Recovering Valuable Information Behaviour from OSS Contributors: An Exploratory Study

Mário André de F. Farias∗§, Paulo Ortins∗, Renato Novais†, Methanias Colaço Júnior‡, Manoel Mendonça∗¶

∗Federal University of Bahia †Federal Institute of Bahia ‡Federal University of Sergipe ¶Fraunhofer Project Center for Software and Systems Engineering, Bahia, Brazil
Email: mario.andre@ifs.edu.br, paulo.ortins@gmail.com, renato@ifba.edu.br, mjrse@hotmail.com, manoel.mendonca@ufba.br

Abstract

Context. Distributed software development is currently a modern practice in software industry. This is especially true in Open Source Software (OSS) development community. Understanding how developers’ practices are on those projects may guide communities to successfully manage their projects. Goal. We mined two repositories of the Apache Httpd project in order to gather information about its developers’ behavior. Methods. We developed an approach to cross data gathered from mail list and source code repository through mining techniques. The approach uses software visualization to analyze the mined data. We conducted an experimental evaluation of the approach to assess the behavioral patterns from OSS development community. Results. The collected data built a rich set of information. The results highlight Apache developers’ behavior patterns. We perceived that there is a correlation over time between emails and commits of the Apache Project developers. Conclusion. The use of data mining and software visualization to analyze crossing data sources can spot important properties of the development process.

I. INTRODUCTION

Software repositories have been used to discover useful knowledge about the development, maintenance and evolution of software. However, these data sources are not built in a structured and organized way. So, it is necessary a considerable effort to gather evidence from those repositories. To this end, researchers have been developing different approaches [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]. They use data mining, software visualization (SoftVis), text mining and mining software repository. Some of them analyze each repository separately, even that combining different techniques (e.g. metrics, SoftVis) is a promising approach [11].

According to Kagdi et al. [6], the mining software repositories activity combines different areas, such as: knowledge discovery [7] and information retrieving [8]. Combine approaches from different areas is a promise strategy. For example, enrich those techniques with information visualization may produce solutions that aim to reveal software properties that were initially hidden in these environments.

Based on this premise, this paper presents an exploratory study that uses data mining and SoftVis techniques to analyze OSS developers’ behavior. The study is particularly interested in analyze discussion mailing lists and source code repositories through the use of SoftVis. The main characteristic of the used approach is the integration between the repositories data. It aims to integrate and analyze data originated from Apache Httpd mailing list and source code data.

This paper is organized as follows. Section II presents related works. Section III describes our experimental evaluation. Section IV reports and discusses our findings. Finally, Section V concludes the paper with a discussion of future work.

II. RELATED WORK

This section discusses some related work concerned with identifying patterns in development community through mining repository software and software visualization. Heller et al. [5] proposed a strategy that mined a GitHub repository metadata and used visualization techniques to identify patterns in development community. The study focused on specific patterns, such as the effect of geographic distance on developer relationships, social connectivity and influence among cities, and variation in project specific contribution styles.

Some works have already considered email specific analysis to study OSS development process and behavior of people [4, 12]. In [4], the authors investigated the impact of computer-mediated interaction on person perception. In particular, they studied how important traits for socialization and collaboration may be detected from the text of an email communication. Pattison et al. Finally, [12] studied the relation between several software entities mentioned in emails and the number of times these entities were included in the changes made.

Two works are closest to the research presented here. First, Canfora et al. [1] mined explicitly documented cross-system bug fixings from versioning repository and data from two project mailing lists. It aimed to identify Cross-System-Bug Fixings activities between FreeBSD and OpenBSD, and to understand the social role of developers performing such activities by means of social network analysis. We based our cross-system mailing list in this work. Second, in [3], the authors performed an experimental evaluation of the approach. They conducted an experiment to assess the PRS of top developers at Apache server mailing lists.
III. EXPERIMENTAL EVALUATION

The experimental process follows the guidelines by [15]. Here, we present the experimental definition and the planning process. The next section presents the gathered evidence and the results.

The main goal of our study is to reveal interesting behavioral patterns in OSS contributions, such as the crossing analysis of mailing lists contribution, commits in the projects, and geographic location (geo-location) of contributions in OSS.

A. Planning

1) Context Selection: The experiment context was open source project repositories. These repositories have large amount of e-mails and commits. Commonly, the data is not ready to use. It is necessary to clean the data to avoid misleading understanding. For example, there is an arduous manual work of searching for valid emails and commits data. For that, we developed powerful computational procedures following [2][3]. On top of that, we did a detailed manual analysis of the committers’ profiles in order to gather geographic information. The approach study followed three steps: First, we extracted data from: a) Apache’s commits repository; b) Apache developer’s mailing list; and c) geographic information from geo-location services; Second, we cross the data collected in the previous step in order to associate the data to the developer that produced it; and finally, we built interactive visualizations that help users to discover relevant information. Next, we present each of these steps in detail.

Three modules were developed by us for this study in order to provide an able environment to integrate and analyze two different source repositories. The first is a module for Extraction, Transformation and Load (ETL) of emails. The second module is for mining source code repositories. Finally, the third is a visualization module. These visualizations are publicly available at (http://goo.gl/RSs4VR).

2) Research Questions: This work aims investigate OSS Developers’ behavior. To do this, we mined two software repositories in order to analyze the research questions addressing the distribution of email and commits over time in the project. Our research questions are described as follows:

i. How commits and emails are distributed over time among the Apache Project developers?

ii. How commits and emails are distributed over time by Apache Project top committers?

3) Participant and Artifact Selection: In this study, we used as object of analysis the Apache Httpd Project (http://httpd.apache.org/). Over its 17 years of development, the project received more than 60,000 commits, totalling more than two millions of lines of code written by more than 100 developers around the world. These developers use a mailing list to communicate with each other.

To answer our research questions, we extracted and analyzed the body of 100,479 email messages and 33,586 commits from the Apache repositories between 1995 and 2005. We selected the four developers who had the greatest number of commits. We refer to these developers as "Dev A", "Dev B", "Dev C" and "Dev D". We also grouped all the other developers, and refer to them as "Cluster", these developers represent the rest of population. We analyzed the same developers and same period used in the related study [3].

4) Preparation: The experimental study preparation took two months. One month and three weeks for understanding and develop the modules and one week for the preparation and execution of a pilot. The pilot study was carried using a small sample of emails and commits which was chosen at random. Thus, the pilot helped us to calibrate some specific characteristics of our modules and to find improvement point, such as performance of the crossing data and geographic information.

B. Experiment execution

We retrieved the geographic information (latitude/longitude and the time offset) for each committer aiming to know the origin of commits and emails. Apache Httpd project does not have this data for all developers. They provide this information only for the core committers (the ones who contributes more to the project).

In these cases, Apache Httpd project provides a page with complementary information about them. Unfortunately, core committers represent only 63 from the total developers (110). For the others, we needed to perform a manual task to retrieve their geographic information. They also have different time offsets. This brings out another issue, since it is necessary to consider the time zone when collecting the weekday and time for each commit. In this case, we needed to get each developer’s time zone and adjust the times for the Apache server time.

After retrieving geographic information, we found more 27 profiles, totalling 90 from the 110 available developers. We decided remove the commits from those developers which we could not find geographic information. So, we reduced the amount of analyzed commits from 33,586 to 31,611. Even considering all crossing data process, it was not possible to find a match between some emails and commits. In those cases, the data was ignored. This corresponded to 29,698 emails. So, we considered in our analysis 70,781 emails.

To establish a link between extracted data from email and commits we needed that at least one property was shared by both data sources. For example, either an ID or an email in both data sources should be equal. However, this was not possible, since, in Apache Http project, the data sources have different users’ profiles. This was another challenge in our approach. It was necessary to match different kinds of data, e.g. email address with nickname and name with nickname. In order to perform this task, we adapted the approach proposed by [1].

IV. RESULTS AND DATA ANALYSIS

The collected data built a rich set of information. Nonetheless, in general, data extracted from software repositories are too difficult to be analyzed in the same state that they were stored [9]. Thus, we decide to use visualizations to reorganize them in such manner that users can easily understand the
whole database. We discuss now the results of this study. To answer the research questions, we analyzed the data taking into consideration (i) the relation between emails and commits of the Apache Project developers (ii) the relation between emails and commits of the Apache Project top committers (iii) The beginning of Apache Project.

i. Relation between emails and commits of the Apache Project developers: Through the interaction with the period filter to generate heat maps over the time, we perceive that the heat zones (regions where contributions were made) used to appear first in the emails’ map and after in the commits’ map, it’s evidence that in this project developers first interact in the email list and after commit code to the repository.

We could confirm this behaviors in the Apache Httpd Web Site. According with the site, changes to the code are proposed and voted on the mailing list and only after they are approved, they are committed in the repository. On top of that, we could also identify that the regions that have more participation in the emails list are also the regions with more participation in the code repository (see Fig. 1). An exception is the Japanese developers’ behavior. In this case, there is a considerable amount of commits (bottom of Fig. 1) but a low participation in the discussion mailing list (top of Fig. 1). It may suggest an introverted behavior due to cultural factors.

![Heat maps showing amount of (a) emails and (b) commits](image)

Fig. 1: Heat maps showing amount of (a) emails and (b) commits

We also perceived that there is a correlation between emails and commits timestamps. Developers normally commit code and discuss in the list in the same time as well in the same weekday. Figure 2a shows the interactions over time between emails and commits of the Apache Project developers. If we analyze the evolution, we may confirm that commits and emails follow the same pattern distribution.

ii. Relation between emails and commits of the Apache Project top committers: For a more detailed analysis, our crossed data approach allows us to explore individually data from every developer in the project. Due the restriction of this paper’s size, we choose to analyze the relation between emails sent and commits made only by the top committers. The Fig.

![Interactions over time between emails and commits of developer population](image)

Fig. 2: Interactions over time between emails and commits of developer population

iii. The Apache Project beginning: Analyzing carefully the data evolution in Fig. 2 we could see that the discussion in the email list started a year before the first commits. This is an interesting behavior, since they had time to discuss before to start the implementation. However, it is common to observe OSS projects starting on the other way around. First, a project is created with few developers. They start to commit and create the first release. After that, users start to use it. Using the software, bug fix and request for new features will rise. So, users use a bug tracking system to report the issues. At the end, the developers start the discussions in the email list.

We decided to investigate why the Apache http evolved in this way. We looked the website and discovered that the Apache Httpd was a continuation of NCSA Httpd, which stopped to be developed when Rob McCool left NCSA in 1994. A group of webmasters then started to develop their own extensions and bug fixes, in 1995, they solved to join all
Fig. 3: Distribution of emails and commits of the Apache Project top committers. (a) Dev A, (b) Dev B, (c) Dev C, (d) Dev D.

this features and bug fixes in a unique distribution and then the Apache Group was created.

V. CONCLUSION AND FURTHER WORK

In this paper, we presented an useful and innovative approach that extract information from two important data sources in a software project. We mined and try to match emails list and source code repository data. This approach can be used to discover hidden behavioral patterns in unstructured databases. We also believe that our approach can be used by OSS leaders to increase developers’ contributions or to keep contributors in their projects. OSS managers can also use our approach to split tasks accordingly to each developers’ profile or to tracking team’s contributions over time considering weekdays and day periods.

We have evidences that discussion lists and repositories can be used to measure project activity or to predict each other. We now draw answers to our research questions stated in the section III. Regarding RQ1, we may confirm that commits and emails follow the same pattern distribution in the Apache evolution. Considering RQ2, we could see that this relation is very similar to rest of developers. Besides, we identified from our visualizations is possible to find some work pattern.

Our future work will address three key issues: (1) Improve our approach by extracting other relevant data from other OSS. This work is in process; and (2) Extend this study to mine data from Postgree, emails and commits, aiming to compare to findings performed by [2, 3]; and (3) Develop new interactive visualizations.

REFERENCES