IJISET - International Journal of Innovative Science, Engineering & Technology, Vol. 1 Issue 6, August 2014.
www.ijiset.com
ISSN 2348 – 7968

Prioritizer: Prioritizing test cases for regression testing

Nikita Goel, Madhuri Sharma

Department of computer science & engineering, UPTU
Bharat Institute of Technology, Partapur By-pass Meerut, India

Abstract

In software development, Regression testing is a costly process. It is the process of retesting the modified parts of the software and ensuring that no new errors have been introduced into previously tested source code due to these modifications. Regression test selection plays an important role in it. A regression test selection technique selects an appropriate number of test cases from a test suite that might expose a fault in the modified program.

***Keywords*— Regression testing, Prioritization, Test case, Test suite, statement coverage.**

I. INTRODUCTION

Test Case Prioritization for means to organise test cases which aims at increasing the efficiency and effectiveness of goals. Some possible goals of prioritization are: increasing the rate of fault detection of a test suite; increasing the coverage of code in the system which is under test; increasing the rate of high - risk faults detection . In section 2 of this paper we have defined regression testing in detail, its procedure and discussed two techniques to reduce the regression testing cost: Regression Test Selection and Test Suite Minimization. In section 3 we have described the test case prioritization which attempts to increase the fault detection rate of a test suite, then moving into the advantages and goals of test case prioritization. In section 4 we have included a catalog of several prioritization techniques and a detailed description of concept of statement coverage. Section 5 presents overall conclusion and then its next section precisely states future work.

II. REGRESSION TESTING

Regression Testing means [1], [20] an execution of the collection of test cases on a program in order to ensure that its revision does not produce faults. Regression testing is costly process and is defined as [2] the process of retesting the modified parts of the software in which test suites are executed with an intend of ensuring that no new errors have been introduced in previously tested code.

Let P be a program [3], P' be a modified version of P and T be a test suite for P program where Test suite can be defined [4] as a collection of test cases that are intended to be used to test a software program to show that it has some specified set of behaviours. A test suite often contains detailed instructions or goals for each collection of test cases and information on the system configuration to be used during testing. A regression testing means to reuse T on P' and checking out whether previous functionality is working fine, and ensuring that no new errors got introduced.

As described in [5], a typical regression testing procedure can be elaborated as:

Step 1 Involves the regression test selection i.e. selecting a subset T' of T in order to test P'

Step 2 Involves test suite execution i.e. test P' with T' means executing the test suites and checking test results to measure the correctness of P'

Step 3 Involves the identification of code coverage i.e. to determine whether P' has new functionality which requires to create new test cases for P'

Step 4 Involves executing new test cases to test P', establishing correctness

Step 5 Involves the test suite maintenance i.e. updating and storing test information means maintaining test execution profile for P'

A. *REGRESSION TEST SELECTION*

Regression test [16] selection technique is performed to lower the regression testing cost as it selects a subset of the test suite to rerun them inspite of rerunning the whole set of test suite. Selection of test cases, from existing test suite, is done on the basis of information about the program, modified version of program, and already existing test suite. This technique costs less than rerunning whole test suite as regression test selection reduces the size of test suite by selecting an appropriate subset of test suite. We focuses advantage of this technique: Regression test selection may create new test cases which test the program for areas which are not covered by existing set of test cases. This technique have drawback also, although there are safe regression test selection techniques ([14], [15], [16], [17]) that can ensure that the subset of a test suite, the conditions under which safety can be achieved do not always hold.

B. *TEST SUITE MINIMIZATION*

Test suite minimization technique selects minimum number of test cases from test suite taking test coverage criteria into account. We focuses an advantage of test suite minimization technique: Reduced set of test suite covers equivalent amount of code as the original test suite covers. This technique have drawback also that the fault detection capabilities of test suites can be compromised by minimization.

ISSN 2348 – 7968

III. TEST CASE PRIORITIZATION

Test case prioritization technique line up the test cases in order to increase the rate of fault detection of test suite. Fault detection rate [19] means a measure of how fast faults are detected in the testing process. These techniques [6], [7] enables testing engineers to organise the test cases with an aim that the test cases which have highest priority executes earlier than the test cases the test cases which have lower priority. Some criterion based on which prioritization of test cases takes place are: to achieve maximum code coverage in a program; to increase the fault detection rate.

We define the test case prioritization problem [12] as follows: Let T be a test suite, PT be the set of permutations of T, f be a function from PT to the real numbers.

Problem is to find T' belongs to PT such that (for all T'') (T'' belongs to PT) (T'' is not equal to T') [f(T') greater than equal to f(T'')]. PT represents the set of all possible prioritizations of T and f is a function that, applied to any such ordering.

Test case prioritization technique [13] is implemented taking some of the following goals into account:

- To increase the fault detection rate of a test suite. It means that how quickly the faults are found out in the process of regression test.
- To increase the coverage of code in the system which is to be tested. It means increasing the amount of code coverage hence meeting the criterion of code coverage.
- To increase the rate of detecting those faults earlier which are high risk faults.
- To speed up the process of release of the software.

We mention few advantages of technique of prioritizing the test cases for regression testing [22] :

- An improved rate of fault detection in regression testing process results in providing feedback earlier.
- Due to improved rate of fault detection, engineers can perform their debugging activities.
- It leads to several cost benefits.

IV. TECHNIQUES OF PRIORITIZATION

NO.	NAMES OF TECHNIQUES
1	NO PRIORITIZATION
2	RANDOM PRIORITIZATION
3	OPTIMAL PRIORITIZATION
4	TOTAL STATEMENT COVERAGE PRIORITIZATION
5	ADDITIONAL STATEMENT COVERAGE PRIORITIZATION
6	TOTAL BRANCH COVERAGE PRIORITIZATION
7	ADDITIONAL BRANCH COVERAGE PRIORITIZATION
8	TOTAL FAULT-EXPOSING-POTENTIAL (FEP) PRIORITIZATION
9	ADDITIONAL FAULT-EXPOSING-POTENTIAL (FEP) PRIORITIZATION

Table 1 A Catalog of Prioritization Techniques

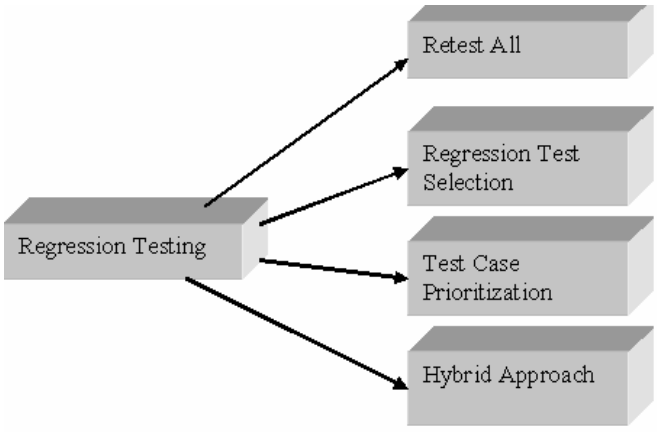

Figure1 Regression Testing Techniques

A. *STATEMENT COVERAGE*

The statement coverage can be calculated as shown below:

$$\text{Statement coverage} = \frac{\text{Number of statements exercised}}{\text{Total number of statements}} \times 100\%$$

To understand the statement coverage in a better way lets take an example which is basically a pseudo-code, it is not any specific programming language:

READ X

READ Y

IF X>Y THEN Z=0

END IF

ISSN 2348 – 7968

To achieve 100% statement coverage of this code segment just one test case is required, one which ensures that variable A contains a value that is greater than the value of variable Y, for example, X = 12 and Y = 10. Now, let's take another example where we will measure the coverage first. In order to simplify the example, we will regard each line as a statement. A statement may be on a single line, or it may be spread over several lines. One line may contain more than one statement, just one statement, or only part of a statement. Some statements can contain other statements inside them. In the code sample below we have two read statements, one assignment statement, and then one IF statement on three lines, but the IF statement contains another statement (print) as part of it.

```
1 READ X
2 READ Y
3 Z =X + 2*Y
4 IF Z> 50 THEN
5 PRINT large Z
6 ENDIF
```

Let us consider some test cases:

Test 1Case: X= 2, Y = 3
Test Case2: X =0, Y = 25
Test Case: X =47, Y = 1

In Test Case 1, the value of Z will be 8, so we will cover the statements on lines 1 to 4 and line 6.

In Test Case 2, the value of Z will be 50, so we will cover exactly the same statements as Test Case1.

In Test Case 3, the value of Z will be 49, so again we will cover the same statements. Since we have covered five out of six statements, we have 83% statement coverage (with three tests).

Now we take another example of test case:

Test Case 4: X = 20, Y = 25

This time the value of Z is 70, so we will print 'Large Z' and we will have exercised all six of the statements, so now statement coverage = 100%. Notice that we measured coverage first, and then designed a test to cover the statement that we had not yet covered. Note that Test Case 4 on its own is more effective which helps in achieving 100% statement coverage, than the first three tests together. Just taking Test Case 4 on its own is also more efficient than the set of four tests, since it has used only one test instead of four. Being more effective and more efficient is the mark of a good test technique.

IV. CONCLUSION

There are several techniques for prioritizing test cases for regression testing have the abilities to improve the rate of fault detection of test suites. Rate of fault detection means to improve the speed of detecting faults in regression testing. It is beneficial to order tests during their initial creation, for use in the initial testing of software. The technique increases confidence in the correctness of the modified program, will help in preserving the quality and reliability of the software. The test cases selected using the proposed technique will identify and locate errors in the modified or changed program and also ensure the software's continued operation.

V. FUTURE WORK

Regression testing is a type of software testing process [21], [25] that intends to uncover new software errors, or regressions, in already existing functional and non-functional areas of a system after modifications made. These modifications include configuration changes, enhancements etc. There can be number of test cases for a particular program but little resources so that main problem software engineers face is to determine which of those test cases should be executed. In order to solve this problem, we are in need of learning an art of prioritization technique of test cases for the purpose of regression testing. Prioritization here means to schedule the test cases for a program so that the task of performing regression testing would be effective and easy. Prioritizing the test cases is necessary to improve the fault detection rate. The future work in these areas can be to develop a framework which prioritize the test cases for a program.

REFERENCES

[1] Gaurav Duggal, Mrs. Bharti Suri, "Understanding Regression Testing Techniques", Proceedings of 2nd National Conference on Challenges and Opportunities in Information Technology.

[2] K.K.Aggarwal & Yogesh Singh, "Software Engineering Programs Documentation, Operating Procedures," New Age International Publishers, Revised Second Edition – 2005.

[3] Sebastian Elbaum, Praveen Kallakuri, Alexey G. Malishevsky, Gregg Rothermel, Satya Kanduri, "Understanding the Effects of Changes on the Cost-Effectiveness of Regression Testing Techniques," Journal of Software Testing, Verification, and Reliability, 13(2) pages:65-83, June 2003.

[4] http://en.wikipedia.org/wiki/Test_suite

[5] G. Rothermel and M. J. Harrold, Analyzing Regression Test Selection Techniques, IEEE Transactions on Software Engineering, V.22, no. 8, August 1996, pages 529-551.

[6] G. Rothermel, R.H. Untch, C. Chu, and M.J. Harrold, ªTest Case Prioritization: An Empirical Study,º Proc. Int'l Conf. Software Maintenance, pp. 179±188, Aug. 1999.

[7] W.E. Wong, J.R. Horgan, S. London, and H. Agrawal, ªA Study of Effective Regression Testing in Practice,º Proc. Eighth Int'l Symp. Software Reliability Eng., pp. 230±238, Nov. 1997.

[8] T.Y. Chen and M.F. Lau, ªDividing Strategies for the Optimization of a Test Suite,º Information Processing Letters, vol. 60, no. 3, pp. 135±141, Mar. 1996.

[9] M.J. Harrold, R. Gupta, and M.L. Soffa, ªA Methodology for Controlling the Size of a Test Suite,º ACM Trans. Software Eng. and

ISSN 2348 – 7968

Methodology, vol. 2, no. 3, pp. 270±285, July 1993.
[10] G. Rothermel, M.J. Harrold, J. Ostrin, and C. Hong, ªAn Empirical Study of the Effects of Minimization on the Fault Detection Capabilities of Test Suites,º Proc. Int'l Conf. Software Maintenance, pp. 34±43, Nov. 1998.
[11] W.E. Wong, J.R. Horgan, S. London, and A.P. Mathur, ªEffect of Test Set Minimization on Fault Detection Effectiveness,º Software± Practice and Experience, vol. 28, no. 4, pp. 347±369, Apr. 1998.
[12] G. Rothermel, R.H. Untch, C. Chu, and M.J. Harrold, "Prioritizing Test Cases for Regression Testing," IEEE Trans. Software Eng., vol. 27, no. 10, pages 929-948, Oct. 2001.
[13] Gregg Rothermel, Roland H. Untch, Mary Jean Harrold, IEEE Transactions on software engineering, Vol. 27, No. 10, October 2001.
[14] F.I. Vokolos and P.G. Frankl, ªPythia: A Regression Test Selection Tool Based on Textual Differencing,º Proc. Third Int'l Conf. Reliability, Quality & Safety of Software-Intensive Systems (ENCRESS '97), May 1997.
[15] T. Ball, ªOn the Limit of Control Flow Analysis for Regression Test Selection,º Proc. ACM Int'l Symp. Software Testing and Analysis, pp. 134±142, Mar. 1998.
[16] Y.F. Chen, D.S. Rosenblum, and K.P. Vo, ªTestTube: A System for Selective Regression Testing,º Proc. 16th Int'l Conf. Software Eng., pp. 211±222, May 1994.
[17] G. Rothermel and M.J. Harrold, ªA Safe, Efficient Regression Test Selection Technique,º ACM Trans. Software Eng. and Methodology, vol. 6, no. 2, pp. 173±210, Apr. 1997.
[18] Xuan Lin, "Regression Testing in Research And Practice", Computer Science and Engineering Department University of Nebraska, Lincoln, 2007.
[19] W.E. Wong, J.R. Horgan, S. London, and A.P. Mathur, "Effect of Test Set Minimization on Fault Detection Effectiveness," Software- Practice and Experience, vol. 28, no. 4, pp. 347-369, Apr. 1998.
[20] http://en.wikipedia.org/wiki/Regression_testing.

[21]http://tmcpfs.ops.fhwa.dot.gov/cfprojects/uploaded_files/Final%20Q&A.pdf.

[22]http://www.seguetech.com/blog/2012/08/10/importance-test-case-prioritization.

[23] Thillaikarasi Muthusamy, Dr. Seetharaman.K, "A Test Case Prioritization Method with Weight Factors in Regression Testing Based on Measurement Metrics", Vol.3, Issue12, dec.2013.
[24] B. Beizer, "Software Testing Techniques," Van Nostrand Reinhold, New York, 1990.
[25] Srinivasan Desikan, Gopalaswamy Ramesh, "Software Testing Principles and Practices", PEARSON