

# Identifying and Predicting Key Features to Support Bug Reporting

Md. Rejaul Karim\*, Akinori Ihara<sup>†</sup>, Eunjong Choi<sup>‡</sup>, and Hajimu Iida\*

\*Nara Institute of Science and Technology (NAIST), Japan

<sup>†</sup>Wakayama University, Japan

<sup>‡</sup>Kyoto Institute of Technology, Japan

Email: rejaul.karim.qw4@is.naist.jp, ihara@sys.wakayama-u.ac.jp, echoi@kit.ac.jp, iida@itc.naist.jp

**Abstract**—This paper empirically investigates the features of the bug reports that are important for the developers to triage and fix bugs. Through a case study of three projects (Camel, Derby, and Wicket) from the Apache ecosystem and two projects (Firefox and Thunderbird) the Mozilla ecosystem, we identify five key features that developers often request during bug fixing. We also find that missing these key features in the initial bug reports significantly affect the bug fixing process. In addition, we build classification models using four popular classification techniques to predict whether reporters should provide certain key features to make a good bug report by providing a minimum number of features at the initial submission.

This paper is an extended abstract of a paper accepted in the *Journal of Software: Evolution and Process (JSEP)*. Now, the paper is under the publication pipeline that is going to be published in the upcoming issue. The original paper is attached with this abstract.

## I. OVERVIEW

Bug reports are the primary means through which developers triage and fix bugs. A typical bug report contains input fields (e.g., a summary and description), which contain unstructured features such as what the reporters saw happen (Observed Behavior), what they expected to see happen (Expected Behavior), and a clear set of instructions that developers can use to reproduce the bugs (Steps to Reproduce). These unstructured features are crucial for the developers to triage and fix the bugs [1]. However, reporters often omit these features in their bug reports [2]–[5]. A previous empirical study on the eclipse project found that 78.1 percent of bug reports contain fewer than 100 words [6]. Thus, developers have to spend much time and effort in order to understand the bugs based on features provided or need to ask reporters to provide additional features [2], [7], [8]. Hence, a well-described bug report is crucial to improving the bug-fixing process.

To better understand which features are important, in this paper, we perform an exploratory study on five OSS projects using qualitative and quantitative analyses. In particular, we first perform a qualitative analysis to identify the key features by examining those that are provided initially (i.e., features that reporters provide in the initial bug reports), as well as additional required features (i.e., features that reporters missed in their initial submission, but that were required by developers to fix bugs). Through qualitative analysis, we identify five key

features that reporters often miss in their initial bug reports and developers require them for fixing bugs.

One of the main reasons for the lack of unstructured features in bug reports is inadequate tool support [2], [9]–[11]. Thus, in order to help reporters, we build prediction models to predict whether reporters should provide certain key features to make a good bug report by leveraging historical bug-fixing activities. We build prediction models using four popular text-classification techniques since the performance of prediction models varies between the text-classification techniques [12], [13] depending on the context. We evaluate our models using the bug reports of Camel, Derby, Wicket, Firefox, and Thunderbird projects. Our models achieve promising f1-scores when predicting key features. Our comparative study of the classification techniques reveals that NBM outperforms the other techniques in terms of predicting key features. We believe that Our findings can benefit reporters to improve the contents of bug reports.

## II. SATISFYING THE CRITERIA FOR JOURNAL-FIRST PAPER

Accepted Date	This paper is accepted on May 13, 2019 for the the Journal of Software: Evolution and Process (JSEP).
The paper is in the scope of the conference.	This paper fits under software engineering research topics such as empirical studies of software maintenance and evolution, change and defect management, and mining software repositories.
The paper reports completely new research results and/or presents novel contributions that significantly extend and were not previously reported in prior work	This paper reports several findings of an empirical study and proposed an approach in an effort to improve the bug management process.
The paper does not extend prior work solely with additional proofs or algorithms (or other such details presented for completeness), additional empirical results, or minor enhancements or variants of the results presented in the prior work.	This paper is not an extension to any of our prior work or that of others.
The paper has not been presented at, and is not under consideration for, journal-first programs of other conferences.	The paper and its material has not been presented at, and is not under consideration for, other conferences.

### III. SUMMARY

In this journal, we have reported completely new research results of an exploratory study on five OSS projects from two ecosystems using qualitative and quantitative analyses. Our research work is closely related to the three main research topics of the ICSME 2019. We have not presented our findings at any conference. Thus, we strongly believe that our paper satisfies the ICSME journal first track criteria for the above reasons. The ICSME 2019 journal first track would be a great opportunity for us to present and share our findings in the international software engineering community.

### REFERENCES

- [1] N. Bettenburg, S. Just, A. Schroter, C. Weiss, R. Premraj, and T. Zimmermann, "What makes a good bug report?" in *16th ACM SIGSOFT International Symposium on Foundations of Software Engineering, SIGSOFT '08/FSE-16*, 2008.
- [2] S. Brey, R. Premraj, J. Sillito, and T. Zimmermann, "Information needs in bug reports: Improving cooperation between developers and users," in *2010 ACM Conference on Computer Supported Cooperative Work, CSCW '10*, 2010.
- [3] S. Davies and M. Roper, "What's in a bug report?" in *8th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, ESEM '14*, 2014.
- [4] J. Aranda and G. Venolia, "The secret life of bugs: Going past the errors and omissions in software repositories," in *31st International Conference on Software Engineering, ICSE '09*, 2009.
- [5] T. Dal Sasso, A. Mocci, and M. Lanza, "What makes a satisficing bug report?" in *IEEE International Conference on Software Quality, Reliability and Security (QRS), QRS '16*, 2016.
- [6] T. Zhang, J. Chen, H. Jiang, X. Luo, and X. Xia, "Bug report enrichment with application of automated fixer recommendation," in *25th International Conference on Program Comprehension, ICPC '17*, 2017.
- [7] M. R. Karim, A. Ihara, X. Yang, H. Iida, and K. Matsumoto, "Understanding key features of high-impact bug reports," in *International Workshop on Empirical Software Engineering in Practice, IWESEP*, 2017.
- [8] M. Erfani Joorabchi, M. Mirzaaghaei, and A. Mesbah, "Works for me! characterizing non-reproducible bug reports," in *11th Working Conference on Mining Software Repositories, MSR '14*, 2014.
- [9] T. Zimmermann, N. Nagappan, P. J. Guo, and B. Murphy, "Characterizing and predicting which bugs get reopened," in *34th International Conference on Software Engineering, ICSE '12*, 2012.
- [10] O. Chaparro, J. Lu, F. Zampetti, L. Moreno, M. Di Penta, A. Marcus, G. Bavota, and V. Ng, "Detecting missing information in bug descriptions," in *2017 11th Joint Meeting on Foundations of Software Engineering, ESEC/FSE '17*, 2017.
- [11] "An open letter to github from the maintainers of open source projects. available online: <https://github.com/dear-github/dear-github>," 2016.
- [12] X.-L. Yang, D. Lo, X. Xia, Q. Huang, and J.-L. Sun, "High-impact bug report identification with imbalanced learning strategies," *Computer Science and Technology*, vol. 32, 2017.
- [13] X. Xia, D. Lo, W. Qiu, X. Wang, and B. Zhou, "Automated configuration bug report prediction using text mining," in *IEEE 38th Annual International Computers, Software and Applications Conference, COMPSAC '14*, 2014.