Efficient Test Case Prioritization in Regression Testing

Prashant Malangave
Walchand Collage of Engineering
Dept. of Computer Science & Engg.
pmgave@yahoo.com

Dr. Dinesh B. Kulkarni
Walchand Collage of Engineering
Dept. of Computer Science & Engg.
d_b_kulkarni@yahoo.com

Abstract

Regression Testing is an important and costly activity of the software maintenance lifecycle. In theory several Regression test selection and prioritization techniques are specified, based on the past history, coverage, risk & specification. The coverage based technique has potential for practical implementation in the industrial environment. This paper aims to discuss a Framework through a case study which prioritizes the test cases using code coverage during regression testing and experimental results

1. Introduction

Optimizing the time and cost of the regression testing without compromising the fault exposing capability is always challenging for the testing team. Testing team always face constraints like lack of resources, squeezed testing schedule, changing & ambiguous requirement, which in terms impacts and reduces the effectiveness of regression testing

The Test automation tool will help testing team speed-up the test execution. But prioritizing the test cases using code coverage information helps testing team to improve fault detection rate during testing. This paper discuss the regression test framework which orders test cases based on code coverage during execution helps in improving effectiveness of regression testing.

2. Problem Statement

Practically it is not possible to perform rigorous testing. Tester has to select subset of test cases from the original test suite. This makes test case selection quite challenging. This selected regression suite should cover all the functionality i.e. adequate functional coverage and greater fault exposing potential.

Due to squeezed test schedule, testing team may not able to execute all test cases from the selected regression suite. Sequencing of test cases based on some criteria helps testing team to achieve the goals whilst reducing testing cycles.

The success of regression test selection and prioritization techniques is mostly depend on actor i.e. tester. There are no commercial tools available which assist the tester in test case selection. The test automation tool only helps in reducing execution time and not in selection or prioritization.

There are more chances of variation in results achieved from test suites prepared by different tester.

3. Methodology

The subjectivity in regression testing process can be reduced by appropriately selecting test suit and ordering/sequencing it based on some criteria.

Regression Test Selection: Tester should understand impact of changes in the applications with help of business analyst and development team. This affected portion of code should be main focus in regression testing. The

Requirement traceability table gives the test cases covering the affected portion of application.

The exercise discussed above helps the tester in preparing the regression test suit manually.

Test Case Prioritization: Test cases can be prioritized based on the code coverage criteria i.e. the test case which covers the maximum code executed first [1] [2]. This method ensures the execution of test cases with more fault exposing. Thus fault detection rate can be increased. This task can be automated using Automation, Coverage analysis tool

If the test cases covering the affected functionality, executed earlier in testing phase increases the chance of finding defects in the earlier stage. This gives more time for developer to fix the defect which ultimately helps in reducing the cost of testing.

4. Regression Test framework

The regression test prioritization discussed in the section 3 is implemented as Test Framework built over a standard test automation tool Rational Functional Tester. The prioritizer orders the test cases based on code coverage information like lines of code, methods, and blocks. The code coverage information is collected during actual execution by the framework and stored in repository. This data is analyzed and used in prioritization during subsequent execution cycle(s). The framework is integrated with coverage analysis tool EMMA, an open source tool, which collects code coverage information of each test case during run time.

5. Experimental results

This section describes the study conducted to evaluate the effectiveness test prioritizer. The study is conducted on industrial applications that comprised 200 KLOC. The regression test selection is done using the process discussed in section 3. This selected test suit is prioritized and executed by the test framework. The prioritized and unprioritized test suits are executed on the same version of application. The criteria used for prioritization is the total number LOC covered by test cases. The un-prioritized test suite is prepared based on the random selection.

The effectiveness of regression test prioritizer is measured using following criteria.

5.1 Rate of fault detection

This is an important objective of regression testing. It increases the likelihood of early fault detection during execution of regression suite using prioritized test suite [3].

The numbers of defects detected after certain interval are analyzed. The study shows that the rate is much higher for prioritized test suite.

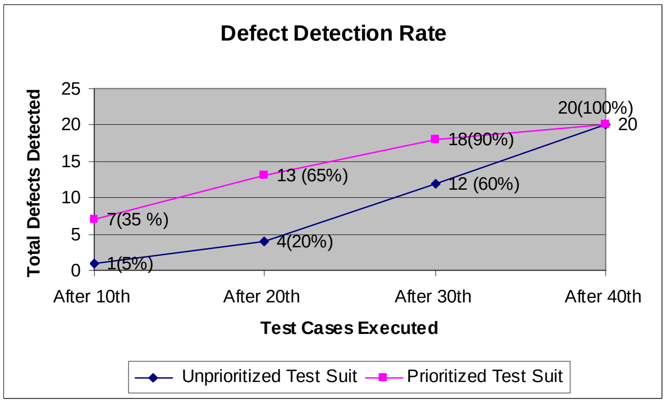

Figure 1 Defect Detection Rate of Unprioritized versus Prioritized Test Suite

5.2 Code coverage achieved

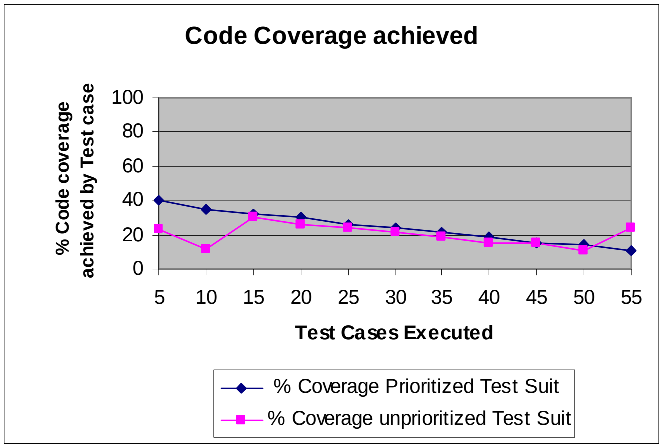

Figure 2 code coverage criteria achieved by Test cases

This measures the rate at which the code coverage criteria are achieved. The coverage effectiveness is one of the measures for the quality of the test suite prioritization [3]. The code coverage % is analyzed after certain execution interval for both test suites. The study shows that the prioritized test cases achieve greater coverage in earlier execution phase than the un- prioritized test cases. The % code coverage decreases as the execution moves in case prioritized test suite. In un-prioritized test suite execution, it varies period to period and depends on the test cases executed during that period.

6. Conclusion

In this paper we have discussed the practical problem which lowers the effectiveness of regression testing in industrial environment. Section 3 suggests the use of test prioritization in conjunction with the test selection. This increases likelihood of defect detection related to affected portion of the code during the earlier phase of testing. We have discussed the practical implementation of the test prioritizer as extension to Test Automation tool in section 4.

This paper also presents the results of empirical studies that evaluate test prioritizer.

The initial result shows that 25- 30 % gain is achieved in fault detection rate by prioritized test suite. The defects related affected portion of application are detected earlier. The Average Percent of Fault Detected (APFD) measure is also higher by 20 – 25% in case of prioritized test suite.

The results may vary if prioritization is done using different criteria like methods, block, classes or its combination. If different tool is used for coverage analysis, result of test prioritization may vary as different techniques are used by the tool for measuring coverage. The performance of Automated test execution is reduced as testing is done on instrumented version of application.

The study has shown that the prioritization of the appropriately selected test suite using code coverage can significantly increase the fault detection rate at earlier phase of regression testing as compared to the un-prioritized test suite.

7. Reference

[1] Gregg Rothermel, R. H. Untch, C. Chu, M. J. Harrold, “Prioritizing Test Cases For Regression Testing” , IEEE Transactions on Software Engineering. Vol. 27, No. 10, October 2001, pages 929-948.

[2] Gregg Rothermel, R. H. Untch, C. Chu, M. J. Harrold, “Test Case Prioritization: An Empirical Study”, Proceedings of the International Conference on Software Maintenance, August 1999, pages 179-188

[3] Gregory M. Kapfhammer, Mary Lou Soffa, “Using Coverage Effectiveness to Evaluate Test Suit Prioritizations”, Proceedings of the ACM International Workshop on Empirical Assessment of Software Engineering Languages and Technologies, Atlanta, Georgia, November 2007.