Empirical Study of Test Coverage, Unit Tests and Single Line Bugs

Saqib Ameen and Habibur Rahman

1   Introduction

Writing unit tests is a common practice in industry to ensure software quality. Usually meeting a certain criteria of test coverage serves as a segway to the product release. Despite all efforts, the software remains error prone. Single line bugs are one of the frequent types of bugs found in programs. Bugs that appear in just a single line and can be fixed by modifying that line are called single line bugs. They are one of the most common and easy to fix types of bugs. However, the very nature of them makes it difficult to identify them. Remember the Apple return bug?

This study aims at analyzing the relation of single statement bugs with test coverage and unit test types. The results could be helpful in future to direct the testing efforts during the software development life cycle. Further, they can also be used in automatic bug detection and program repair.

2   Research questions

The goal of this project is to answer the following: *(1). Is there a relation between test coverage and single line bugs?, (2). Is there a relation between certain types of unit tests and their effectiveness on different types of single line bugs?*

Our hypothesis is that there is a weak to moderate relationship between single line bugs and test coverage. We also believe that certain types of unit tests will be effective to mitigate certain types of single line bugs.

3   Related Work

Empirical studies [5, 2, 6] at different scales and under different settings have found a no, weak or moderate correlation between coverage and test case. These studies examined all types of bugs that were found in the systems. A part of literature focuses on a subset of bugs such as easy to fix one-line bugs, mostly in the context of program repair. However, There is a gap in study related to the effectiveness of test suites for such bugs. Previously there was no large dataset available to conduct such studies. The recent studies in automatic program fixing have produced large classified [7] and unclassified [3] datasets of one-line bugs.

In our study, we aim to fill the gap by studying the relation of unit testing and small one line bugs.

4   Methodology

To address our research questions, we will use ManySStuBs4J[7] dataset which is a corpus of simple fixes to Java bugs. In the dataset, it has divided the single line fixes into 16 different categories like Change Identifier names, Same Function Swap Args, Change Operand and so on. To collect the respective projects we will use GHTorrent [4]. For measuring code coverage for each project based on the collected test cases, we will use JaCoCo [1], a free code coverage library for Java. Furthermore, we will write our own python scripts, as needed, to mine data.

To find out the correlation between the test suite present in the project and SStuBs we will mine the relevant project repositories for the presence of the tests. To get a better understanding of this relationship, we check the test cases for code smell as well using [—-]. To investigate the relation between test types and different types of SStuBs, we plan to utilize code coverage.

References

[1] EclEmma - JaCoCo Java Code Coverage Library.

[2] Vard Antinyan, Jesper Derehag, Anna Sandberg, and Miroslaw Staron. Mythical unit test coverage. *IEEE Software*, 35(3):73–79, 2018.

[3] Zimin Chen, Steve James Kommrusch, Michele Tufano, Louis-Noël Pouchet, Denys Poshyvanyk, and Martin Monperrus. Sequencer: Sequence-to-sequence learning for end-to-end program repair. *IEEE Transactions on Software Engineering*, 2019.

[4] Georgios Gousios. The ghtorrent dataset and tool suite. In *Proceedings of the 10th Working Conference on Mining Software Repositories*, MSR '13, pages 233–236, Piscataway, NJ, USA, 2013. IEEE Press.

[5] Lucas Gren and Vard Antinyan. On the relation between unit testing and code quality. In *2017 43rd Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, pages 52–56. IEEE, 2017.

[6] Laura Inozemtseva and Reid Holmes. Coverage is not strongly correlated with test suite effectiveness. In *Proceedings of the 36th international conference on software engineering*, pages 435–445, 2014.

[7] Rafael-Michael Karampatsis and Charles Sutton. How often do single-statement bugs occur? the manysstubs4j dataset. In *Proceedings of the 17th International Conference on Mining Software Repositories*, pages 573–577, 2020.