Correlation between bug notifications, messages and participants in Debian's bug tracking system*

Miguel Pérez Francisco and Pablo Boronat Pérez
Universitat Jaume I
Campus Riu Sec,
12071 Castellón, SPAIN
{mperez,boronat}@uji.es

Gregorio Robles
Libre Software Engeneering Lab.
Universidad Rey Juan Carlos
Móstoles, SPAIN
grex@gsyc.escet.urjc.es

Abstract

Bugs are an essential part of software projects because they lead its evolution. Without bug notifications developers cannot know if their software is accomplishing its tasks properly. However, few analytical studies have been made about this aspect of projects. We have developed a tool to extract and to store information from Debian's BTS (Bug Tracking System) in a relational database. In this paper we show that there is a strong dependence between three variables which can be used to analyze the activity of a project through its bugs: bug notifications, communications between users and developers and people involved.

1 Introduction

Free software[1] is becoming a serious alternative to *traditional* privative software. Developer communities formed around free software projects have demonstrated that excellent results can be achieved in spite of being geographically dispersed. These communities are usually formed by volunteers, who organize using the Internet[6].The Research community is now studying the free software movement [5].

One interesting characteristic of these projects is their internal data is in general publicly available. The most relevant information is the source code, which allows its study, modification and improvement, but usually, there are other profitable data, like access to the control versions system, mailing lists, bug tracking systems, forums, *changelogs* or *todo lists*. In this work, we will focus on the data from bug tracking systems, an aspect which has not been well studied.

Previous research on bug tracking system has been devoted mainly to integrate it with facts from the version control [4, 1].

The typical data stored in bug tracking systems for each bug are following: the person who notified the bug, the notification time, the severity, the related platform, the developer assigned to solve it, the time the bug was closed (i.e. fixed), the state (solved, discarded, still open, ...) and so on.

There are many applications for bug tracking, being *bugzilla* one of the most popular. Other projects have their own bug tracking system, as it is the case for the Debian project which uses a tool called BTS (*Bug Tracking System*).

The purpose of this paper is to check the correlation between the most significant variables to express the activity in bug tracking systems (reported bugs, messages and involved people) using *btsextract*, a tool to retrieve information from the BTSof the Debian project,

2 The `btsextract` tool

Debian is a free operating system using the Linux kernel with a lot of free applications, most of them from the GNU project. The Debian project started in 1993 with a team of about ten programmers and some packages[2] for the i386 architecture. Nowadays in the project, more than a thousand developers are involved, more than 15,000 packages are provided and it can be installed in more than ten computer platforms [7].

Debian's bug tracking system (BTS)is used since 1994 because the project growth required a rigorous track of bugs [2]. The last version, still in use, dates from 1999. BTS stores all details of bugs reported by users. All interactions

*This work have been partially funded by project P1.1A2004-13 from *UJI* and *Fundació Caixa-Castellá Bancaixa* agreement and by the Europan Comission, through projects FLOSSMetrics, FP6-IST-5-033982, and Qualipso, FP6-IST-034763

[1] By *free software* we refer to that in which users enjoy freedom to execute, to copy, to distribute and to study or to modify it. By *privative software* we refer to the software not fulfilling any of previous conditions.

[2] A *package* is a compiled program ready to be installed in a system.

0-7695-2886-4/07 $20.00 © 2007 IEEE
DOI 10.1109/ESEM.2007.65

IEEE COMPUTER SOCIETY

with BTS are made through emails (even if specialized tools as `reportbug` or `bug-buddy` are used) , which are also stored and classified in the BTS.

Three possible ways can be used to access bug data reports in Debian's BTS: by email, via web or downloading the whole database using the `rsync` utility. `btsextract`[3] allows to perform more complex studies than the web access provided by Debian to its BTS given that the data are stored in a relational database.

`btsextract` works in two phases. First, the data stored in the Debian BTS is downloaded with `rsync`. Then, the data is processed and stored in a relational database. Figure 1 depicts the conceptual schema of the database used by `btsextract`. The same schema is used in another bug extracting tool, called `bugextract` [3], aimed to work with `Bugzilla`. This will make compatible the analysis of projects using different bug tracking systems.

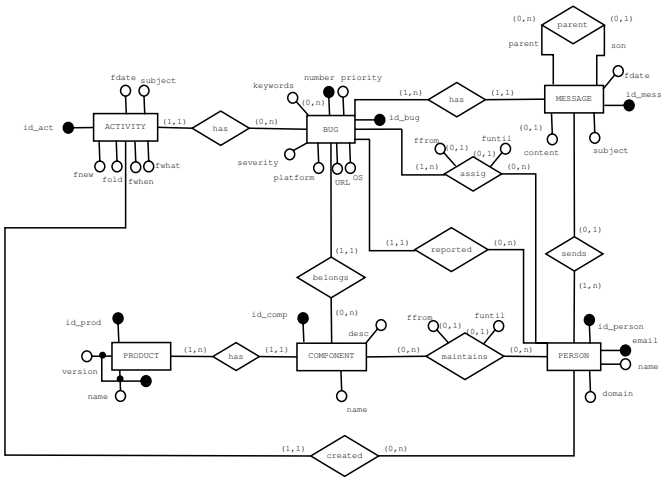

Figure 1. Conceptual schema of the database used by `btsextract`.

For each *bug* several characteristics (number, priority, platform, ...) are stored as it can be seen in Figure 1. The respective Debian package is also stored[4].

In `btsextract` all messages of a bug report are stored, including information about the submitter. Another interesting data is the package's maintainer when the bug was notified. This information can provide an approximative view about package maintainers over time.

[3] `btsextract` has been developed and released by the SóLiDo group at the Jaume I University under the GPL license. The last version can be downloaded through `CVS` from `http://forja.act.uji.es/`.

[4] In this schema, Debian packages are called "products" in order to maintain the compatibility with `bugextract`.

3 Results

Several variables can be used to characterize the behavior of software projects attending to the data obtained from bug tracking systems. These variables should be combined in the future with other kind of data, such as the time between releases, popularity factors, or the number of code lines.

The most obvious variables to reflect temporal activity of bug tracking systems are the amount of bug notifications, the number of messages exchanged, and the amount of people involved. In this paper we show the correlation between these three variables in order to know if they can be used indifferently in subsequent studies.

Table 1 shows that a linear correlation among bug notifications, messages and participants in Debian's BTS exists. The correlation has been performed for the whole system and for some packages (or groups of packages) in order to check if there are any differences. Data have been grouped by years (from 1996 to 2006).

We have grouped the packages that conform the `gnome`, `kde` and `xfree86+xorg` projects because they are probably the most used packages in GNU/Linux desktop systems. The other packages have been chosen among the most popular ones according to the *Debian Popularity Contest*.

	Total	gnome	kde	xfree86-xorg	base-passwd
AB	0.98	0.95	0.95	0.91	0.83
AC	0.99	0.96	0.95	0.99	0.98
BC	0.99	0.98	0.99	0.91	0.86
	dpkg	**debianutils**	**apt**	**bash**	**gzip**
AB	0.84	0.96	0.97	0.85	0.87
AC	0.91	0.99	0.99	0.97	0.97
BC	0.93	0.98	0.99	0.93	0.94

Table 1. Linear correlation between bug notifications (A), messages (B) and participants (C) in Debian's BTS. The Data have been summarized by years, from 1996 until 2006.

As it can be seen in Table 1, the linear correlation among bug notifications, messages exchanged and participants is very high. The minimum value, 0.83, is for messages and participants for the `base-passwd` package, but this is still high.

The three variables measured have all different subtleties. For instance, participants could be classified as users and developers, bug reports have several degrees of severity (from critical bugs to new desired options) and for messages we could differentiate, for instance, between those sent by users and those sent by developers. In this brief version we present the raw data. In future work we plan to perform more correlation tests to check if partial aspects of these

variables affect their correlations.

Figures 2 and 3 show the temporal evolution of the whole activity in BTS with respect to the three variables we are dealing with. They show graphically the strong correlation detected between the variables.

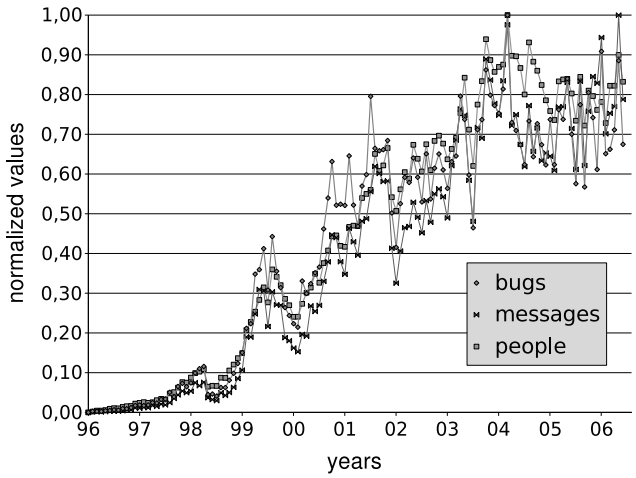

Figure 2. Monthly evolution of bug notifications, messages and participants (normalized data).

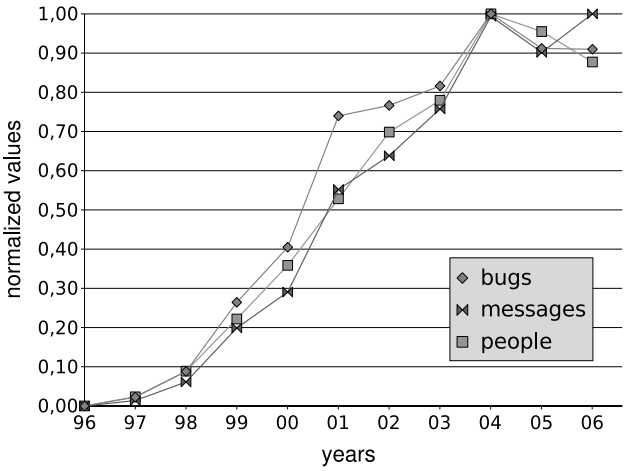

Figure 3. Annual evolution of bug notifications, messages and participants (normalized data).

4 Conclusions and future work

We have started to analyze the behavior of bug tracking systems of free software projects in order to characterize and better understand these complex emergent social networks. The results provided in this paper correspond to data from Debian's BTS obtained with `btsextract` and stored in a local relational database.

From the data that can be extracted from bug tracking systems, the most straight variables that can be used to study the temporal evolution are the number of bug notifications, the amount of messages exchanged and the amount of people involved. In this paper we have analyzed the correlation between these three variables. We have observed a high dependence among them, so that any of the variables can be used with limited loose of generality in further studies.

We have briefly shown the result of performing various temporal analysis and we have contrasted the results obtained for some *Debian packages*.

However, some aspects have still to be verified. For instance, other free software projects not related with Debian have to be analyzed and also, as previously mentioned, it have to be contrasted if the particularities of the variables have an impact on their relation.

References

[1] G. Antoniol, M. D. Penta, H. Gall, and M. Pinzger. Towards the integration of versioning systems, bug reports and source code meta-models. *Electronic Notes in Theoretical Computer Science*, 127(3):87–99, April 2005.

[2] D. Benham. debbugs: Debian bug tracking system. in `http://www.benham.net/debbugs/` (visited on 4/7/2006).

[3] P. L. Calvo-Barbero, P. Boronat-Pérez, and M. Pérez-Francisco. Desarrollo de un sistema de análisis de sistemas de seguimiento de errores. Technical report, Universitat Jaume I, 2006. `http://forja.act.uji.es/`.

[4] M. Fischer, M. Pinzger, and H. Gall. Populating a release history database from version control and bug tracking systems. In *Proceedings of the International Conference on Software Maintenance*, pages 23–32, Amsterdam, The Netherlands, September 2003.

[5] L. López, G. Robles, J. M. González-Barahona, and I. Herraiz. Applying social network analysis techniques to community-driven libre software projects. *International Journal of Information Technology and Web Engineering*, 2006. In press.

[6] E. S. Raymond. *The Cathedral & the Bazaar*. O'Reilly, January 2001.

[7] G. Robles, J. M. Gonzalez-Barahona, M. Michlmayr, and J. J. Amor. Mining large software compilations over time: Another perspective of software evolution. In *Proceedings of the Third International Workshop on Mining Software Repositories*, pages 3–9, Shanghai, China, May 2006.