

# What Do Bug Repositories Tell Us About Software Development Workflow? An Exploratory Study

Rodrigo Souza and Christina Chavez  
*Software Engineering Labs*  
*Department of Computer Science - IM*  
*Federal University of Bahia (UFBA), Brazil*  
*{rodrigo,flach}@dcc.ufba.br*

**Abstract—Background.** Bug repositories have been increasingly studied with the objective of unveiling relevant knowledge about development practices and coordination tasks in software projects. Although there is plenty of publicly available repositories, researchers have raised issues regarding the accuracy of the data recorded in such repositories.

**Aims.** In this paper, we investigate some bug repositories in depth in order to understand the possible meanings behind the data (e.g., what does it mean to say that a bug has been “verified”?) for each particular software project analyzed.

**Method.** We have analyzed bugs from two large software projects, Eclipse and Netbeans. We have performed in-depth qualitative analysis within a small sample of bugs from both projects. Also, we have performed a broad textual analysis by applying latent Dirichlet allocation on all bug reports. Finally, we have used descriptive statistics in order to account for temporal (pre-release, post-release) and social (role of user: committer, fixer, reporter, end user) factors in the bug fixing workflow.

**Results.** We have found that statuses mean distinct things in different projects. Also, the kind of activity associated with a certain status change is dependent on the roles of the participants and the stage of the software release cycle. The results reinforce the threats involved in mining bug repositories and provide a way to estimate the inaccuracy in data. We recommend that future researchers use our method prior to mining bug repositories in order to select projects in which the meaning of the data records match their expectations.

**Keywords**-software evolution; software development practices; bug repositories; empirical study;

## I. INTRODUCTION

## II. BACKGROUND

### A. *Bug Tracking Systems*

Bug workflow. Bugzilla. [1]

### B. *The Eclipse Project*

History.

Recommendations about Bugzilla usage. Four-eyes principle.

### C. *The Netbeans Project*

History.

Recommendations for developers.

## III. EXPERIMENTAL SETUP

Data extraction

Data processing:

- filtering (bugs up to december 20xx)
- classification script

Sampling:

- random sampling (for coding)
- entire data set

Analysis:

- grounded theory (open coding)
- latent Dirichlet allocation
- descriptive statistics

## IV. RESULTS

## V. CONCLUSION

## ACKNOWLEDGMENT

The first author is supported by FAPESB under grant BOL0119/2010.

## REFERENCES

- [1] C. Bird, A. Bachmann, E. Aune, J. Duffy, A. Bernstein, V. Filkov, and P. Devanbu, "Fair and balanced?: bias in bug-fix datasets," in *Proceedings of the the 7th joint meeting of the European software engineering conference and the ACM SIGSOFT symposium on The foundations of software engineering*, ser. ESEC/FSE '09. New York, NY, USA: ACM, 2009, pp. 121–130. [Online]. Available: <http://doi.acm.org/10.1145/1595696.1595716>