An Empirical Analysis of Reopened Bugs Based on Open Source Projects

**ABSTRACT**

Background: Bug fixing is a long-term and time-consuming activity. A software bug experiences a typical life cycle from newly reported to finally closed by developers, but it could be reopened afterwards for further actions due to reasons such as unclear description given by the bug reporter and developer negligence. Bug reopening is neither desirable nor could be completely avoided in practice, and it is more likely to bring unnecessary workloads to already-busy developers. Aims: To the best of our knowledge, there has been little previous work on software bug reopening. In order to further study in this area, we perform an empirical analysis to provide a comprehensive understanding of this special area. Method: Based on four open source projects from Eclipse product family, they are CDT, JDT, PDE and Platform, we first quantitatively analyze reopened bugs from perspectives of proportion, impacts and time distribution. After initial exploration on their characteristics, we then qualitatively summarize root causes for bug reopening, this is carried out by investigating developer discussions recorded in Eclipse Bugzilla. Results: Results show that 6%-10% of total bugs will lead to reopening eventually. Over 93% of reopened bugs place serious influence on the normal operation of the system being developed. Several key reasons for bug reopening have been identified in our empirical study. Conclusions: Although reopened bugs have significant impacts on both end users and developers, it is quite possible to reduce bug reopening rate through the adoption of appropriate methods, such as promoting effective and efficient communication among bug reporters and developers, which is supported by empirical evidence in this study.

This paper was presented at the 20th International Conference on Evaluation and Assessment in Software Engineering, June 1-3, 2016, Limerick, Ireland

Supervisor: Dr Jacky Keung

Research Interests: Software Engineering; Defect Reporting; Empirical Experiments; Metrics and Measurements; Data Mining and Analytics; Machine Learning

All are welcome!