

# A Review on Regression Testing techniques

<sup>1</sup>Anjali Choudhary, Tarun Dalal<sup>2</sup>

<sup>1</sup>CBS group of Institution, Maharish Dayanand University,  
Jhajjar, Haryana, India

<sup>2</sup>CBS group of Institution, Maharish Dayanand University,  
Jhajjar, Haryana, India

## Abstract

Software development is a major process in Software Development life cycle. Regression Testing is a process of retesting the modified parts of the software and ensuring that no new bug is introduced into previously working functionality. It is crucial and often important software maintenance activity. This paper will represent a review on various Regression testing techniques by various researches in details.

**Keywords:** Regression Testing, Regression Testing techniques, test cases, prioritization.

## 1. INTRODUCTION TO REGRESSION TESTING

Software maintenance activity includes enhancement, bug fixing, error correction, optimization, deletion of obsolete functionality. These modifications in software can cause the software to work in inappropriate manner and may also affect other part of the software. Regression Testing is used to revalidate the changes made in the software. Regression testing an expensive yet time consuming process to ensure that no new error is left in the previously tested code. In next section, we have shown the various types of Regression test techniques and classification of these techniques. After which moving into the details of test case selection and prioritizing, discussing algorithms. In Section 3, we have discussed approaches which may be used to compare various Regression Testing techniques.

## 2. REGRESSION TESTING

Regression Testing is defined as the [1] “process of retesting the modified part of the software and ensuring that no new error or bug is introduced into previously tested code.”

[2] Let P is the original software product, P' is the modified software product and T is the set test case to test P. A typical Regression Testing on modified software proceeds as follow”

- Select  $T' \subseteq T$ , a set of test case to execute on the modified software product P'.
- Test P' with T', to verify modified software product's correctness with respect to T'.
- If necessary, create T'', a set of new test cases to test P'.
- Test P' with new test case T'', to validate P' with respect to T''.

- Create T''', a new test suite and test history for P', from T, T', T''.

There are various Regression Testing techniques:

- Retest all;
- Regression test selection;
- Regression test prioritization;
- Hybrid approach. See Fig 1.

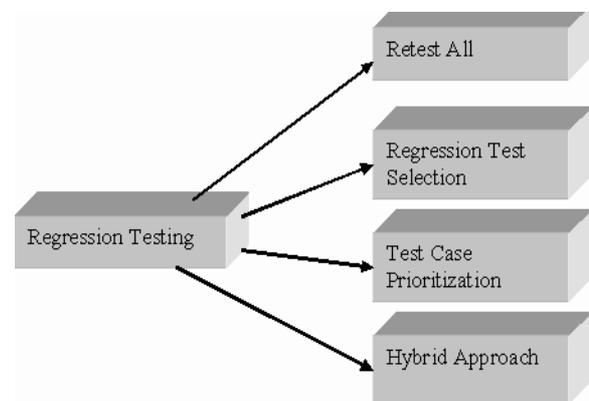


Figure1 Regression Testing techniques

## 3. REGRESSION TESTING TECHNIQUES

### 3.1 Retest All

Retest all is a method for Regression Testing in which all the test cases in the test suite are re-run. So retest all test cases is expensive as compared to other testing techniques which will be discussed further. This techniques is time and cost consuming.

### 3.2 Regression test selection

As Retest All technique is expensive and time consuming, regression test selection (RTS) is performed. Instead of running complete test suite, we select a part of test suite which will analyses if rerunning a part of test suite is less costly than rerunning the complete test suite. RTS allow us to omit the non usable test cases from the test suite and will only allow rerunning the test cases which will mostly effect the application. RTS divided the test suite into 1) P Reusable test cases; 2) Re-testable test cases and 3) Obsolete test cases.

RTS techniques are broadly divided into 3 categories [1]:

1. Code Coverage techniques: They usually prefer the test coverage criteria into account. They usually focus on parts that are modified and select the test case which will focus

on that part.

2. Minimization technique: They are similar to code coverage technique but they select test suite with minimum test cases.

3. Safe technique: They are completely different from coverage technique; instead they select those test cases that produce different output with a modified program as compare to original version.

Rothermel [3] has identified the various categories in regression test selection technique can be evaluated and compared. These categories are

- Inclusiveness;
- Precision;
- Efficiency;
- Generality.

a)Inclusiveness: is the way to measure of extent up to which a technique chooses test cases which will cause the change to program to produce different output than the original program

b)Precision: is the way to measure the ability of technique to prevent choosing the test case that will not make change program to produce different output than the original program.

c)Efficiency: measure the practicality of a technique.

d)Generality: is the way to measure the ability of a technique to handle the complex modification, realistic language construct and realistic testing application.

### 3.3 Test case Prioritization

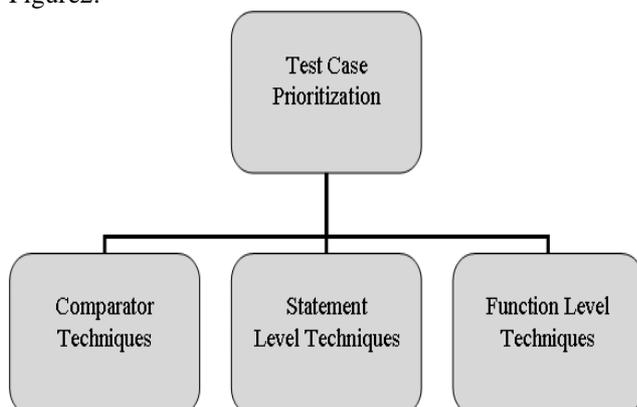
This technique of testing will prioritize the test cases in order to increase a test suites rate of fault detection.

This is of two types [4]:

- General Prioritization and
- Version specific prioritization.

In General Prioritization, it attempts to select an order of test case which will effect on average subsequent version of application and in Version specific Prioritization, it focuses on particular version of the application.

There are 18 different test case prioritization techniques [5] which are categorized into 3 groups as shown in Figure2.



**Figure2.** Classification of test case prioritization

#### 3.2.1 Comparator Technique

- P1: Random Ordering: Test cases are randomly prioritized.
- P2: Optimal Ordering: Test cases are poertized to optimize the fault detection rate.
- P3: Total Statement Coverage prioritization: Test cases are prioritized in terms of total number of statement by sorting them in order on coverage achieved.
- P4: Additional statement coverage prioritization: It is similar to total coverage prioritization but differ in one sense, that it depends on the feedback about coverage attained to focus on areas which are not yet covered.
- P5: Total FEP prioritization: Prioritization is done on the probability of the exposure of faults by test case. To approximate Fault Exposure Potential (FEP), mutation analysis is used.
- P6: Additional FEP Prioritization: Total FEP is extended to additional FEP when the total statement coverage prioritization is extended to additional statement coverage prioritization.

#### 3.2.2 Function level Techniques (Coarse Granularity)

- P7: Total function coverage prioritization: It is same as total statement coverage but in-place of using statements it used functions. It has got coarse granularity so the process of collecting statement level trace in the total statement coverage.
- P8: Additional Function coverage prioritization: This is similar to Additional statement coverage but only difference is that it is considering function level coverage.
- P9: Total FEP prioritization: It is analogous to total FEP prioritization but differ in one aspect that it is using functions.
- P10: Additional FEP Prioritization: Similar to Additional FEP with only difference that it uses functions instead of statements.
- P11: Total Fault index (FI) Prioritization: Fault proneness is measurable software attribute which used for this technique.
- P12: Additional fault index (FI) prioritization: Total fault index (FI) prioritization is extension to Additional Fault index (FI) prioritization.
- P13: Total FI with FEP coverage prioritization: It is the combination of Total FI and FEP coverage prioritization to achieve good rate of fault detection.
- P14: Additional FI with FEP coverage prioritization: Total FI with FEP coverage prioritization is extended to Additional FI with FEP coverage prioritization.
- P15: Total Diff Prioritization: It is same as Total Fault Index prioritization with difference that Total FI prioritization requires collection of metrics and total difference prioritization requires calculation of syntactic difference between program and modified program.
- P16: Additional Diff Prioritization: Additional Diff prioritization is extension of Total Diff Prioritization.

- P17: Total diff with FEP prioritization: Same as Total FI with FEP coverage prioritization but it is dependent on data derived from diff.
- P18: Additional Diff with FEP Prioritization: This is Total diff with FEP prioritization extension.

### 3.4 Hybrid Approach

Hybrid Approach combines both Regression test selection and test case prioritization. Hybrid approach is based on the selection and prioritization of test cases for inter-procedural program. It is a version specific technique which takes into account the variable usage in old as well as modified code. The technique requires the test cases in original test suite not only have test case identification, expected input and output but also the variable that will be checked by the test suite and module.

## 4. REGRESSION TESTING AND CONFIGURATION MANAGEMENT

Configuration Management during Regression Testing is vital in Agile Environment where code is continuously changed. To ensure effective Regression Testing, we need to observe the following:

- 1) No Changes should be allowed to done during Regression Testing phase.
- 2) Database must be kept isolated which is used for Regression Testing

## 5. DIFFERENCE BETWEEN REGRESSION TESTING AND RE-TESTING

Retesting refers to retesting the functionality or bug again to validate that the code is fixed. If it is not fixed, then defect is reopened and if it is fixed then it is closed. Basically Regression Testing is done to make sure that bug is resolved and failed functionality is working or not. Retesting is planned testing and Regression Testing is a generic testing.

Regression Testing refers to test software application when it undergoes any code change that are made by the developer, to ensure that no new bug or error is introduced in the application due to which other working functionality are not affected. Regression Testing has low priority as compared to Re-testing.

## 6. APPROACHES AND CHALLENGES

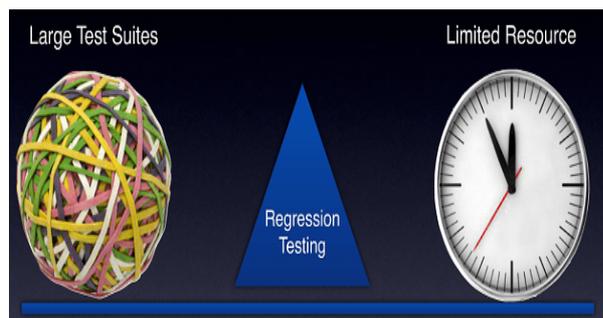


Figure 3. Regression Testing Challenges

Regression Testing have some problems and challenges which are as following:

- With successive Regression test, Test suite becomes fairly large.
- Due to time and cost constraints, it is sometime becomes impossible to execute the entire test case.
- Minimizing the test case remains a challenge.
- To determine the frequency of Regression test i.e. after every modification or every bug fixing

Two approaches [5] are there so that all the techniques discussed above can be made comparable:

- 1) Controlled Experiments and
- 2) Case Studies.

- Controlled Experiments: They are performed on objects that are drawn the code and are further created and modified in controlled environment. Its advantage is that independent variable can be changed to find their effect on dependent variable. Disadvantage of the same can be that it is threat to external validity shaped by manufacturing of errors, test cases and manipulations.
- Case Studies are performed on objects taken “from field”, that have error data and existing test suites. Advantage is that this approach reduces cost and disadvantage is that some factors are not controlled.

Therefore, Case studies and Controlled Experiments have some advantages and disadvantages.

## 7. CONCLUSION

Regression Testing is performed in maintenance phase of Software development Cycle to retest the enhancements made in the software application. During Regression Testing, approximately 50% of the software cost is involved in the maintenance period so researchers are working hard to reduce the cost to some extent. There are several Prioritization techniques that are applied on various examples to get the effective results. In future we can work on the mentioned prioritization techniques to get the effective and optimal results. Furthermore ACO which is a promising technique for solving test cases selection and prioritization. So, researchers can focus on same for reducing time for finding error and faults.

## References

- [1] K.K Aggarwal & Yogesh Singh, “Software Engineering Programs Documentation, Operating Procedure,” New age International Publishers Revised Edition - 2005
- [2] J.M Kim and A. Porter, “A history based test prioritization technique for Regression testing in resource constraint environments,” ACM proceeding of the 24<sup>th</sup> International Conference on Software Engineering, pp 119-129, 2002
- [3] Sebastian Elbaum, Member, IEEE, Alexy G. Melishvesky, Student Member, IEEE and Gregg Rothermel, Member, IEEE, “Test case Prioritization:

A family of empirical studies,” vol. 28, No.2, Pages 159-182, Feb 2002.

- [4] G. Rothermel, R.H Untch, C. Chu and M.J Harrold, “Prioritizing test cases for Regression testing.” IEEE Trans Software Engg., Vol 27, no. 10, pages 929-948, Oct 2001
- [5] Rothermel R, “Efficient Effective Regression Testing using Safe Test Selection Techniques,” Ph.D Thesis, Clemson University, May 1996.

## **AUTHORS**

**Anjali Choudhary** received the B.tech degree in Information and Technology from Rajasthan Technical University, Rajasthan in 2008. She is now pursuing M.tech in Computer Science Engineering CBS group of institution from Maharishi Dayanand University, Haryana.

**Tarun Dalal** is Asst Professor of Computer Science Engineering department in CBS Group of Institution affiliated to Maharishi Dayanand University, Haryana.