Various Techniques of Regression Testing

Sandeep Kaur[1], Rajandeep Kaur[2]

[1]Assistant Professor, [2]Lecturer

Sant Baba Bhag Singh Institute of Engg & Technology Jalandhar

Corresponding author e-mail:- nattsandeep@yahoo.co.in

Abstract: *Regression Testing is one of the most important and expensive activities of Software Maintenance. It involves testing the modified program to reveal faults introduced during maintenance and keeping the software's level of reliability, as cheaply as possible. Two techniques of regression testing are discussed. The first is a selective technique which identifies the modified structural attributes of a program (required elements) and selects a resettable test case set to exercise them. This technique is based on the Potential-Uses criteria family and is implemented by a regression testing tool working in conjunction with a testing tool. The second technique comprises procedures for testing and regression testing activities.. In this paper we have presented the various types of regression testing techniques their classifications.*

Keywords: *Software Maintenance, Regression testing, Test Case prioritization.*

Introduction

Software maintenance is an activity which includes enhancements, error corrections, optimization and deletion of obsolete capabilities. These modifications in the software may cause the software to work incorrectly and may also affect the other parts of the software, so to prevent this Regression testing is performed. Regression testing is used to revalidate the modifications of the software. Regression testing is an expensive process in which test suites are executed ensuring that no new errors have been introduced into previously tested code.

Background

Experience has shown that as software is fixed, emergence of new and/or reemergence of old faults is quite common. Sometimes reemergence occurs because a fix gets lost through poor revision control practices (or simple human error in revision control). Often, a fix for a problem will be "fragile" in that it fixes the problem in the narrow case where it was first observed but not in more general cases which may arise over the lifetime of the software. Frequently, a fix for a problem in one area inadvertently causes a software bug in another area. Finally, it is often the case that when some feature is redesigned, some of the same mistakes that were made in the original implementation of the feature were made in the redesign. Regression testing is an integral part of the extreme programming software development method. In this method, design documents are replaced by extensive, repeatable, and automated testing of the entire software package at every stage in the software development cycle.

Regression Testing

Regression testing is any type of software testing that seeks to uncover new errors, or regressions, in existing functionality after changes have been made to the software, such as functional enhancements, patches or configuration changes. The intent of regression testing is to assure that a change, such as a bug fix, did not introduce new bugs "One of the main reasons for regression testing is that it's often extremely difficult for a programmer to figure out how a change in one part of the software will echo in other parts of the software."Common methods of regression testing include rerunning previously run tests and checking whether program behavior has changed and whether previously fixed faults have re-emerged. Regression testing can be used to test a system efficiently by systematically selecting the appropriate minimum set of tests needed to adequately cover a particular change.

There are various regression testing techniques (1) Retest all; (2) Regression Test Selection; (3) Test Case Prioritization; (4) Hybrid Approach.

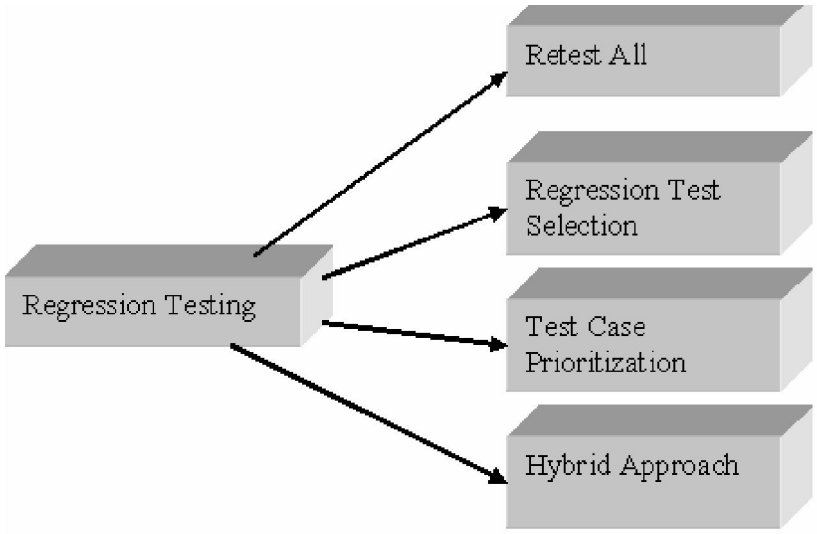

Figure1. Clasification of Regression Testing

Regression Test Selection (RTS)

Due to expensive nature of "retest all" technique, Regression Test Selection is performed. In this technique instead of rerunning the whole test suite we select a part of test suite to rerun if the cost of selecting a part of test suite is less than the cost of running the tests that RTS allows us to omit. RTS divides the existing test suite into (1) Reusable test cases; (2) Retest able test cases; (3) Obsolete test cases. In addition to this classification RTS may create new test cases that test the program for areas which are not covered by the existing test cases. RTS techniques are broadly classified into three categories. 1) Coverage techniques: they take the test coverage criteria into account. They find coverable program parts that have been modified and select test cases that work on these parts. 2) Minimization techniques: they are similar to coverage techniques except that they select minimum set of test cases. 3) Safe techniques: they do not focus on criteria of coverage; in contrast they select all those test

cases that produce different output with a modified program as compared to its original version.

Test Case Prioritization

This technique of regression testing prioritizes the test cases so as to increase a test suites rate of fault detection that is how quickly a test suite detects faults in the modified program to increase reliability. This is of two types:(1) General prioritization which attempts to select an order of the test case that will be effective on average subsequent versions of software .(2)Version Specific prioritization which is concerned with particular version of the software. The test case Prioritization having three techniques, 1) Comparator techniques**:** In this we select the test case using two ways, i) Random ordering ii) Optimal ordering ,, in random ordering the test cases in test suite are randomly prioritized. On other hand Optimal ordering the test cases are prioritized to optimize rate of fault detection. 2) Statement level technique: in which we used three ways to select the test case, i) Total statement coverage prioritization: in which test cases are prioritized in terms of total number of statements by sorting them in order of coverage achieved. ii) Additional statement coverage prioritization: which is similar to total coverage prioritization, but depends upon feedback about coverage attained to focus on statements not yet covered? iii) Total FEP prioritization: in which prioritization is done on the probability of exposing faults by test cases. Mutation analysis is used to approximate the Fault-Exposing-Potential (FEP) of a test case. 3) Function level techniques in this technique also we used three ways to select the test cases. i) Total function coverage prioritization: it is similar to total statement coverage but instead of using statements it uses functions. ii) Additional function coverage prioritization: it is similar to Additional statement coverage prioritization with only difference that instead of statements, it is considering function level coverage. iii) Total FEP prioritization (function level)U: it is analogous to Total FEP prioritization with only difference that instead of using statements it is using functions. iv) Additional FEP prioritization (function level): this technique is similar to Additional FEP prioritization with only difference that instead of using statements it is using functions.

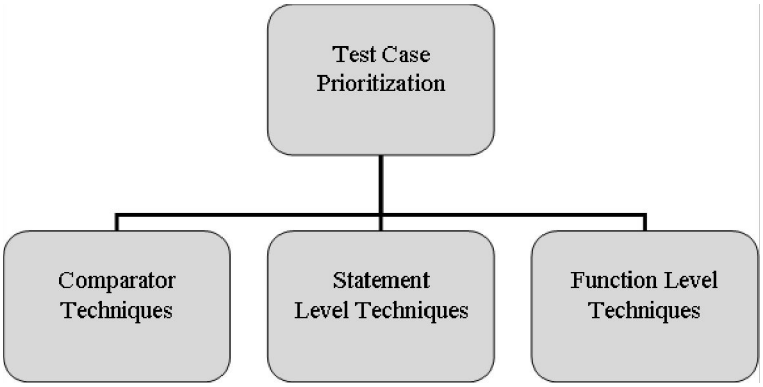

Figure2. Classification of test case prioritization

Hybrid Approach

The fourth regression technique is the Hybrid Approach of both Regression Test Selection and Test Case Prioritization

Application and Challenges

The application of regression testing, regression testing can be used not only for testing the correctness of a program, but often also for tracking the quality of its output. For instance, in the design of a compiler, regression testing could track the code size, simulation time and time of the test suite cases. Regression testing should be part of a test plan. Regression testing can be automated. Regression tests can be broadly categorized as functional tests or unit tests. Functional tests exercise the complete program with various inputs. Unit tests exercise individual functions, subroutines, or object methods. Both functional testing tools and unit testing tools tend to be third party products that are not part of the compiler suite, and both tend to be automated. Functional tests may be a scripted series of program inputs, possibly even an automated mechanism for controlling mouse movements. Unit tests may be separate functions within the code itself, or driver layer that links to the code without altering the code being tested.

In regression testing there is two major challenges are (1) finding objects of study (programs, releases of these programs, test suites, fault data) (2) selecting the appropriate approach to answer research questions

Conclusion

Regression testing is done in the maintenance phase of the software development life cycle to retest the software for the modifications it has undergone. Approximately 50% of the software cost is involved in the maintenance phase so researchers are working hard to come up with best results by developing new Regression Testing techniques so that the expenditure made in this phase can be reduced to some extent.

References

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

[2] 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.

[3] H. Leung and L. White, "Insights into regression testing," In Proceedings of the Conference on Software Maintenance, pages 60-69, Oct. 1989.

[4] Rothermel R., "Efficient Effective Regression Testing Using Safe Test Selection Techniques," Ph.D Thesis, Clemson University, May, 1996.

[5] Y. Chen, D. Rosenblum, and K. Vo. TestTube, "A system for selective regression testing," In Proceedings of the 16th International Conference on Software Engineering, pages 211-220, May 1994.

[6] K. Fischer, F. Raji, and A. Chruscicki, "A methodology for retesting modifed software," In Proceedings of the National Telecommunications Conference B-6-3, pages 1-6, Nov. 1981.

[7] R. Gupta, M. J. Harrold, and M. Soffa, "An approach to regression testing using slicing," In Proceedings of the Conference on Software Maintenance, pages 299-308, Nov. 1992.

[8] N.Mansour, and K. El-Faikh, "Simulating annealing and genetic algorithms for optimal regression testing," Journal of Software Maintenance, Vol. 11, pages 19-34, 1999.

[9] M.J.Harrold, R.Gupta, and M.L. Soffa," A methodology for controlling the size of the test suite, " ACM Transaction on Software Engineering and Methodology, pages 270-285, July 1993.

[10] H.Agrawal,J.R. Horgan, and E.W. , Krauser, "Incremental regression testing," In: Proc. Conference on Software Maintenance, pages 348-357,1993.