Predicting bug-fixing time: A replication study using an open source software project (link)

Shirin Akbarinasaji[1] shirin.akbarinasaji@ryerson.ca

Bora Caglayan bora.caglayan@ryerson.ca

Ayşe Başar ayse.bener@ryerson.ca

Original Paper:

Predicting Bug-Fixing Time: An Empirical Study of Commercial Software Projects (link)

Hongyu Zhang, Liang Gong, Steve Versteeg, Jue Wang, Zeqi Shen, and Janine Radford

WHAT: In this work, we aim to replicate a model for predicting bug fixing time with open source data from Bugzilla Firefox. We conduct the same RQs as the original study:

- How many bugs can be fixed in a given amount of time?
- How long would it take to fix a given number of bugs?
- How much time is required to fix a particular bug?

WHY: Estimates of the time required to fix known bugs (the "bug fixing time") would assist managers in allocating bug fixing resources when faced with a high volume of bug reports. The key factor for scheduling the order of defect reports is a quantitative understanding of the time constraints for fixing the bugs and the effect of postponing bug fixing activities on the bug resolution time. Instead of starting afresh, we relied on a fixing time prediction model from the literature. We chose Zhang et al.'s study mainly because their bug handling model considers the "Deferred" state. Additionally, their research questions can serve as a preliminary analysis for our future research goal of managing deferred bugs. Hence, the purpose of this replication is twofold: first, it provides insight for our academic research program; second, it verifies the merits of the original study on an open source data.

HOW: They conducted the experimental study on three commercial software projects by CA Technologies. CA Technologies is an IT management solutions company. To address their research questions, Zhang et al. defined a simplified bug handling process. Initially, bugs are identified and submitted into a bug tracking system either by the QA team or by end users. Once the bugs are confirmed, they are assigned to the developer team for resolution. They also verified whether the bugs were prioritized correctly. Hence, as an initial analysis, they explored the distribution of bug fixing time for those projects and performed some exploratory analysis on the fixing times of the bugs based on their severity and priority.

WHERE: The experiment of this study was performed on the Mozilla Firefox project. Originally, Zhang et al. performed an empirical study on three projects from CA Technologies. Therefore, the first difference is associated with the differences between open source and commercial bug tracking systems. We were able to extract all the features explained in the original study to construct a KNN classification scheme, with the exception of "ESC" and "Category." We substituted "Category" with "Component."

DISCUSSION: Original study was conducted on commercial project data from CA Technologies. To generalize the methodology, we replicated that study by using an open source bug tracking system. The results show that although the bug tracking system of the Firefox project may appear to be complex, in general, it is similar to the bug handling system of CA Technologies. Therefore, different states in the Markov model did not significantly affect the results. n the replicated study, we showed that the simplified process was a threat for their study and a finer Markov model would improve the accuracy slightly. Furthermore, compatible results indicate that their model is robust enough to be generalized, and we can rely on it in our future study on managing deferred bugs in an issue tracking system. Going forward, we can rely on this model as a baseline and determine whether deferring bugs would change their expected resolution time in management of defect debt.

[1] *corresponding author, GitHub ID: ShirinAkbari;*