



Introduction of Software Maintenance Testing

Reetika Nagar, Sachin Kumar

School of Information and Communication Technology
Gautam Buddha University, India

Vipin Kumar

MCN Solution Pvt. Ltd.
Noida, India

Abstract— *Maintenance of software is a very crucial and important task but it is a very expensive process. Thus, Software Maintenance Testing is essential during software testing phase. Defects and errors found during testing process must undergo a re-test process so that flaws, that is, defects and errors can be eliminated easily. By doing so, test cases are absolutely needed to evolve and modify according to the changing requirements. In this paper, a brief introduction of several maintenance testing's such as confirmation testing and regression testing; and introduction of maintenance testing techniques has been given. These techniques are to be used in testing during software development to make testing process effective.*

Keywords— *Maintenance Testing, Confirmation Testing, Regression Testing, Regression Test Selection, Test Case Prioritization.*

I. INTRODUCTION

Software maintenance is an essential phase of software development life cycle. Software maintenance is becoming very expensive with the changing requirements and trends. During software maintenance following activities is performed such as enhancement of software functionalities, error correction, optimization, etc. Software functionalities need to be updated concurrently to avoid existing software being out-dated as compared to newly developed software with enhanced features. Thus, software need to be maintained, that is, changed, updated, deleted, extended over time during software evolution. Software maintenance activities may cause the software to work incorrectly and generate a huge number of defects in software if activities are not implemented properly. Therefore, verification and validation of software maintenance activities is need to be done to ensure their processing. This leads to generate software maintenance testing process.

Maintenance testing as defined in ISTQB glossary terms (standard glossary terms ver2.0) “*testing the changes to an operational system or the impact of a changed environment to an operational system*”. It has been shown that 80% of overall testing budget is for retesting the software and 50% of total software maintenance is consumed by retest alone [1]. Therefore, maintenance testing is a very important process during maintenance phase of software development life cycle.

There are two types of testing process to be performed during maintenance phase of software development life cycle that relates to changes and improvement in software during maintenance phase. The two types of maintenance testing are confirmation testing and regression testing. Confirmation testing also known as retesting is the process to ensure that the tests cases in which defects were found and test cases which failed in last execution are corrected and another test execution will undergo to pass after the defects and re-confirm that the failure and defects that has been fixed does not exist. It is important to ensure that confirmation testing should be performed in exactly the identical way in which initial testing was performed using the same data, inputs and environments conditions. If the confirmation testing passes over the defects, it fixes but does not guarantee that the defect has been corrected. It might insert defects somewhere else, hence regression testing is needed. Regression testing is implemented to ensure the defect does not spread to other functionalities of software. Regression testing proves that the entire software system is not affected during maintenance work. In this paper, a brief introduction of maintenance testing that is, confirmation testing and regression testing is given. This paper also discusses about various techniques of regression testing. The techniques of regression can further be implemented using different methods and techniques such as heuristics, meta-heuristics and others on different software systems and problems to make testing cost and time effective.

The paper is organized as follows. Section II gives a brief idea about regression testing. Section III discusses about different techniques of regression testing. Section IV discusses about the difference between regression testing and confirmation testing. Section V discusses conclusion and future scope.

II. REGRESSION TESTING

Regression testing is a type of software testing that attempt to detect new faults or bugs in software after changes have been made in existing functional and non-functional areas of software system . Regression testing is also a verification process that determines that previous features and functionality of software remains after a change is made in the software. Regression testing is defined [2] as the process of retesting and validating the modified parts of the software. Its main goal is to ensure that no new errors have been introduced into previously tested code or software system after changes have been made in the software. One of the main reasons to introduce regression testing is to determine whether

a change in one part of the software affects other parts of the software. Let P be a program [3], let P' be a modified version of P, and let 'T' be a test suite for P. Regression testing consists of reusing 'T' on P', and determining where the new test cases are needed to effectively test code or functionality added to or changed in producing P'. Regression testing is a necessary feature of the extreme programming software development method. Even in the corporate world, regression testing has been performed by a software quality assurance team.

Regression testing is used not only for testing the correctness of a program, but often also to trace the quality of its output [4]. For instance, in the design of a compiler, regression testing could trace the size of code, and the time it takes to compile and execute the entire test suite cases. Regression testing is a risk free process occurs at an optimal cost and in minimum time. The purpose behind regression testing is very simple and straight. It increases the productivity and efficiency of software products and quality assurance applications.

III. REGRESSION TESTING TECHNIQUES

Regression testing has been categorized to different testing techniques so that maintenance testing process can be performed effectively and easily with minimum time and cost requirements. Regression testing techniques are given below:

- a) Retest all
- b) Regression Test Selection
- c) Test Case Prioritization
- d) Hybrid Approach

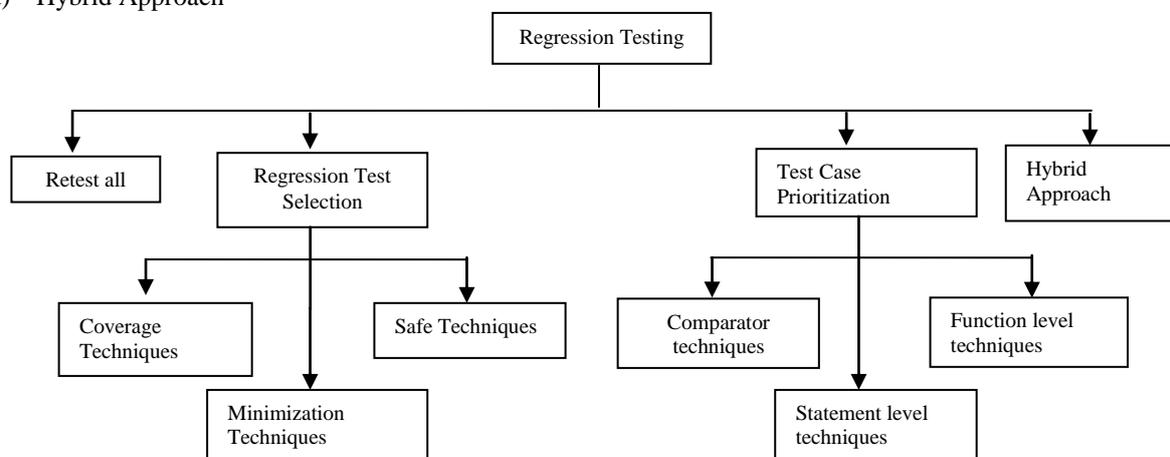


Fig. 1 Regression Testing Techniques

A. Retest All

Retest all method or technique is one of the traditional techniques for conducting regression testing. As the name indicates retest all is the techniques in which all the tests are runned and executed again in the existing test suite functional and non-functional areas. So the retest all technique [5] is very expensive and time consuming. As regression test suites are costly to execute in full as it require more time and budget other techniques like Regression Test Selection and Test Case Prioritization are required to be introduced to make regression testing cost and time effective.

B. Regression Test Selection

Due to expensive nature of “retest all” technique, Regression Test Selection (RTS) is performed. In this technique instead of rerunning and executing the whole test suite we select a part of test suite to rerun provided the cost of selecting a part of test suite is less than the cost of running the all tests that RTS allows us to omit. RTS divides the existing test suite into following test cases:

- a) Reusable test cases
- b) Retestable test cases
- c) 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 [6]:

- 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.

Regression test selection problem is about choosing a subset of test cases from the test suite. In regression test selection, test cases are selected because their execution is relevant to the changes between the previous and the current version of the software system. In regression test selection technique instead of rerunning and executing the whole test

suite only a selected part of test suite is to be executed again provided the cost of selecting a part of test suite is less than the cost of running the all tests that RTS allows us to omit. Regression test selection is used to reduce the time required to retest the modified program by selecting test cases.

Let us consider a program P with its modified program P' and its test suite T created to test P. When we modify P to P', we would like to execute modified portions of the source code or software system and the portions that are affected by the modifications to see the correctness and efficiency of modifications. But we neither have time nor resources to execute all test cases of T. Thus, our objective is to reduce the size of T to T' using some selection criteria, which may help us to execute tests on the modified portion of the source code and the portions affected by modifications.

Regression Test Selection Technique can be evaluated and compared on the basis of following features [7]:

- a) Inclusiveness: It is the measure of extent up to which a technique chooses the test cases which will cause the changed program to produce different output than the original program, resulting in exposure of faults due to modifications.
- b) Precision: It is the measure of ability of technique to prevent choosing test cases that will not make the changed program to produce different output than the original program.
- c) Efficiency: It measures the practicality (computational cost) of a technique.
- d) Generality: It is the measure of ability of a technique to handle complex modifications, realistic language constructs and realistic testing applications

C. Test Case Prioritization

Test Case Prioritization (TCP) technique of regression testing prioritize the test cases so as to increase a test suite's 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:

- a) General prioritization [8] which attempts to select an order of the test case that will be effective on average subsequent versions of software.
- b) Version Specific prioritization which is concerned with particular version of the software.

Test Case Prioritization can be classified further as given below:

- 1) *Comparator techniques*: These consist of random ordering and optimal ordering.
- 2) *Statement level techniques*: These techniques are also known as Fine Granularity. They consist of total statement coverage prioritization, additional statement coverage prioritization, total fault-exposing-potential (FEP) prioritization and additional FEP prioritization.
- 3) *Function level techniques*: These techniques are also known as Coarse Granularity. They consist of total function coverage prioritization, additional function coverage, total FEP prioritization, additional FEP prioritization, total fault index (FI) prioritization, additional Fault Index (FI) prioritization, total FI with FEP coverage prioritization and additional FI with FEP coverage prioritization.

Test Case Prioritization (TCP) technique of regression testing prioritize the test cases. Prioritization of test cases is generally done to reduce the cost of regression testing. Test cases are prioritized so that those which are more important, by some measure, are made to run earlier in the testing phase. There exists a large variety of prioritization techniques. Mostly used techniques are coverage-based prioritization techniques, that is, prioritization in terms of the number of statements, path coverage, branch coverage and fault coverage. Effective test case prioritization technique for regression testing is necessary to ensure optimum utility and no side effect in the software after modification [9]. Test case prioritization is rearranging the order of the test case based on certain constraints so that the most beneficial test cases may be executed first.

Rothermel et al. [10] define the test case prioritization problem as follows: Given: T, a test suite; PT, the set of permutations of T; f, a function from PT to the real numbers. Problem: Find $T' \in PT$ such that $(\forall T'') (T'' \in PT) (T'' \neq T') [f(T') \geq f(T'')]$. Here, PT represents the set of all possible prioritizations (orderings) of T and f is a function that, applied to any such ordering, yields an award value for that ordering.

D. Hybrid Approach

Hybrid Approach of regression testing as the name suggests is the hybrid of both Regression Test Selection and Test Case Prioritization. This approach contains features and properties of both test case selection and test case prioritization techniques. It is an efficient approach. There are number of researchers working on this approach and they have proposed many algorithms for it. For example,

- a) Test Selection Algorithm: proposed by Aggarwal et al. Implementation of algorithm [11]:
 - Input
 - Test Selection algorithm: Adjust module and Reduce module
 - Output.
- b) Hybrid technique proposed by Wong et al which combines minimization, modification and prioritization based selection using test history [12], etc.

IV. DIFFERENCE BETWEEN REGRESSION TESTING AND CONFIRMATION TESTING

Regression testing is usually performed after confirmation testing but both are quite different. In regression testing test cases are extracted from functional test cases to verify that new defects should not be introduced in the existing software & check whether original features and functionality is working as requirements or not. Once the regression test suite is created you can automate test cases using automation tool but same is not applicable for retesting or confirmation testing. Difference between regression testing and confirmation testing is given below [13, 14]:

TABLE 1
 DIFFERENCE BETWEEN REGRESSION TESTING AND CONFIRMATION TESTING

S. No.	Regression Testing	Confirmation Testing
1	Regression testing is a type of software testing that intends to ensure that changes like defect fixes or enhancements to the module or application have not affecting unchanged part	Confirmation Testing is done to make sure that the tests cases which failed in last execution are passing after the defects against those failures are fixed.
2	Regression testing is not carried out on specific defect fixes. It is planned as specific area or full regression testing.	Confirmation Testing is carried out based on the defect fixes
3	In regression testing, you can include the test cases which passed earlier. We can say that check the functionality which was working earlier.	In confirmation testing you can include the test cases which failed earlier. We can say that check the functionality which was failed in earlier build.
4	Regression test cases we use are derived from the functional specification, the user manuals, user tutorials, and defect reports in relation to corrected problems.	Test cases for confirmation testing cannot be prepared before start testing. In confirmation testing only re-execute the test cases failed in the prior execution.
5	Automation is the key for regression testing. Manual regression testing tends to get more expensive with each new release. Regression testing is right time to start automating test cases.	You cannot automate the test cases for confirmation testing.
6	Defect verification is not comes under regression testing.	Defect verification is comes under confirmation testing.
7	Based on the availability of resources the regression testing can be carried out parallel with confirmation testing.	Priority of confirmation testing over regression testing is higher, so it is carried out before regression testing.

V. CONCLUSIONS

In this paper we gave a brief introduction of different maintenance testing and their techniques. Maintenance testing such as confirmation testing and regression testing are very efficient and helps to reduce the overall time and cost of maintenance testing. As maintenance testing requires almost 80% of the total budget of testing it should be performed very carefully under expert's observation. Maintenance of software's is generally done to change the features and functioning of software with the changing requirements of users. Maintenance of software's avoids software from being out-dated. To ensure that maintenance process has been performed properly maintenance testing is necessary. Issues of future research may also include implementation and automation of the maintenance testing techniques. Researchers can also aim to implement these techniques using meta-heuristic approaches in future on different case studies. Regression testing technique helps to handle large and complex software systems easily and efficiently.

REFERENCES

- [1] Harrold, M.J., "Reduce, reuse, recycle, recover: Techniques for improved regression testing", *IEEE International Conference on Software Maintenance*, pp. 5-5. , Edmonton, AB, Canada (2009).
- [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*, vol. 13, no. 2, pp. 65-83, June 2003.
- [4] H. Leung and L. White, "Insights into regression testing," In *Proceedings of the Conference on Software Maintenance*, pages 60-69, Oct. 1989.

- [5] G. Duggal, B. Suri, "Understanding Regression Testing Techniques", *COIT*, 2008, India.
- [6] Rothermel R., "Efficient Effective Regression Testing Using Safe Test Selection Techniques", Ph.D. Thesis, Clemson University, May, 1996.
- [7] G. Rothermel, R.H. Untch, C. Chu and M.J. Harold, "Test Case Prioritization," *IEEE Transactions on Software Engineering*, vol. 27, no. 10, pp. 928-948, Oct., 2001.
- [8] Sahil Batra, Dr. Rahul Rishi, "Improving Quality Using Testing Strategies", *Journal of Global Research in Computer Science*, vol. 2, no. 6, June 2011.
- [9] Zheng Li, Mark Harman, and Robert M. Hierons, "Search algorithms for regression test case prioritization," *IEEE Trans. On Software Engineering*, vol. 33, no. 4, April 2007.
- [10] K. K. Aggrawal, Yogesh Singh, A. Kaur, "Code coverage based technique for prioritizing test cases for regression testing," *ACM SIGSOFT Software Engineering Notes*, vol. 29, no.5 September 2004.
- [11] W. E.Wong, J. R. Horgan, S. London and H.Agrawal, "A study of effective regression testing in practice," In *Proceedings of the 8th IEEE International Symposium on Software Reliability Engineering (ISSRE' 97)*, pages 264-274, November 1997.
- [12] *Software Engineering: A Practitioner's Approach*, 7/edition by Roger S. Pressman.
- [13] <http://www.softwaretestingclass.com/difference-between-regression-testing-vs-retesting/>.