Volume 6, Issue 5, May 2016 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

A Review Paper on Test Case Selection in Regression Testing

Ritu, Sukhdip Singh
Department of Computer Science & Engineering, DCRUST, Murthal, Sonepat,
Haryana, India

Abstract- Regression testing is selective retesting of the system; executed with an objective to ensure the bug fixes work and those bug fixes have not caused any un-intended effects in the system.The approach mainly based on the multiple criteria test cases selection. The method mainly aims to maximize the coverage size by executing the test cases effectively. The selected test cases are then prioritized based on the priority which depends on the test cases to achieve the desired results.

Keywords - Regression Testing, bug, prioritization, multiple criteria test selection.

I. INTRODUCTION

Regression testing is the process to test and validate the modified module of software so that the unchanged module or part will remain unaffected. It is part of software maintenance activity and it accounts one-third of the overall cost of software maintenance. Regression testing is the retesting of the modified parts of the software and test case selection helps to retest the selective test cases from the test suite to reduce time, cost and resources. Retesting of software is done frequently during the software development lifecycle and in particular in regression testing. In regression testing software grows and evolves that create new test cases and added to a test suite to exercise the latest changes in the software. Due to many versions of the development of the projects, the possibility of redundant test cases in test suite is more. The redundant test case may in respect to the testing requirements for which they were generated. Due to limitation of time and resource for retesting the software every time before a new version is released, it is really important to search for techniques that ensure manageable test suites size by removing redundant test cases without hampering the performance of the software. A number to test case selection techniques are used to minimize the test cases. These techniques have various procedures to select test cases from the original test suite. Some of these techniques are code based regression testing technique, model based, module based and many more, according to their defined approach. Test case selection in regression testing helps to make the retesting cost effective and it also saves a lot of time, which overall reduces the cost of software.

There are some most effective tips which should be followed while selecting test case selection for regression testing:

1. Include the test cases which have frequent defects.
2. Include the test cases which verify core features of the product.
3. Include the test cases for functionalities which have undergone more and recent changes.
4. Include all the integration test cases.
5. Include all the complex test cases.
6. Prioritize the test cases for regression testing.
7. Categorize the selected test cases as reusable or obsolete test case.
8. Choose the test cases on “Case to Case” basis i.e. the criticality and impact of bug fixes.
9. Change the regression test cases whenever required.

II. PREVIOUS WORK ON REGRESSION TESTING

A number of researchers have developed various techniques to select test cases from the original test suite to save time, cost and resources. At different levels, regression testing has been done to the updated software and test cases have been retested to know the negative impact of modification. Also, some techniques are combined together to develop a new technique. [1], proposed a technique based on the model of software. They define the relationship between the model elements and test cases. This technique defines the model and traceability infrastructure. It uses the UML approach for sequence diagrams. The main limitation of this technique is that we cannot use it at higher abstraction level. It also has no connection with the coding part of software. To overcome this limitation a new technique is developed where the modification is to be done in the coding section and the modification is done in semantic form. Also various techniques have been developed based on graphs. [3], proposed a new technique in which object dependency graphs are used to identify the change. In object dependency graph we can add and delete the modifications which are to be done in the updated version. They also proposed an algorithm to select the test case from the original test suite that should be retested. It detects the changes at the source code level. The main limitation is that they do not have any tool to apply this technique. [4], proposed a technique which uses differential control flow graphs (DCFG) for modification and then to select test cases for regression testing. This technique uses an update complexity no. for each module under test i.e. each

module has a UCN on the basis of which test cases are being selected. A module with higher UCN is being selected for test case based on DCFG. It also defines the relationship between module and existing test cases. It works in three steps from defining relationship between module and test cases to selecting test cases from original test suite. Limitation of this technique is that it cannot calculate the structural complexity of module. [5], developed a technique based on dynamic behavior. This technique is a combination of both code based technique and model based technique. It captures the dynamic behavior of software from UML diagram. With the help of UML diagram it identifies the changes in software. A lot of other techniques are there, like state machine based technique [9], web based technique [10], hierarchical slicing technique [2] etc. In the field of regression test case selection techniques, a lot of work has been done and published but we are still trying to make it more and more cost effective and time saving.

III. TECHNIQUES AND ALGORITHMS

1. Model based technique:

Here, the negative impact of the quality of software is identified by using selective model based regression testing. It drives a relationship between model elements and test cases and works in various phases by creating model and traceability infrastructure. Its framework uses the UML approach and sequence diagrams for modeling .Model based technique works in two phases, transforms the sequence diagram into model based control flow graphs and then the graphs into test generation hierarchy. Further it uses the graph to create infrastructure and differencing model. Future work of model based approach is to use it at higher abstraction levels [1].

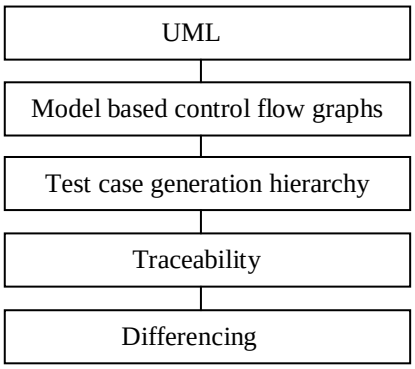

Fig 1 Model based technique framework

2. Technique using object dependency graph:

This technique is used to identify the change and also the test cases which should be retested to make the software cost-effective. It uses object dependency graph to show the modification which represents the structure and relation of object in object-oriented programming using vertex and edges. Framework of this technique includes the change object identifier, which add or delete objects and also changes the functionalities of objects.

ALGORITHM:

1. Select T' from T, to execute on P'.
2. Test P' with T', establish the correctness of P'.
3. If necessary, check T'' for P'.
4. Test P' with T''.
5. Create T''', a new test suite for P'.

It detect new changes in source code level and also identify the test cases which need to be retested to consume time and resources. Future work is to develop a tool in this framework to reduce time and consumption in regression testing [3].

3. Technique using differential control flow graphs.

Here, test cases are selected on the basis of an update complexity no. for each module of software under test which is further based on a differential control flow graph. This whole relationship between existing test cases and module is defined by structure chart. An UCN is decided on the basis of graph-based metrics for a DCFG, which consider a no. of arcs and branches. The proposed technique mainly consists of three steps:

STEP 1: Define relationship between modules and existing test cases.

STEP 2: Calculate the UCN'S based on DCFG's.

STEP 3: Selection of test cases from existing test cases on the basis of UCN's.

Module with larger UCN's are selected for test cases based on DCFG. This technique focus on the updated parts and existing parts those are adjacent to the updates parts. Future work is to construct the metrics that can calculate the structural complexity of DCFG's by identifying the nodes and also to develop a tool to implement this technique [4].

4. Technique based on Dynamic Behavior

This technique combines the code based technique and model based technique together to generate a safe regression test selection technique. Technique: Teat cases are selected based on changes made to software specification represented in UML diagram and code that represented in any programming language. It consists of three functions:

1. Capture dynamic behavior.
2. Identify changes.
3. Select regression test suite.

It captures the dynamic behavior of the system from UML class diagram and sequence diagram and then it identifies the syntactic and semantic changes to code by comparing the original program and modified program. In future, it would focus on investigating the technique that automatically identify major changes made to code and general test cases that validate these changes [5].

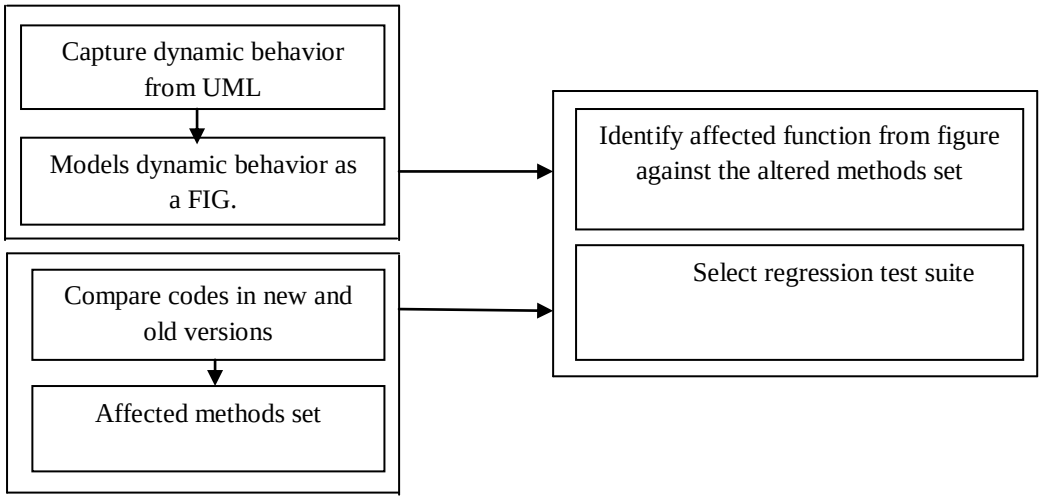

Fig 2 Framework explains dynamic behavior technique [5].

5. Specification based approach

In this Technique, a concept of behavioral slicing is being proposed to structure activity diagram. Also, a prototype tool has been developed which conduct real-world case studies. User is requires to provide structure to the activity which uses the concept of behavioral diagram. This approach is based on the changes made to the software specification represented as UCAD's. It mainly consists of three steps:

1. Structure the activity diagram of the system using the concept of behavior slicing using units of behavior.
2. Identify the modified pars of the activity diagram. The corresponding UCAD's reflects the modified specification.
3. Selection of the test suite that need to be re-executed.

This approach helps to identify the change at user requirement level and to select the regression test cases from the modifications made to the activity diagram. It also provides the concept of node version number to structure the activity diagram [2].

6. Hierarchical Slicing technique

This approach selects the test cases from higher level of program to lower level of program and is being used by object oriented programs. The main concept is to apply hierarchy slice into regression testing to calculate the slices to input to get program output at different levels i.e. from package level to statement level. The model of hierarchy slicing technique consists of three parts:

- Hierarchical slicing criteria
- Hierarchical Dependence Graph
- Stepwise slicing algorithm

The primary work of this technique is to obtain hierarchical coverage information for every test case and then to compute hierarchical slice set for modifications step by step. Finally, calculate their intersection of them at different levels. If the intersection set of test case I not empty, then it will be selected. Precision of test case selection in slicing technique is also done. Study of this approach selects the test cases step by step from higher to lower level and also improves the precision of regression test case selection. Future work is to refine the approach at different levels which includes selection and prioritization [6].

IV. CONCLUSION AND FUTURE WORK

The goal of our work is to select and prioritize test cases for performing regression testing activity. In this work, we implement and validate a regression test selection and prioritization technique. The proposed technique increases confidence in the correctness of the modified program. The test cases selected using the proposed technique will identify and locate errors in the modified program. The proposed technique will help in preserving the quality and reliability of the software. Test cases selected by the proposed technique will ensure the software's continued operation. The results show that the proposed regression selection and prioritization technique will help in reducing the test cases by a

significant number. Therefore, the software developers and testers can use this technique in practice and this technique can reduce the cost of regression testing significantly. In future we will analyze the proposed algorithm on large programs.

REFERENCES

[1] Leila Naslavsky, Hadar Ziv, Debra J. Richardson," A Model-Based Regression Test Selection Technique" IEEE 2009

[2] Chuanqi Tao, Bixin Li, Xiaobing Sun, Chongfeng Zhang**."** An Approach For Regression Test Selection Based On Hierarchical Slicing Technique: IEEE 2010.

[3] Adipat Larprattanakul, Taratip Suwannasart," An Approach For Test Case Selection Using Object Dependency Graph, IEEE 2013.

[4] Shun Akimoto, Rihito Yaegashi, Tomohiko Takagi,"Test Case Selection Technique For Regression Testing Using Differential Control Flow Graphs" IEEE 2015.

[5] Walid Said Abd El-hamid, Sherif Said El-etriby, Mohiy Mohamed Hadhoud," Regression Test Case Selection Technique Based On Dynamic Behavior" IEEE 2010.

[6] Ravi Prakash Gorthi, Anjaneyulu Pasala, Kailash KP Chanduka and Benny Leong,"Specification Based Approach To Select Regression Test Suite To Validate Changed Software" IEEE 2008.

[7] Sheng Huang, Zhong Jie Li, Ying Liu, Jun Zhu,"Regression Testins As A Service" IEEE 2011.

[8] Gregg Rothermel, Mary Jean Harrold," A Framework For Evaluating Regression Test Selection Techniques" IEEE 1994.

[9] Qurat-ul-ann Farooq, Muhammad Zohaib Z. Iqbal, Zafar I Malik, Aamer Nadeem," An Approach For Selective State Machine Based Regression Testing" ACM 2007.

[10] Abbas Tarhini, Zahi Ismail, Nashat Mansour,"Regression Testing Web Applications, IEEE 2008.

