Influences on Developer Participation in the Debian Software Ecosystem

Eric Ververs
Utrecht University
(+31)0653189267
l.j.ververs@students.uu.nl

Rick van Bommel
Utrecht University
(+31)0623324926
r.a.a.vanbommel@students.uu.nl

Slinger Jansen
Utrecht University
slinger@slingerjansen.nl

ABSTRACT

Nowadays, more and more open source software developers are starting to create software in decentralized communities. For these software ecosystems and their many end-users it does not always guarantee contributions where needed. This paper maps the influential factors that determine developer participation with a case study on the open source software Debian website. Data was gathered on potentially influential events and all commits in the year 2000 to 2011. The two previous data results are compared in a correlation study to derive relationships. In 11% of the types of events that were studied, there was a reoccurring relation with in- or decreased developer participation. Ecosystem project leaders can stimulate or prevent these events to prevent missing in action developers and welcome new developers.

Categories and Subject Descriptors

None.

General Terms

Management, Measurement

Keywords

Software ecosystem, developer involvement, developer activity, developer participation, open source software, free software, Linux, Debian.

1. INTRODUCTION

Software development can be divided in two types, i.e., centralized and decentralized. Examples of centralized software developers are Red Hat and Microsoft. An example of a decentralized software project is Debian. Decentralized and open source projects are not necessarily less capable of creating large sized distribution as the centralized projects [1]. However, especially in open source and decentralized software projects, it does not always guarantee contributions where needed. Jansen et al. [2] proposed a study on the orchestration of ecosystems. Jansen et al. define in [2] a software ecosystem as: "a set of businesses functioning as a unit and interacting with a shared market for software and services, together with the relationships among them". Messerschmitt and Szyperski [3] describe all the aspects of the software ecosystem in detail. Not uncommonly do decentralized Open Source Software (OSS) projects not provide financial rewards to stimulate participation in development [4]. This understudied type of rewarding exposes decentralized OSS projects to a risk that centralized and paid development projects do not entail. If fewer developers participate in developing OSS, the continuation of the OSS project is at risk. In order to regain Missing In Action (MIA) developers, or to attract new developers, it is important to know what events influence developer participation. By reusing the positively affecting influential events mapped in this paper, decentralized OSS project orchestrators (such as project leaders) can more effectively increase developer involvement in their projects.

Following up on this background, this paper targets to map the relations between influences and developer involvement in terms of the software ecosystem by answering the following question:

What factors influence developer participation in the Debian software ecosystem?

The research is divided into the following sub questions:

- What is the total number of developers and average number of commits in each week?
- Which events can potentially influence developer participation?
- What reoccurring relations can be found between events and activity in the average software commits and total developer participation?

In order to identify these relations, a case study is performed on the decentralized OSS Linux distribution Debian website. In May 2011, Debian had 1643 developers, which were spread out over the world [5, 6].

The participation of developers can be researched by looking at what drives OSS developers in the development process, as described in a research performed by Hars and Ou [7]. This research was not conclusive, because it only looked at personal motivations, instead of outside factors.

Roberts et al. [8] went a step further by describing the relation between motivation, participation and performance (by studying the quality of the submitted code) of OSS developers. In practice, this usually means looking at the commits. Goeminne and Mens [9] define a commit as: "atomic changes done on the source code".

Merely investigating the quality of source code, is not sufficient according to a recent qualitative framework design and study on ecosystem evolution [9]. In order to fully understand the workings of an ecosystem the community needs to be studied as well. For example, this can be done by looking at bug trackers, mailing lists and forums.

The scope is this paper is the Debian ecosystem of the last 11 years (2000-2011). Based on the acquired information, an analysis can take place to see if there are any reoccurring relations between events and developer participation changes.

Data obtained from the Debian software version control system (Git) will be used to find new relations between events and developer participation. The structure of the paper first details the chosen research approach, after that the results from the three sub questions leading towards the last results chapter, where the main research question is answered. Some discussion is opened to elaborate on uncertainties, newly derived questions and applicability of this research.

2. RESEARCH APPROACH

In this research, a case study is performed on the OSS project Debian. Debian is an OSS operating system for desktops and servers. Development of Debian is a collaboration between developers (1600+) working from all over the world. What makes Debian special compared to other software projects, is the

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

MEDES'11, November 21-24, 2011, San Francisco, USA.
Copyright© 2011 ACM 978-1-4503-1047-5/10/10...$10.00.

capability to create an OS including 29.000 packages (a lot from separate smaller projects within the Debian community), while using a decentralized development team.

The methods outlined later in this chapter are aimed towards the use of the Debian Git repository and Debian project website to gather data from the year 2000 to 2011. The approach was chosen to forgo personal interpretations of developers and since it would otherwise only be possible for a select part of the total developer community. The Debian Git repository and Debian website, on the other hand, hold a sizable range of historical data of past developer activity and events.

Data mining in this paper targets the following two parts: Debian Git repository and the Debian website. In order to complete the first part, a Linux machine was set up and two Git commands for each developer project were issued in the Bash terminal:

```
git clone http://git.debian.org/git/[project]
git log >> [project].txt
```

Respectively resulting into the download of all Debian projects into one local folder and an extract of all Git commit logging data into a plain text file. The plain text files containing the log data on the developer activity were then converted to a Structured Query Language (SQL) database by using a small Hypertext Preprocessor (PHP) script containing various regular expression matching functions (preg_match) specifically applied for this task..

The second part of the data mining consists of looking for events that could potentially influence developer participation. For that the Debian website was used as a source. All the events together with their corresponding category are also stored in the same SQL-database as the data from the first part. After all the data was collected and categorized, the database was queried to find a relation between the categorical events and developer participation. To find this relation, the Pearson linear correlation co-efficient was calculated for each category of events in the range of four weeks before and after the event occurred. This statistic measures the strength and direction of a linear relationship between two variables [10].

Eventually in order to evaluate the results, the running Debian leader (2011) checked this paper, and gave feedback on the accuracy and the scope of the research method and results.

3. RESULTS

3.1 Total number of developers and average number of commits in each week

The Debian Git repository holds key historical information on software revisions. For the purpose of this study, the Git database was downloaded and aggregated into an average developer commit and total number of developer perspective. Time intervals were set at one week, and data was collected from 2000 to 2011. In total, 1.000.000 commits were downloaded and used for this paper.

A clearly noticeable change in mid-2009 define a continuing decrease in total commits by the Debian developers, which is still ongoing in 2011, when the data was extracted.

A more definite answer can, however, be given for multiple identical events of the same category which are dispersed over 10 years. Such studies can, to a lesser degree, be influenced by de- or increasing trends in the worldwide Debian developer community. This is exactly what this study is focussed on.

3.2 Potentially influential events

The Debian project is transparent in its information sharing regarding internal and external events, making it a suitable matter for study. The study was aimed to recognize and gather events that could potentially influence the level of activity of the Debian ecosystem. Each event was stored together with the corresponding date, category and week number. In total 116 were deemed feasible. These were extracted from the following Debian website sources:

- Mailing lists (lists.debian.org), announcements relevant for developers such as frozen releases. General announcements regarding the Debian project and community spin-offs that take form as new projects. Furthermore, a large effort is put into matching enthusiastic and capable developers with projects that were abandoned. Such events can influence the activity of developers. They can for example be the root-cause for new developers to sign up or contribute in regaining activity to existing projects;
- Bug tracking system (bugs.debian.org), some release-critical bugs might spur the developer community to become more active. Especially when a general release is around the corner. A bug report is published in the mailing list that shows the current number of critical bugs, essentially depicting the degree of preparedness for a new general release. Such events might play a vital contribution to participation;
- Quality Assurance (QA) group (qa.debian.org), software quality and continuity are aspects which the QA group monitor and steer. Missing developers, dependency issues and mass bugs filings that could have a great impact on the developers are particularly interesting for this study;
- Wiki (wiki.debian.org), release notes, pending issues, upgrades and a wide variety of project information, which might lead to some influential events;
- Event list (debian.org/events/), Debian hosted multiple events for community groups, for the developer group a Debian conference is hosted. In addition a work camp is organized to stimulate cooperation between developers that would not do so otherwise. Furthermore, an open day is held to introduce the Debian project to anyone interested. These activities could help in gaining more members in addition to increasing activity and commitment with current developers;
- Voting area (debian.org/vote/), general resolutions have marked important events in the past. Some of which could potentially influence the degree of participation.

The sources previously mentioned, led to the formulation of the following potentially influential categories in which the events were placed:

- Stable release, a pending stable release can be influential in a way that it stimulates developers to put in extra efforts, or when the release is over to stay inactive for a while because their projects are not likely to reach the public anytime soon;
- Change of Leadership, each period a new leader is elected. This could for example influence developer morale;
- Community spin-off, the community provides significant input that had effect on the Debian project. With these events working as a spring board for new developments in the project, it is not unlikely that some have worked in favour of developer participation;
- Incidents, some notable incident were found such as a fire that burnt a major operations server;
- Debian Conference, increased morale and cooperation that might occur after these events is not unlikely to influence the participation of developers;
- CeBit, CeBIT is one of the most important computer exhibitions. New people are introduced to the Debian project which might lead to increased participation;
- Debian Day, several speakers talk about current developments. Furthermore, a Debian conference which is open to anyone that is interested in free software;
- End of support for Debian release, once a release is out-dated, the support ends, which could potentially influence developer participation;
- Annual Southern California Linux Expo, this exposition brings together open source companies, Linux, developers and users. Perhaps converting users to enter the developer community;
- Awards, Debian has been nominated and chosen numerous times as best project by magazines and websites;

- General Resolution (GR), some examples of GRs are: electing a new leader, changing developer regulations and many more which can be found at the Debian voting area;
- Major bug, major bugs can be for example bugs that hinder vital functionality of the system. Such bugs need to be fixed as soon as possible and might cause increased activity shortly after;
- Dependency issues, some events were seem that had influence on hundreds of packages that depended on it. For example, when the developer changed its popular package, a lot of developers that had packages that depended on that package had to change theirs;
- Release frozen, a frozen release marks the end of any new package contributions. By doing this, the Debian community can focus on the upcoming stable release, rather than introducing new packages;
- Introduction of developer service, making the job of the developer easier, allowing them to do a better job in a shorter time, resulting in increased participation.

In order to make sure that all the important resources were consulted and the most important events are included in this research, the current leader of Debian was consulted.

3.3 Reoccurring relations between events and activity in developer participation

Based on the data in chapter 3.1 and 3.2, in this chapter relationships between influences and activity in developer participation will be identified using statistical analysis.

There are some influences that are happening at the same time or short after on another. This will create noise in the research results because it cannot be determined based on what particular influence the developer participation was affected. Therefore, the influences are divided to fifteen categories. If influences are now taking place short after one another, the combination of the influences in the category will compensate for the potential disturbances that take place. This will lead to the reduction of noise in the research results.

An influence can affect developer participation in the future and the past. For example, there could be a deadline for uploading commits before a conference. But it is also possible that developers will start submitting commits after discussing with other developers at that conference. To find all these relationships, every week, from 4 weeks before, to 4 weeks after the event were investigated for changes. 4 weeks were chosen, because it is the average time between individual events.

As mentioned earlier, the developer participation is split up in two perspectives, i.e., the total amount of developers and average commits per developer. When looking at the amount of developers, it can be determined if a certain influence resulted in more or less active developers. But the shortcoming of only using that measurement is that it only says something about the community as a whole rather than the degree of participation of individual active developers. To overcome this problem, the measurement of average commits per developer will also be determined.

To determine what effect the influences had on the two perspectives, the Pearson linear correlation co-efficient was used [10]. This calculation is suitable for this particular situation, because it can be used to show the strength and direction of a linear relationship between multiple sets of variables. In order to determine the correlation of a point in time prior to the influence, the variables (average commits per developers or amount of developers that submitted a commit) of that point in time and the variables of the time of the influence are used. To calculate the correlation of future points in time compared to the time the influence took place, a similar approach is used, but with a varying timeline. Because of this, the same correlation number means something different for past and future events.

The results of applying the correlation to all the categories of influences can be found in table 3. To interpret these results, some sort of scale is required. The scale proposed by D J. Rumsey, PhD [10] to interpret the correlation is used in this research.

The result tables show the categories per week where the correlation was either positive (+0,30 or higher) or negative (-0,30 or lower). Because the developer participation is split up in two groups, there are also two result tables. Table 1 shows the correlation of the average commits per developer in a particular week. Table 2 depicts the correlation of developers that submitted commits.

Table 1. Correlation of the average commits per developer in a particular week

Time	Category	Correlation
4 weeks before influence	Incidents	0,353
	Release frozen	0,320
3 weeks before influence	CeBit	-0,383
	General Resolution	0,344
	Introduction of developer service	0,605
	Release frozen	0,310
2 weeks before influence	CeBit	-0,370
	Debian Day	0,414
	General Resolution	0,319
	Incidents	-0,351
	Major bug	-0,362
1 week before influence	Dependency issues	-0,325
	Introduction of developer service	0,607
1 week after influence	Introduction of developer service	-0,574
2 weeks after influence	Debian Day	-0,344
	Dependency issues	0,482
	Stable release	0,338
3 weeks after influence	Dependency issues	-0,330
	Incidents	0,583
	Introduction of developer service	-0,369
4 weeks after influence	CeBit	0,334
	Debian Day	0,495

Table 2. Correlation of developers that submitted commits

Time	Category	Correlation
4 weeks before influence	Introduction of developer service	-0,484
3 weeks before influence		
2 weeks before influence		
1 week before influence		
1 week after influence	Introduction of developer service	-0,369
2 weeks after influence	Introduction of developer service	-0,301
3 weeks after influence		
4 weeks after influence	Introduction of developer service	-0,405

Based on these tables, the following results can be derived from the correlation of the average commits per developer in a particular week:

- The categories Annual Southern California Linux Expo, awards, change of leadership, community spin-off, Debian conference, end of support for Debian release have no effect on the activity of developers;
- 2-3 weeks before and 4 weeks after CeBIT started there was an increase in developer activity;
  2 weeks prior to Debian day, there is a decrease in developer activity. After Debian day there is a decrease in developer activity with the lowest point 2 weeks after Debian day. After

Table 3. Correlation table

Categories	Amount of influences	Linear Correlation Coefficient															
		4 weeks before influence		3 weeks before influence		2 weeks before influence		1 week before influence		1 week after influence		2 weeks after influence		3 weeks after influence		4 weeks after influence	
		Average commits per developer	Amount of developers that submitted a commit	Average commits per developer	Amount of developers that submitted a commit	Average commits per developer	Amount of developers that submitted a commit	Average commits per developer	Amount of developers that submitted a commit	Average commits per developer	Amount of developers that submitted a commit	Average commits per developer	Amount of developers that submitted a commit	Average commits per developer	Amount of developers that submitted a commit	Average commits per developer	Amount of developers that submitted a commit
Annual Southern California Linux Expo	7	-0,286	0,137	0,242	0,088	0,088	0,070	-0,161	0,095	0,117	-0,037	0,008	-0,074	0,069	-0,003	-0,074	-0,060
Awards	12	0,144	0,095	-0,039	0,025	-0,186	0,013	-0,063	0,047	0,064	0,018	0,221	-0,001	0,141	0,019	-0,038	-0,004
CeBIT	9	-0,290	0,044	-0,383	0,021	-0,370	0,004	-0,199	-0,006	0,189	-0,016	0,228	-0,021	0,151	0,084	0,334	-0,036
Change of leadership	8	0,083	-0,039	0,006	-0,043	-0,050	-0,101	0,007	-0,017	-0,040	0,024	0,062	0,040	0,076	0,024	0,017	0,011
Community spin-off	17	-0,181	-0,028	-0,113	-0,020	-0,038	-0,007	-0,093	-0,013	0,064	0,028	0,114	0,054	0,101	0,055	0,058	0,080
Debian Conference	11	0,042	-0,015	-0,088	-0,017	0,038	0,009	-0,048	-0,002	0,166	0,052	0,015	-0,024	-0,072	0,008	0,140	0,000
Debian Day	4	-0,233	-0,136	-0,200	-0,162	0,414	-0,053	0,102	-0,079	-0,062	0,085	-0,344	0,085	0,111	0,071	0,495	0,138
Dependency issues	3	-0,092	0,055	-0,257	0,028	-0,152	0,042	-0,325	0,097	-0,044	-0,028	0,482	-0,073	-0,330	-0,008	0,028	0,057
End of support for Debian release	5	0,169	0,037	0,045	0,046	-0,184	0,041	0,108	-0,001	-0,033	0,060	-0,231	0,028	0,001	0,056	-0,037	0,042
General Resolution	18	0,064	0,107	0,344	0,102	0,319	0,094	0,099	0,086	-0,002	-0,023	-0,060	-0,121	-0,300	-0,174	-0,164	-0,049
Incidents	3	0,353	0,094	-0,085	0,061	-0,351	0,244	0,250	0,217	0,200	-0,028	0,225	0,009	0,583	-0,060	-0,043	-0,088
Introduction of developer service	2	-0,287	-0,484	0,605	-0,213	-0,256	0,051	0,607	-0,153	-0,574	-0,369	0,009	-0,301	-0,369	-0,280	-0,175	-0,405
Major bugs	6	0,222	-0,019	0,203	-0,056	-0,362	-0,024	0,234	-0,064	-0,164	0,032	-0,151	-0,013	0,229	-0,046	-0,251	0,044
Release frozen	5	0,320	-0,101	0,310	0,016	0,185	-0,032	0,136	-0,061	-0,067	-0,036	-0,194	-0,085	-0,153	0,021	-0,279	-0,012
Stable release	6	-0,186	0,001	-0,207	0,028	-0,047	0,029	-0,119	-0,009	0,170	0,086	0,338	0,033	0,167	0,033	0,297	0,015

: Correlations with a weak, moderate or strong positive or negative relationship

these 2 weeks the developer activity significantly rises again with the highest point after the event at 4 weeks;
- 1 week prior to dependency issues there is an increase in developer activity, 2 weeks after there is an increase, followed by a decrease in the third week;
- 2-3 weeks before a general resolution there is a decrease in developer activity;
- Incidents have increased developer activity 3 weeks after the incident occurred;
- 2 weeks prior of major bugs, there is a weak increase in developer activity;
- 3-4 weeks before a release is frozen, there is a decrease in developer activity;
- 2 weeks after a stable release, there is a weak increase in developer activity;
- The introduction of developer services leads to a decrease in developer activeness 3 and 1 week before and leads to a decrease in developer activeness in week 1 and 3 after the influence.

In the joined or hosted events, Debian works on introducing the project to new and unfamiliar developers and users, which could be a probably cause for increased participation. However, during the study of the Debian community it became clear that the new developer approval process tends to take months instead of weeks. It is therefore more likely that groups of developers that were less active, started to become more active, or that already active groups started to become more active. The conclusions based on the correlation of weekly active developers are as follows:

- The categories Annual Southern California Linux Expo, awards, CeBIT, change of leadership, community spin-off, Debian conference, Debian day, dependency issues, end of support for Debian release, general resolution, incidents, introduction of developer service, major bugs, release frozen and stable release have no effect on the amount of developers that submitted commits;
- Only the introduction of developer services seems to have a reasonable effect on the amount of developers that submit a commit. 4 weeks before the influence, there is an increase in the amount of developers. The first 4 weeks after the event all show a decrease in the amount of developers, with a peak in the first week after the event.

And finally, the conclusions about the entire research in general:

- Influences have more effect on the degree of participation by developers, rather than causing more or less developers to become active;
- In 98,32% of all cases there was no or a weak relation;
- In 10,82% of all cases there was a weak, moderate or strong relation;
- There seems to be no relationship between the degree of participation and the amount of active developers.

4 DISCUSSION AND FUTURE WORK

The use of categories to group influences, resulted in the reduction of noise in the analysis. But even with this reduction, it remains uncertain if there was any noise. The categories with the highest amount of influences have less noise than the categories with few influences. Also, the influences limited to the Debian ecosystem itself were considered, because these influences are more likely to be relevant than external factors. To get even more accurate results, other factors should also be taken into account. Debian was exclusively investigated. The results can be generalized to decentralized OSS ecosystems, however, to further verify the results in this paper an identical study needs to be performed. Other decentralized OSS projects can also use the results of this research, but because the results are not verified with a second case study, the reliability of this research remains uncertain. Furthermore, to find out whether sub groups (e.g., dormant, sporadic or hyperactive) exist within the ecosystem that respond differently to internal and external ecosystem influences, types of commit frequency and amounts need to be analysed. By doing so, a more detailed picture can be obtained for supporting software ecosystem orchestration. Another study can be done on the visitor statistics of the Debian website in relation to events that happened, such as study is particularly interesting in the Debian project, because there is a clear website and purpose separation due to the use of subdomains (e.g., vote.debian.org, qa.debian.org and bugs.debian.org).

5 CONCLUSIONS

Developer participation in OSS development communities is important. The continuation of the project is at risk without a constant flow of new commits. In order to maintain developer participation it is important to know what influences affect developer participation. All the data of the commits for Debian in the year 2000 to 2011 were collected. In total, 1.000.000 commits were collected and analysed. 117 influences were found. These influences were spread out over 15 categories of influences that could potentially influence the developer participation of the Debian community. Based on the obtained data, a statistical analysis was performed to find relationships between the commits and the categories of influences.

In only 10,82% of all measurement points, there was a weak, moderate or strong relation. It can be concluded that although some influences have a clear change in developer participation, the effect is not large. The biggest influences proved to be hosted or joined events (i.e., CeBIT, Debian day), new or frozen releases, incidents, dependency issues and the introduction of new developer services. Debian and other OSS projects can use this research to pay more attention to the events that cause a change in developer participation. By doing this, they can obtain a more streamlined orchestration of their developer ecosystem.

6 REFERENCES

[1] Ammann, J., González-Barahona, JM. and de las Heras Quirós, P. 2001. Counting Potatoes: the Size of Debian 2.2. The European Online Magazine for the IT Professional. 11, 2, 60-66. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.48.8320&rep=rep1&type=pdf.

[2] Jansen, S., Finkelstein, A. and Brinkkemper, S. 2009. A sense of community: A research agenda for software ecosystems. Software Engineering - Companion Volume, 2009. ICSE-Companion 2009, pp.187-190.

[3] Messerschmitt, D.G. and Szyperski, C. 2003. Software Ecosystem: Understanding an Indispensable Technology and Industry. The MIT Press, Massachusetts, Cambridge.

[4] Ye, Y. and Kishida, K. 2003. Toward an Understanding of the Motivation of Open Source Software Developers. 25th International Conference on Software Engineering (ICSE'03), Portland, Oregon.

[5] Debian Project - debian.org Developers LDAP Search, http://db.debian.org/search.cgi.

[6] Developers per country, http://www.perrier.eu.org/weblog/2010/08/07#devel-countries-2010.

[7] Hars, S. and Ou, S. 2002. Working for Free? Motivations for Participating in Open-Source Projects. International Journal of Electronic Commerce. 6, 3 (2002), 25-39.

[8] Roberts, J., Hann, I. and Slaughter, S. 2006. Understanding the Motivations, Participation and Performance of Open Source Software Developers: A Longitudinal Study of the Apache Projects. Marshall School of Business Working Paper No. IOM 01-06; Management Science, Forthcoming. Available at SSRN: http://ssrn.com/abstract=918518.

[9] Goeminne, M. and Mens, T. 2010. A framework for Analysing and Visualising Open Source Software Ecosystems. In: Proceedings of the Joint ERCIM Workshop on Software Evolution, New York, USA.

[10] Rumsey, D.J 2011. Statistics for dummies second edition, Indianapolis, Indiana: Wiley Publishing, 279-294.