	15
percentage	0.13	0.13	2.78	20.00	41.19	26.09	7.68	1.99

major chasms in the open source project evolution. It is critical how to manage the two chasms in project evolution for each project.

The future research includes in-depth analysis chasm management for open source development and cross-project comparisons.

REFERENCES

- [1] LiMo Foundation, "LiMo Foundation Home page," <http://www.limofoundation.org/>, January 2007.
- [2] Symbian Foundation, "Symbian foundation web page," Available at: <http://www.symbianfoundation.org/>, 2008.
- [3] Google, "Android - an open handset alliance project," <http://code.google.com/android/>, November 2007.
- [4] E. S. Raymond, "The magic cauldron," <http://www.catb.org/esr/writings/cathedral-bazaar/magic-cauldron/>, August 2000.
- [5] M. Fink, *Business and Economics of Linux and Open Source*. Prentice Hall Professional Technical Reference, October 2002.
- [6] S. A. Baird, "The heterogeneous world of proprietary and open-source software," in *ICEGOV '08*. ACM, December 2008, pp. 232–238.
- [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] A. Bonaccorsi, S. Giannangeli, and C. Rossi, "Entry strategies under competing standards: Hybrid business models in the open source software industry," *Management Science*, vol. 52, no. 7, pp. 1085–1098, July 2006.
- [9] S. Y. Sohn and M. S. Mok, "A strategic analysis for successful open source software utilization based on a structural equation model," *Journal of Systems and Software*, vol. 81, no. 6, pp. 1014–1024, June 2008.
- [10] M. D. Gallego, P. Luna, and S. Bueno, "User acceptance model of open source software," *Computers in Human Behavior*, vol. 24, no. 5, pp. 2199–2216, September 2008.
- [11] W. Scacchi, "Free/open source software development," in *ESEC-FSE '07*. ACM, September 2007, pp. 459–468.
- [12] N. Ducheneaut, "Socialization in an open source software community: A socio-technical analysis," *Computers in Human Behavior*, vol. 14, no. 4, pp. 323–368, August 2005.
- [13] C. Bird, A. Gourley, P. Devanbu, M. Gertz, and A. Swaminatha, "Mining email social networks in postgres," in *MSR'06*. ACM, May 2006, pp. 185–186.
- [14] C. Bird, D. Pattison, R. D'Souza, V. Filkov, and P. Devanbu, "Latent social structure in open source projects," in *SIGSOFT '08/FSE-16*. ACM, November 2008, pp. 24–35.