The JIRA Repository Dataset: Understanding Social Aspects of Software Development

Marco Ortu[1], Giuseppe Destefanis[2], Alessandro Murgia[3], Michele Marchesi[1], Roberto Tonelli[1] and Bram Adams[4]
[1]DIEE, University of Cagliari
{marco.ortu,michele,roberto.tonelli}@diee.unica.it
[2]CRIM, Computer Research Institute of Montreal, Canada
giuseppe.destefanis@crim.ca
[3]University of Antwerp, Belgium
alessandro.murgia@uantwerpen.be
[4]École Polytechnique de Montréal, Canada
bram.adams@polymtl.ca

ABSTRACT

Issue tracking systems store valuable data for testing hypotheses concerning maintenance, building statistical prediction models and the social interactions of developers when interacting with peers. In particular, the Jira Issue Tracking System (ITS) is a proprietary tracking system that has gained a tremendous popularity in the last years and offers unique features like a project management system and the Jira agile kanban board. This paper presents a dataset extracted from the Jira ITS of four popular open source ecosystems (as well as the tools and infrastructure used for extraction), i.e., the Apache Software Foundation, Spring, JBoss and CodeHaus communities. Our dataset hosts more than 1K projects, containing more than 700K issue reports and more than 2 million issue comments. Using this data, we have been able to deeply study the communication process among developers, and how this aspect affects the development process. For example, we found that comments posted by developers contain not only technical information, but also valuable information about sentiments and emotions. With this repository we would like to encourage further studies in these directions.

Keywords

Mining software repository, Issue Report, Affective Analysis

1. INTRODUCTION

The issue tracking system (ITS) is a software repository that hosts all development tasks of a software organization, i.e., new features, bug fixes and other maintenance tasks. For each such task, the ITS provides a description, administrative data like the state of the issue (e.g., opened or fixed) and the priority, as well as a chronology of comments and attachments by developers to discuss the task at hand. Moreover, issues can link to each other, for example when some issue is a duplicate of another one or depends on another issue to be completed first.

ITS data represents a gold mine for empirical research [6, 7]. ITSes have been widely used for testing hypotheses concerning maintenance [8], building statistical prediction models [18, 2]. The issue comments, in which developers discuss issues by providing technical details entwined with opinions provide rich information about the "why" of certain design decisions or about the status of a project. By looking at these comments it is possible to study how developers interact, as well as how they feel about the project and their peers. Hence, ITSes are an important source of information to study the productivity of teams of developers [12] or of developers' affectiveness [9, 10, 11, 14, 16].

The Jira repository is one of the most common ITS technologies. Jira contains the standard ITS information mentioned above, as well as more project management-related information. For example, Jira has a special board, which is a virtual representation of the agile kanban boards widely used by developers [1]. Jira also has the possibility to track team progress, to manage backlog, and plan sprints.

Given the high-level project information provided by Jira, we mined the Jira repository of four open source communities: Apache[1], Spring[2], JBoss[3] and CodeHaus[4]. We selected these ecosystems, since they are well known by practitioners. The resulting data set not only allows to study traditional ITS topics like bug triage, bug tossing, and bug priority, but also less common questions like "how do agile developers collaborate remotely?" or "which developers team is more efficient?". We have used the data in several studies [10, 11, 12], and now provide it in SQL form for other researchers. Our dataset hosts more than 1K projects, 700K issue reports

[1] http://www.apache.org
[2] https://spring.io
[3] http://www.jboss.org
[4] http://www.codehaus.org

and 2 million of comments.

The rest of the paper is structured as follows. First we describe how the dataset is built (Section 2) and organized (Section 3). Then we report the research opportunities based on its adoption (Section 4) and finally the conclusions (Section 5).

2. DATASET EXTRACTION

The main blocks of the architecture used to extract the dataset are presented in Figure 1.

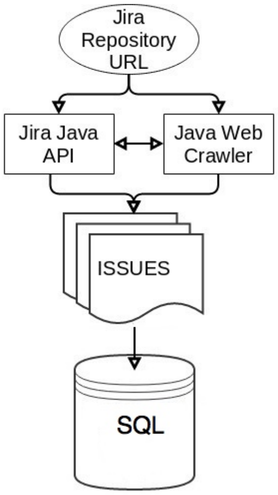

Figure 1: Extraction architecture

Jira Java API. We used the Jira REST API[5] to extract most of the issue data. We then manually parsed the issue pages in order to collect the extra data that was not possible to extract with the REST API, such as issue's attachements.

Java Web Crawler. We used a web crawler for parsing the web pages of Jira. To create the crawler we used the open source framework Jsoup[6], as it is lightweight and allows to parse a web page searching for html elements containing the desired information. For each requested web page, the crawler extracts the html body then parses it to create the relative Document Object Model[7] (DOM) representation. The crawler can navigate the DOM searching for elements containing the desired information using CSS.

Issues. Jira issue reports are characterized by four main sections: Activity, Attachments, Details and Description.

The Activity section may contain information about commits. If such information is available, the crawler extracts the commit's data. Otherwise, it analyzes the section Attachments looking for patch-files containing commit information. Furthermore, the section Activity contains a comment tab where developers can post comments related to the report. Sections Details and Description contain the title and the description of an issue, and additional information such as reporter, assignee, and status.

3. DATASET DESCRIPTION

Issues in Jira are divided in categories such as bugs, improvements, feature requests or tasks. The presented dataset contains issues belonging to all three categories.

The persistence layer of our tool maps the object model into a relational database. The database schema is shown in Fig. 2. Our database contains the following tables:

ISSUES_REPORT. It stores the information extracted from the issue reports. Issues are associated with comments and attachments and history changes.

ISSUES_COMMENTS. It represents all the comments posted by users and developers in a Jira issue report. This table is associated with the *ISSUES_REPORT* table. An example of a comment extracted is the following:

```
Hey <dev_name_a>,
Would you be interested in contributing
a fix and a test case for this as well?
Thanks,
<dev_name_b>
```

ISSUE_BOT_COMMENT. It represents automatically generated comments from tools such as Jenkins or Jira itself.

ISSUES_FIXED_VERSION. It records the software version of fixed issues.

ISSUES_AFFECTED_VERSION. It records the software version affected by issues.

ISSUE_ATTACHMENT. It represents all files attached to an issue report.

ISSUE_CHANGELOG_ITEM. It represents all operations made on an issue such as editing, updating, status changing, etc.

To illustrate the schema, the following SQL query describes how to retrieve the number of stored comments (more than 1 million) from the Apache Software Foundation (ASF):

```
SELECT COUNT(c.id)
FROM jira_issue_comment c,
     jira_issue_report i
WHERE i.id = c.issue_report_id
AND i.repositoryName = 'ASF'
```

Our full dataset (comprising the Apache projects) hosts 3516 tasks, 16173 files and 25306 comments by 1375 authors. The statistics of dataset are shown in Table 1. The dataset is publicly available on the tera-PROMISE website[8] in SQL format.

[5] https://developer.atlassian.com/jiradev/api-reference/java-api-policy-for-jira

[6] http://jsoup.org

[7] Document Object Model, an object representing the HTML document

[8] http://openscience.us/repo/social-analysis/social-aspects.html

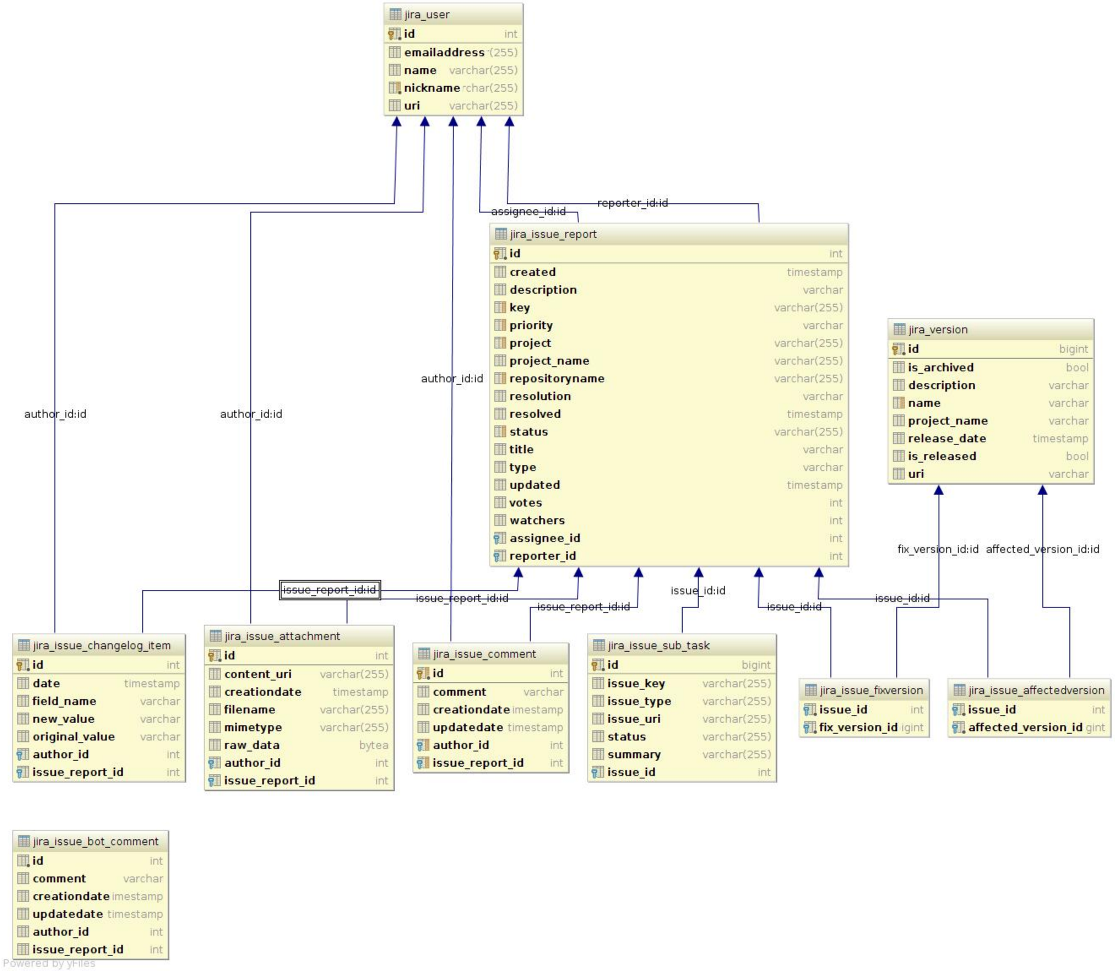

Figure 2: Database schema

Description	Values
# Issues	700K
# Comments	2M
# Users	100K
# Attachments	60K

Table 1: Dataset statistics

Although the dataset is limited to four open source ecosystems, we are confident about the data extracted is complete and consistent. When considering the link between SCM and ITS, the main limitation of this dataset is the fact that not each issue report might contain all links to the source configuration management (SCM). This creates a bias since not all issues committed in SCM are considered, neither all commit messages reported in SCM.

Another limitation considering the communication process of software development is that not all discussions about an issue are held in the issue tracking system, other communication means such as mailing list and social media are often used. Future extensions of our dataset will consider these media.

4. RESEARCH OPPORTUNITIES

The communication process plays a key role during software development. For this reason, the knowledge of a project should always be easily accessible to the development team. New developers joining the development team benefit of a good communication process, since it helps to improve productivity and learning-curve. Our dataset can be exploited to:

- study learning-curve, productivity and project's attractivity to new developers [17].
- build predictive models to (i) analyze social and technical debt in software development [13, 15] or (ii) estimate bug fixing time [18] and bug life cycle [2].
- test hypotheses concerning software maintenance [8].
- study the relationship among software metrics, patterns chosen during the development process and emotions [3, 4, 5].

In particular, mining this dataset we validated that issue comments contain indications of emotions, especially gratitude, sadness and anger [9]. Furthermore, we studied the relationship between emotions and developers' productivity in the form of issue fixing time [10]. The more developers express positive emotions in their comments, the shorter the issue fixing time is likely to be. On the contrary, negative emotions, are linked with longer issue fixing time. We studied also the relation between attractiveness of a project, and politeness expressed by developers working on it. We found that the more polite developers were, the more new developers wanted to be part of a project, and the more they were willing to continue working on it over time [11].

5. CONCLUSION

Data stored in ITS is fundamental for empirical research in software engineering since it can be used for verifying, refuting and challenging previous theory and results. Recently, analysis on ITS has focused on social aspects of software development. From this point of view, developer discussions stored as issue comments show how developers interact, as well as how they feel about the project and their peers. This paper presented a rich dataset hosting more than 1K projects, 700K issue reports and 2 million comments. We fetched the data by mining the Jira repository of four open source communities: Apache, Spring, JBoss and CodeHaus. We presented also the tools used for the mining activity and how information is organized in the dataset. We have used this dataset for studying the communication process among developers, and how this aspect affects the development process. We found that comments posted by developers contain not only technical information, but also valuable information about sentiments and emotions. Sharing this repository, we would like to encourage the community to perform replication as well as further studies in this direction.

6. REFERENCES

[1] D. J. Anderson. *Kanban.* Blue Hole Press, 2010.

[2] B. Caglayan, A. T. Misirli, A. Miranskyy, B. Turhan, and A. Bener. Factors characterizing reopened issues: A case study. In *Proceedings of the 8th International Conference on Predictive Models in Software Engineering*, PROMISE '12, pages 1–10, New York, NY, USA, 2012. ACM.

[3] G. Concas, G. Destefanis, M. Marchesi, M. Ortu, and R. Tonelli. Micro patterns in agile software. *Agile Processes in Software Engineering and Extreme Programming*, page 210.

[4] G. Destefanis, S. Counsell, G. Concas, and R. Tonelli. Software metrics in agile software: An empirical study. In *Agile Processes in Software Engineering and Extreme Programming*, pages 157–170. Springer, 2014.

[5] G. Destefanis, R. Tonelli, E. Tempero, G. Concas, and M. Marchesi. Micro pattern fault-proneness. In *Software Engineering and Advanced Applications (SEAA), 2012 38th EUROMICRO Conference on*, pages 302–306. IEEE, 2012.

[6] A. E. Hassan. The road ahead for mining software repositories. In *Frontiers of Software Maintenance, 2008. FoSM 2008.*, pages 48–57. IEEE, 2008.

[7] A. E. Hassan and T. Xie. Software intelligence: the future of mining software engineering data. In *Proceedings of the FSE/SDP workshop on Future of software engineering research*, pages 161–166. ACM, 2010.

[8] A. Murgia, G. Concas, R. Tonelli, M. Ortu, S. Demeyer, and M. Marchesi. On the influence of maintenance activity types on the issue resolution time. In *Proceedings of the 10th International Conference on Predictive Models in Software Engineering*, PROMISE '14, pages 12–21, New York, NY, USA, 2014. ACM.

[9] A. Murgia, P. Tourani, B. Adams, and M. Ortu. Do developers feel emotions? an exploratory analysis of emotions in software artifacts. In *Proceedings of the 11th Working Conference on Mining Software Repositories*, MSR 2014, pages 262–271, New York, NY, USA, 2014. ACM.

[10] M. Ortu, B. Adams, G. Destefanis, P. Tourani, M. Marchesi, and R. Tonelli. Are bullies more productive? empirical study of affectiveness vs. issue fixing time. In *Proceedings of the 12th Working Conference on Mining Software Repositories*, MSR 2015, 2015.

[11] M. Ortu, G. Destefanis, M. Kassab, S. Counsell, M. Marchesi, and R. Tonelli. Would you mind fixing this issue? In *Agile Processes, in Software Engineering, and Extreme Programming*, pages 129–140. Springer, 2015.

[12] M. Ortu, G. Destefanis, M. Kassab, and M. Marchesi. Measuring and understanding the effectiveness of jira developers communities. In *Proceedings of the 6th International Workshop on Emerging Trends in Software Metrics*, WETSoM 2015, 2015.

[13] A. Potdar and E. Shihab. An exploratory study on self-admitted technical debt. In *Software Maintenance and Evolution (ICSME), 2014 IEEE International Conference on*, pages 91–100. IEEE, 2014.

[14] P. C. Rigby and A. E. Hassan. What can oss mailing lists tell us? a preliminary psychometric text analysis of the apache developer mailing list. In *Proceedings of the Fourth International Workshop on Mining Software Repositories*, page 23. IEEE Computer Society, 2007.

[15] D. Tamburri, P. Kruchten, P. Lago, H. Van Vliet, et al. What is social debt in software engineering? In *Cooperative and Human Aspects of Software Engineering (CHASE), 2013 6th International Workshop on*, pages 93–96. IEEE, 2013.

[16] P. Tourani, Y. Jiang, and B. Adams. Monitoring sentiment in open source mailing lists – exploratory study on the apache ecosystem. In *Proceedings of the 2014 Conference of the Center for Advanced Studies on Collaborative Research (CASCON)*, Toronto, ON, Canada, November 2014.

[17] K. Yamashita, S. McIntosh, Y. Kamei, and N. Ubayashi. Magnet or sticky? an oss project-by-project typology. In *Proceedings of the 11th Working Conference on Mining Software Repositories*, pages 344–347. ACM, 2014.

[18] H. Zhang, L. Gong, and S. Versteeg. Predicting bug-fixing time: An empirical study of commercial software projects. In *Proceedings of the 2013 International Conference on Software Engineering*, ICSE '13, pages 1042–1051, Piscataway, NJ, USA, 2013. IEEE Press.