BUG LOCALIZATION BY EMBEDDING VERSION TAGS IN SOFTWARE DELIVERABLE FILES TO IMPROVE SOFTWARE RELIABILITY

ABSTRACT

Practically every software development process has bugs. It will take a considerable amount of time, effort, and money to solve them all, or at least a sizable portion of them. In recent years, a number of academics have performed bug tracking analyses to better understand the issue and, consequently, propose solutions to reduce costs and boost the effectiveness of the bug-fixing task. Debugging and fixing software bugs takes a lot of the time and effort from software engineers. Defects are reported in issue or bug reports by a variety of stakeholders, such as end users, testers, and even developers.

Since then, it has been discovered that one of the time-consuming tasks performed by software developers is debugging. Programmers may need to thoroughly examine the code, use a debugger, collaborate with other programmers, write test cases, use potential fixes, and run regression tests in order to find bugs in the programme. Debugging costs the industry billions of dollars annually, claims a NIST report. Given how challenging it is to find and apply a fix for an existing bug, it is preferable to apply the fix everywhere the bug actually manifests itself. Therefore, automating bug localization in a software project can help engineers concentrate on important tasks by searching for potentially buggy files and then following the sequence of code lines that the bug has affected.

However, lexical mismatch is a significant problem that current automated bug localization systems must deal with. The terms and code tokens used to describe bugs in source files and bug reports are two different things. We offer a special method or model that makes use of local versioning to accomplish strong and significant bug localization accuracy in order to address this. This justification explains why this PhD thesis has documented important contributions to bug tracking and bug localization.

This doctoral research's initial efforts focused on understanding the modern contributions related to maximising bug-fixing speed and accuracy, which was accomplished through the modern validation of recent literature. The doctoral dissertation "An Incremental Risk Management Framework (IRMF) for Realizing Project Efficiency Using Version Control" also provided an ideal bug localization and tracking method. The objective of the second contribution, "Bug Localization by Embedding Version Tags," is to accelerate bug localization through the incorporation of version tags into release builds. The research's concluding contribution, titled "Best Bug Fixing Rate and Bug Fixing Time Using Software Reliability Modelling," presented a novel software reliability model to achieve the best bug fixing rate and bug fixing time in soft maintenance.

On actual software projects, this research's contributions were assessed against the state of the art for bug localization, yielding competitive results both within and across projects. This study makes contributions by demonstrating accuracy in bug localization, which plots the bug's location

and affected code sequences. This might provide a more comprehensive context to support repairing bugs quickly.

Complex and incredibly dynamic due to developers' constant code commits, addition of new features, bug fixes, and subsequent introduction of new bugs. Several hundred commits could be deployed at once. Therefore, localizing the bug to the appropriate commit is one of the most time-consuming yet challenging tasks for minimizing service disruption.