

# How is testing related to Single Statement Bugs?

*Saqib Ameen & Habibur Rahman*

## 1 Introduction

Writing unit tests is a common practice in the 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 prone to bugs. Among them, single statement bugs are one of the common types of bugs found. Bugs that appear in just a single statement and can be fixed by modifying that line are called single statement bugs. The very nature makes it difficult to identify. 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 the 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 questions: *1. Is there a relationship between test coverage and single statement bugs? 2. Is there a relation between certain types of unit testing and their effectiveness on different types of single statement bugs?* We hypothesize 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 in mitigating certain types of single line bugs.

## 3 Related Work

Empirical studies [4, 1, 5] 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 the 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 the literature related to the effectiveness of test suites for single statement bugs. Previously there was no large dataset available to conduct such studies. The recent studies in automatic program fixing have produced large classified [6] and unclassified [2] datasets of one-line bugs.

In our study, we aim to contribute to this part of the literature by studying the relationship between unit testing and single statement bugs.

## 4 Methodology

To address our research questions, we will use the ManySStuBs4J[6] dataset which is a corpus of one-line bugs in open source Java projects. In the dataset, it has divided the single line bugs into 16 different categories like Change Identifier names, Same Function Swap Args, Change Operand, and so on. We will use GHTorrent [3] to search for respective projects and then clone them for our study. For measuring the code coverage for each project, based on the collected test cases, we will use JaCoCo — a free code coverage library for Java. Furthermore, we will write our own python scripts, as needed, to mine data.

We will rely on the test coverage of the projects and the number of bugs present to establish the relationship between single line bugs and test coverage. Further, we will check the quality of the test suits (code smells) present to see if they affect the relationship.

To investigate the effectiveness of the certain type of test coverages, e.g., branch, statement, and functional, on different types of SSuBs, we will use the coverage report to identify the coverage of each test type.

## References

- [1] V. Antinyan, J. Derehag, A. Sandberg, and M. Staron. Mythical unit test coverage. *IEEE Software*, 35(3):73–79, 2018.
- [2] Z. Chen, S. J. Kommrusch, M. Tufano, L.-N. Pouchet, D. Poshyvanyk, and M. Monperrus. Sequencer: Sequence-to-sequence learning for end-to-end program repair. *IEEE Transactions on Software Engineering*, 2019.
- [3] G. 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.
- [4] L. Gren and V. 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.
- [5] L. Inozemtseva and R. Holmes. Coverage is not strongly correlated with test suite effectiveness. In *Proceedings of the 36th international conference on software engineering*, pages 435–445, 2014.
- [6] R.-M. Karampatsis and C. 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.