A Study of the Contributors of PostgreSQL

Daniel M. German
Software Engineering Group, Dept. of Computer Science
University of Victoria
dmg@uvic.ca

ABSTRACT

This report describes some characteristics of the development team of PostgreSQL that were uncovered by analyzing the history of its software artifacts as recorded by the project's CVS repository.

Categories and Subject Descriptors

D.2.9 [**Software Engineering**]: Life Cycle, Programming Teams

General Terms

Management

Keywords

Software evolution, mining software repositories.

1. QUESTIONS ADDRESSED

Our goal was to answer the following questions:

1. Who are the contributors to PostgreSQL and what can we know about the number of their contributions?
2. Has the team's composition changed over the years?
3. Can we identify any patches that are submitted by persons without CVS accounts?
4. Do they keep strong territoriality over the code base? In other words, are most files modified by only one developer?
5. Do contributors have different roles? For instance, can identify people who program, create tests cases, document, etc?

2. INPUT DATA AND APPROACH

We used as the main source for our analysis the CVS repository of the project. We proceeded to mine it twice. The first time was Sept 9, 2004. During this stage we proceeded to materialize every revision of every source code file (i.e. we recreated every version of every source code file ever submitted to the repository). The second time was Feb 21, 2005; this time we only retrieved the metadata of the changes to the system. In both cases the first recorded change was made on July 9, 1996. One important point to highlight is that development of PostgreSQL started long before they started using CVS, and therefore, we only have a fraction of the total history of the project. For instance, Release 1.0 was published in 1995, and some copyright notices in some files date back to 1983.

For the mining of the repository we used the framework provided by softChange [2] (softChange uses PostgreSQL as its storage backend). We proceeded to create some derived information:

- We reconstructed atomic commits (in the rest of this paper we will refer to them as Modification Records –MRs)
- We reconstructed every version of every source code file submitted to CVS from July 9, 1996 to Sept. 9,2004.
- We created various statistics for each version of a file, and every MR, such as LOCSs, number of functions added and removed in each revision/MR, whether the revision/MR included only changes to the source code, etc.
- We have found that larger MRs in PostgreSQL tend to be changes in comments or code reorganizations, and if they are considered in any analysis they can add a significant amount of noise (for instance, in PostgreSQL the largest commits are reindentation of the source code –a task performed on a regular basis–or the update of the copyright's year) [1]. For that reason we have selected a subset of MRs (which we call codeMRs). codeMRs satisfy the following conditions: a) they are committed to the main branch of development; b) they contain at least one source file; and c) they contain at most 25 files. We believe that codeMRs are more representative of programming effort compared to MRs, and, in general, using codeMRs instead of MRs improves the quality of any analysis.

3. ANSWERING THE QUESTIONS

3.1 Who are the contributors?

We identified 28 different contributors to PostgreSQL who have a CVS account. Only 4 of them have contributed more than 5 percent of MRs. The proportion of MRs per contributor is depicted in figure 1. Like many other open source projects, most of the commits are done by a handful of individuals.

3.2 Has the team composition changed over the years?

Figure 2 shows, for any given year, the proportion of contributions of MRs for the top 10 contributors. Some observations can be made: the majority of contributions are performed, in any given year, by two persons (which we will call the core team); and one of the early members of the core team (*vadim*) was replaced by (*tgl*) between 1997 and 1998. Nonetheless, the team's composition has been very stable over the years.

3.3 Can we identify any patches that are submitted by persons without CVS accounts?

One problem faced with the analysis of the evolution of a software system based on CVS metadata is the difficulty of identifying contributions by those without a CVS account (these contributions are commonly known as patches). We reviewed the logs of each of

MSR'06, May 22–23, 2006, Shanghai, China.
ACM 1-59593-085-X/06/0005.

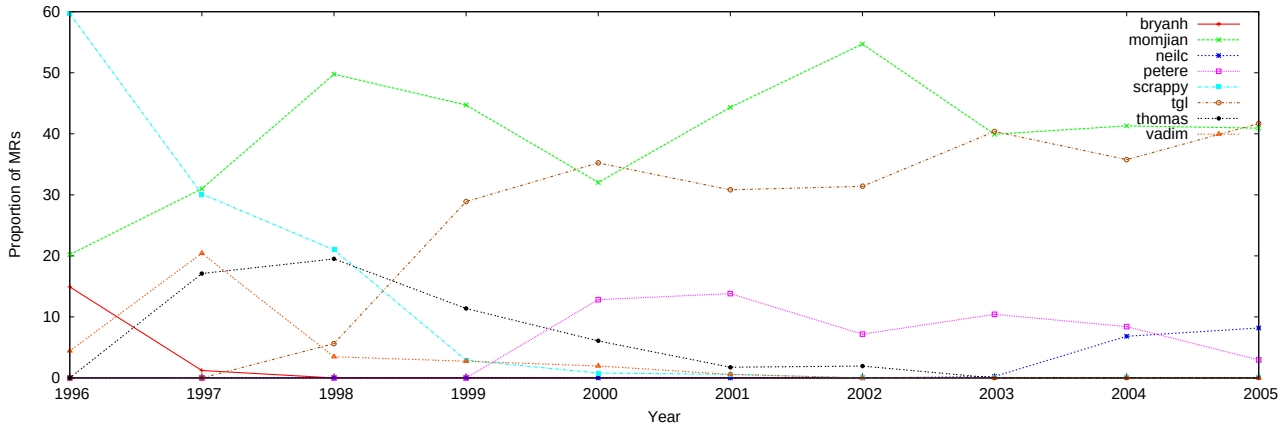

Figure 2: Proportion of contributors of MRs by year

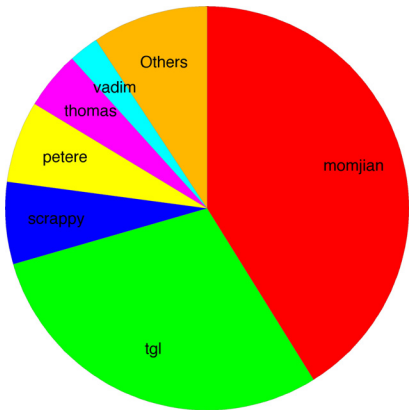

Figure 1: Proportion of contributors of MRs

the 364 codeMRs *momjian* performed during 2005 trying to find any indication of these patches. We were able to identify 110 MRs (roughly 1/3 of the total) to be patches submitted by 46 different individuals. We also found ample evidence of bug reporting by a large number of individuals. It is important to note that the format in which these contributors are acknowledged is different from other projects (at least in the experience of the author): the words "patch" and "contributed" are rarely mentioned, and the email of the developer is not included either. We also inspected some commits by *tgl* to be patches, but they were significantly fewer; but *tgl* committed a large number of MRs where he acknowledges people who submitted bug reports, designs and other contributions.

3.4 Do they keep strong territoriality over the code base?

A change to a file does not necessarily mean somebody has expertise on that file. This observation is best exemplified when the source code of PostgreSQL is reindented (a process that is done on a regular basis, usually before a release) or, at the beginning of a new year, when the copyright statement at the top of each file is changed. The person who reindents the file might not have any idea of what the code being reindented does. For that reason we decided to study changes to files in codeMRs. Furthermore, territoriality might change over time, thus we concentrated in changes performed during 2005. We proceeded to compute, for each pair (contributor, directory):

$$T_c^d = \frac{revisions\ by\ contributor\ c\ in\ directory\ d}{total\ revisions\ to\ directory\ d}$$

During 2005, 173 directories were modified (by a total of 10 people). We found that 123 of these directories had one developer responsible for at least 70% of the changes ($T_c^d \geq 0.7$). In 81 of these directories (primarily in the database engine) the responsible was *tgl* ($T_{tgl}^d \geq 0.7$) . The next was *momjian* with 18 directories; it should be taken into account that *momjian* is responsible for committing patches submitted by contributors without a CVS account (as previously discussed) and therefore he might not have created those modifications (but he probably reviewed them, nonetheless).

3.5 Do contributors have different roles?

We have already discussed that *momjian* is responsible for applying patches, and *tgl* is responsible for most of the source code. Other observations are: *petere* has been responsible for committing most of the internationalization files (.po), while some CVS account holders have taken care of translating PostgreSQL into languages they know (for example *alvherre*, who has committed Spanish translations, or *dennis*, Swedish).

4. CONCLUSIONS

At first we were surprised by how small and stable over the years the core team of PostgreSQL has been. Its CVS repository shows that, in the last years, only two persons have been responsible for most of the source code. We needed to inspect the history of the project in more detail, and were surprised to learn that there is a very large number of contributors who send source code *patches* to the project. This is an important lesson for anybody trying to inspect the history of projects, particularly when the analysis is done automatically. In the end we learned that PostgreSQL has a large and vibrant community who contributes bug reports and patches.

5. REFERENCES

[1] D. M. German. An empirical study of fine-grained software modifications. In *20th IEEE International Conference on Software Maintenance (ICSM'04)*, pages 316–325, Sept 2004.

[2] D. M. German, A. Hindle, and N. Jordan. Visualizing the evolution of software using softChange. In *Proceedings SEKE 2004 The 16th Internation Conference on Software Engineering and Knowledge Engineering*, pages 336–341, June 2004.