Impact of the Creation of the Mozilla Foundation in the Activity of Developers*

Jesus M. Gonzalez-Barahona, Gregorio Robles, Israel Herraiz
GSyC/LibreSoft, Universidad Rey Juan Carlos (Madrid, Spain)
{jgb, grex, herraiz}@gsyc.escet.urjc.es

Abstract

During 2003, the Mozilla project transitioned from company-promoted (sponsored by AOL) to community-promoted (sponsored by the Mozilla Foundation). What happened to the group of developers during this transition? There was any significant impact on its activity or composition? To answer these questions, we have performed an analysis of the CVS repository of Mozilla, using the CVSAnalY tool, finding little on activity, but dramatic changes in the the composition of the development team.

Keywords: *Firefox, Mozilla, source code management system, software development*

1. Introduction

The Mozilla project started in 1998 to create the next-generation Internet suite for Netscape (the company producing the Netscape Navigator) as libre (free, open source) software. Netscape was later acquired by AOL, which continued the support to the Mozilla project until 2003. In July that year, the Mozilla Foundation was registered (with some initial funding from AOL), taking over the responsibility for the support of the project and its development community.

This transition from a company-backed to a more community-backed project (also including corporate support, but not driven by a single company) shows aspects worth researching. Among them, we have focused on its impact on the development team. In particular:

- Was there any significant impact on the activity of the development process?
- Was there any significant impact on the composition of the development team?

Both questions have been raised often in informal discussions, especially when considering the implications of the involvement of companies in libre software development, and what happens when that involvement comes to an end.

We have used a quantitative approach, based on the study of the traces of activity found in the the CVS repository of the project:

- Information about all change records in the Mozilla CVS repository was retrieved using the CVSAnalY tool [3]. This produces (among others) a table in a database, with one record per change. The record includes information about the commiter, the time of the change and the file changed.
- Several queries are performed on this table, using the Generations plugin for CVSAnalY. These queries produce matrices with the number of commits per core group per period (see below for details on generations and periods).
- Matrices are plotted using GNUplot, which generates a 3D graph which can be rotated, zoomed, etc.

The resulting 3D graphs will help to answer the proposed questions.

2. Generations analysis

At any given moment in the life of a libre software project, the level of activity of each of the members of the development team can be very different. Although this probably happens also in traditional software development, the voluntary component found in many libre software developers mean that they can quickly vary

*This work has been funded in part by the European Commission, under the FLOSSMETRICS (FP6-IST-5-033547) and QUALOSS (FP6-IST-5-033547) projects.

Fourth International Workshop on Mining Software Repositories (MSR'07)
0-7695-2950-X/07 $20.00 © 2007 IEEE

their activity over time, depending on their interest and circumstances. To better understand how these levels evolve, and how those most active during a certain period behave before and after it, we devised the generations analysis [2].

For this analysis, the life of the project is split into periods of equal size. For each of these periods, the most active developers (the fraction performing most commits during that period) are identified, and considered as the core group for that period. For instance, considering a fraction of 0.1 and a period of three months, we identify, for each quarter, the 10% of active developers (during that quarter) who performed more commits. Then, we study the activity of those developers during the whole life of the project: the history of that core group, or "generation".

The resulting information is a squared matrix, with the value of a cell, $M_{i,j}$ being the number of commits performed by the core group i (the most active developers during period i) during period j. This information is presented in a 3D graph, in which X, Y values correspond to i, j, and Z value corresponds to $M_{i,j}$. Values are also colored, with black and cold colors (blue) for small values, while hot colors (red) are used for high values (see figure 1 for an example). The values can also be normalized (as a fraction over the overall activity of the period), producing the normalized matrices.

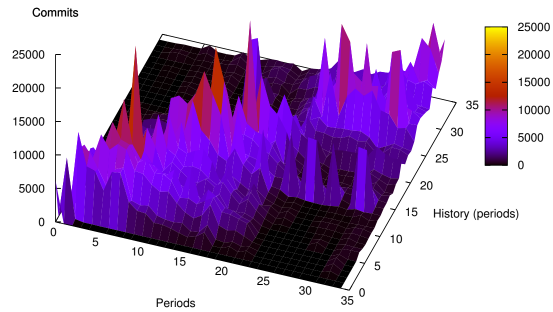

Figure 1. Core teams over time: fraction of 10%

When considered globally, these graphs represent the activity of the different core groups of the project, as will be shown in the case of Mozilla.

For our study, we have used periods of three months, which gives 34 periods, or 8.5 years, from the 3rd quarter of 1998 (quarter 0) to the 4th quarter of 2006 (quarter 33). July 2003, when the Mozilla Foundation was registered, corresponds to quarter 20.

3. Global activity

If we consider the activity matrix for a fraction of 1.0 (that is, 100% of developers), Z values for i, i cells match the total number of commits for period i. We can view the corresponding 3D graph from a perspective that shows those values (projecting on the X,Z plane). This is shown for Mozilla in figure 2, which therefore presents the total number of commits per quarter.

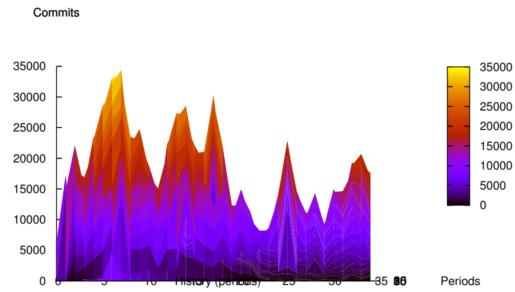

Figure 2. Total activity over time: frontal view of the 3D graph for a fraction of 100%

This single graph answers our first question: the mean level of activity is lower after July 2003 (period 20 in the graph), although it seems to follow a descending trend, from the peak of more than 30,000 commits per quarter around early 2001. However, the effect of the transition is clear, with a minimum of less than 10,000 commits in several quarters around early 2004 (22 to 24).

4. Changes in the most active team

The results about total activity over time are interesting, but may be obtained by simpler means: for instance, calculating the total number of commits per quarter, which is straightforward from the table produced by CVSAnalY.

Answering the second question is more difficult, and here is where the generations analysis shines. To begin with, consider figure 1, which shows the activity for the 10% most active developers for each quarter in the life of Mozilla[1]. At first sight, it is obvious that something dramatic happened in late 2003: activity of core teams for quarters before 22 becomes almost zero for quarters after it. Correspondingly, core teams for latter quarters performed almost no activity before 2003. In fact, this behavior can also be observed with other fractions (such as 20% or 25% most active developers), which shows the stability of the observation.

The impact is even more clear if we normalize the data (considering fractions of total number of commits instead of the absolute number of commits), as is shown in figure 3. The leadership (in terms of activity) switched completely around 2003: two different "mountain chains" are apparent, corresponding to the two leaderships, with a small link around quarter 20.

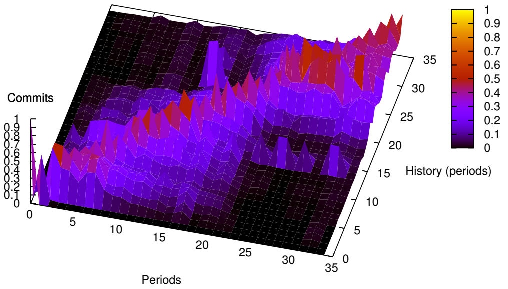

Figure 3. Core teams over time: fraction of 10%, normalized

The heat map in figure 4 (obtained by projecting over the X,Y plane the previous 3D figure) shows the same history. The red diagonal corresponds to the i, i cells (top activity for each core group). The top left black area corresponds with the periods of inactivity of the "Foundation" core groups. The bottom right black area corresponds to the inactivity of the first core groups. The change in leadership is clear.

Some other facts are apparent in these graphs. For instance, the slope for the second mountain chain is steeper, corresponding to a quicker evolution to presence

[1] For each quarter, we produce a list of developers commiting to CVS during that period (active developers), ordered by number of commits. The 10% most active developers are the 10% top developers in that list.

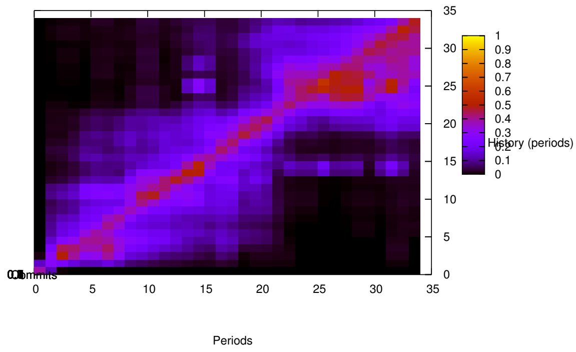

Figure 4. Core teams over time: fraction of 10%, normalized, view from above

in core teams in the Foundation era. However, that chain seems to be also thicker, which implies that developers in core teams are active in the project longer. This could mean less turnover during the Foundation era.

5. Conclusions

After these results, and assuming that each CVS account corresponds to a different, unique developer (see discussion below), we have answered the questions raised in the Introduction:

- The transition from the AOL era to the Mozilla Foundation era caused a clear drop in activity during several quarters. The activity after the transition seems to be lower than before it.
- The group of most active developers changed dramatically after the transition. Most of the developers in the teams showing most activity with the Mozilla Foundation have little or no activity during the AOL period, and vice versa.

In fact, the generations analysis has shown a dramatic disruption in the group of most active developers, happening while the Foundation started to take over the project. A more detailed analysis could explain whether this is due to new policies (e.g., different commit rules to the CVS repository), or a real change in persons involved (e.g., AOL employees leaving the project and being substituted by others).

Whatever the case, the implications of this disruption for the project have to be many. For instance, it is clear

Fourth International Workshop on Mining Software Repositories (MSR'07)
0-7695-2950-X/07 $20.00 © 2007 IEEE

than those with most expertise in the project were not available (in general) after 2003. It would be interesting to study the correlations of this fact with possible architectural changes in the software, if any. It would also be interesting to match this switch in the people most active in the project with the pace of stable releases, or to other parameters related to the performance of the project.

With respect to the methodology used, the main drawback that can be noted is its dependence on the assumption that each CVS identifier corresponds to a different, unique developer. In some cases, several identifiers are used by the same developer due to specific policies of a project, or a single CVS identifier is used to commit contributions of many developers [1]. In the case presented in this paper, a manual review of a sample of CVS identifiers has been performed, in order to check that the assumption is valid. Fortunately, this seems to be valid for most projects, but unfortunately, in general it is not possible to assert the validity of the results without a careful check of the CVS identifiers, and the policies of the project.

The methodology also depends on on a single source of data about the activity of the project. A similar analysis should be performed on other data sources (bug reporting system or mailing lists, for instance) to validate or question it.

We expect that this kind of analysis can be useful in at least two scenarios:

- For those with a deep knowledge about the project, it can be a source of complementary and quantitative information, highlighting and parameterizing situations. In our case, probably most Mozilla developers are aware of this switch in core teams, but maybe they were not aware of its depth.

- For those with an anecdotal knowledge about the project, the analysis can offer some data to produce a first idea about it, backed by facts, which may lead them to further investigation.

In both cases, 3D images can be a rich source of visual information (offering different aspects and focus when rotating or zooming), while numbers in matrices can complement that insight with quantitative data.

As a side note, a similar analysis defining core teams as those performing a certain fraction of the commits (instead of being the top fraction of commiters) shows similar results. However, for some specific information, this alternative approach may be better.

For instance, consider figure 5, which is the heat map corresponding to the core groups performing at least a

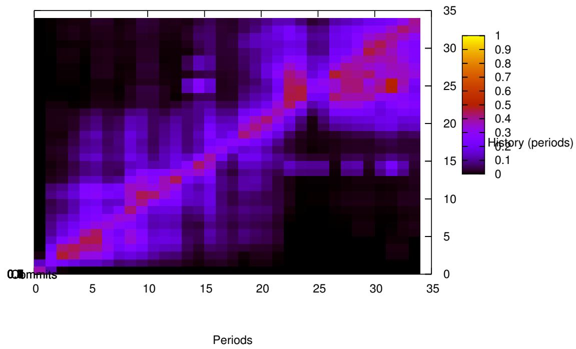

Figure 5. Core teams over time: fraction of 50% (commits), normalized, view from above

50% of all the commits during their periods. Here we can look at those developing more than half the project. The evolution of their activity over time is similar to that described before, but it may be interesting to notice that mountain chains are narrower in this case, and that activity patterns after 2003 are more constant than before. A complementary analysis of the distribution of activity among developers during each period could probably help to explain this behavior, but it could be related to a larger concentration of activity in the pre-2003 era.

References

[1] D. German. A study of the contributors of postgresql. In *Proceedings of the International Workshop on Mining Software Repositories*, pages 163–164, Shanghai, China, May 2006.

[2] G. Robles and J. M. Gonzalez-Barahona. Contributor turnover in libre software projects. In *Proceedings of the Second International Conference on Open Source Systems (OSS2006)*, volume 203/2006 of *IFIP International Federation for Information Processing*, pages 273–286, Como, Italy, June 2006. Springer.

[3] G. Robles, S. Koch, and J. M. González-Barahona. Remote analysis and measurement of libre software systems by means of the CVSAnalY tool. In *Proceedings of the 2nd ICSE Workshop on Remote Analysis and Measurement of Software Systems (RAMSS)*, pages 51–56, Edinburgh, Scotland, UK, 2004.

Fourth International Workshop on Mining Software Repositories (MSR'07)
0-7695-2950-X/07 $20.00 © 2007 IEEE