Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 3, Issue. 7, July 2014, pg.607 – 614

RESEARCH ARTICLE

A Test Case Prioritization Approach in Regression Testing

Isha Sharma[1], Jasleen Kaur[2], Manisha Sahni[3]

[1]Deparment of CSE, GNI Mullana, Kurukshetra University, India
[2]Deparment of CSE, GNI Mullana, Kurukshetra University, India
[3]Deparment of CSE, GNI Mullana, Kurukshetra University, India

[1] ishasharma211@gmail.com; [2] jkaur.1987@gmail.com; [3] manishasahni06@gmail.com

Abstract-- Testing is a process of analyzing behavior of the product to detect difference between existing products and required conditions. It is a process to identify completeness, correctness and effectiveness of computer software. In the existing work test cases are prioritized on the basis of timing and number of faults find out. The existing technique gives higher priority to least important function and low priority to most important function. So to avoid this problem a new technique will be proposed which gives priority to the importance of the function not test cases.

Keywords-- Regression testing, Test case prioritization, Test cases

I. INTRODUCTION

Software engineering is about building, evolving and maintaining software systems. Software engineering is a set of problem solving skills, techniques, technology and methods applied upon a variety of domains to evolve and create useful systems that solve many problems like practical problems. Software engineering is dividing into number of categories and phases. It is an internal part of software development and closely related to software quality. The main aim of software testing is to fulfill user's requirements and make the system error free. So software testing is mainly to find outs the error or bugs to improve the quality of the system [12,13] .Testing is a process of analyzing behavior of the product to detect difference between existing products and required conditions. It is a process to identify completeness, correctness and effectiveness of computer software. It checks whether the expected results match with the actual results**.** It is the last phase of the product before deliver to the customer. Software testing is an important part which tells whether the product is efficient and error-free, work properly and according to the requirements of the customer. There are different types of testing types available. White-box testing is also known as Glass testing and structural testing. In white-box testing code is visible. It has knowledge of the internal mechanism of the components. Black-box testing is also known as functional testing. This testing ignores the internal mechanism of the system. It is a testing which is based upon the output and having no knowledge of internal code [9]. Unit testing is a type of white-box testing. It is a testing for low-level design code. It is done within a class and starts from a individual module [8]. Integration testing is done when two or more modules are combined together into a larger module. [2] System testing verifies the functionality of the module after integration. It is done at the interfaces of the both the structure and component module [9]. This type of testing is done in distributed or client/server modules. System Testing is also known as end-to-end testing. It tests the complete application of environments. Load Testing is also comes under performance testing. It is done to determine whether the system is able to handle given no. of users or not. It is done only to check the performance of the system is it working well or not [4]. Regression Testing fixes bug often and break other things the developer isn't concentrating on. Sometimes bug fixes don't fix the bug. The checking software still runs after making a change in the infrastructure. It also discovers faulty localization. The errors in the build process are used to conforming to standards or regulators.

II. LITERATURE REVIEW

W. Eric Wong et.al (2007) purposed [13] of regression testing is to ensure that changes made to software, such as adding new features or modifying existing features, have not adversely affected features of the software that should not change. Regression testing is usually performed by running some, or all, of the test cases created to test modifications in previous versions of the software. This technique seeks to identify a representative subset of all test cases that may result in different output behavior on the new software version. We report our experience with a tool called ATAC which implements this technique. Corina S. Pasareanu et.al (2009) have [2] introduced the new research trends in symbolic execution, how to test generation and program analysis. First describe a procedure that handles complex programming constructs such as input recursive, arrays, as well as multithreading, data structures. Also describe latest hybrid techniques to overcome some of the inherent limitations of symbolic execution, such as handling native code or availability of decision procedures for the application domain that combine concrete and symbolic execution. Finally, survey of predictive testing, invariant inference also mentioned. Some traditional application such as test generation and analysis is also discussed. Parallelizing and extending the abstraction and composition is also presented in the paper. Some new heuristics techniques that handle the traditional approaches and give beneficial results for the future are also discussed. Praveen Ranjan Srivastava et.al (2011) has [3] discussed about the It consists of estimating testing effort, selecting appropriate test team, designing test cases, executing the software with those test cases and examining the results produced by those executions. It indicates cost of software development is committed to testing, with the percentage for testing critical software being even higher.

This paper makes an attempt using fuzzy logic to estimate reliable software testing effort. In this paper triangular membership functions are chosen with monotonic constraints. Swarnendu Biswas and Rajib Mall (2011) introduced [4] about regression testing is an important and expensive activity that is undertaken every time a program is modified to ensure that the modifications do not introduce new bugs into previously validated code. An important research problem, in this context, is the selection of a relevant subset of test cases from the initial test suite that would minimize both the regression testing time and effort without sacrificing the thoroughness of regression testing. Researchers have proposed a number of regression test selection techniques for different programming paradigms such as procedural, object-oriented, component-based, database, aspect, and web applications. In this paper, we review the important regression test selection techniques proposed for various categories of programs and identify the emerging trends. Gregory M. Kapfhammer (2004) discussed [5] about the different types of software testing and their sub categories. The way to generate test cases, how to execute them also explained in this paper. The execution process of test cases, how to prioritize them using regressing testing and various types of related testing techniques are mentioned in this paper. How the testing is done for the graphical user interface is mentioned in this paper. On the bases of the existing techniques new techniques are proposed.

III. REFRESSION TESTING

Regression testing is a testing that refers to that section of the test cycle in which programs are tested to make sure that changes do not affect features that are not supposed to be affected. The process of verifying the modified software in the maintenance phase is known as Regression testing. Time and budget constraints are its major disadvantage due to complex process [10]. Regression testing is the re-execution of a number of subset of test that has previously been conducted. In regression testing as integration testing proceeds, number of regression tests increases and it is not practical and ineffective to re execute every test for each program function if once change occurs [11]. It is an expensive testing process used to detect regression faults. Research has shown that at least 50% of the total software cost is comprised of testing activities. Companies are often faced with lack of time and resources, which limits their ability to effectively complete testing efforts. Prioritization of test cases in the order of execution in a test suite can be beneficial. Test case prioritization (TCP) techniques organize the test cases in a test suite, allowing for increase in the effectiveness of testing.

There are two types of regression testing:-

1. Corrective Regression Testing
2. Progressive Regression testing

1. Corrective regression testing: It is triggered by corrections made to the previous version.
2. **Progressive regression testing:** It is triggered by new features added to the previous version.

The regression testing is applied directly to the change of the code. We can concern regression testing during development and in the field after the system has been upgraded or maintained in some other way. Good regression tests give us assurance that we can change the object while maintaining its behavior.

IV. APFD (AVERAGE PERCENTAGE OF FAULT DETECTION) METRIC

To analyze the effectiveness of prioritization techniques APFD [1] metric is used. To analyze the fault detection rate of test suite we use this metric. In this metric we consider the weighted average of no. of faults contained in the program. APFD uses the following notations:-

$$APFD = 1 - \frac{(TF1+TF2+\ldots\ldots\ldots+TFm)}{nm} + \frac{1}{2n} \tag{1}$$

where T -> Test Suite
m ->number of faults detected during the execution of test suite
n -> total no of test cases
TFi ->position of the initial test in test suite T that detects fault i

This formula represents that we can calculate APFD only when we have prior knowledge of faults contained in the program.

V. PROBLEM DOMAIN

Research in software testing has experienced a significant growth in recent years. Regression testing is used to verify the modified version of the software. There are many techniques which are developed to define the test cases. Test case prioritization is of different kind i.e. faults finding based or timing based depends upon the requirement. Test cases are prioritized to give importance to the most important function so that will execute first. Test cases can be prioritizing on the basis of faults, time and its requirements. In this work the important function’s test case will be execute first then others function’s test cases.

VI. PROPOSED METHODOLOGY

Regression testing is used to verify the modified version of the software. A test case has an input, an action performs and an expected result as an output. The main goal of test cases is to find out errors. Test case prioritization is of different kind i.e. faults finding based or timing based depends upon the requirement. Test cases are prioritized to give importance to the most important function so that will execute first. Test cases can be prioritizing on the basis of faults, time and its requirements. Test case prioritization schedules the test cases in a regression test suite with a view to maximizing certain to find faults which help reduce the time and cost required to maintain service-oriented business applications. As we discussed earlier test cases are prioritize on the basis of time and faults. But sometimes a particular test case gives 1st priority to that function which is least important on the basis of either faults finding or timing constraints and gives last priority to that which is most important. The existing test cases prioritization has some problems. Test case prioritization is done either on the basis of fault finding or timing. Suppose test cases are prioritized according to faults finding. The test cases which will find maximum faults will be execute first. But this process has some shortcomings. Sometimes test cases gives highest priority to that function which is least important and that function containing test case executed first and most important test case execute at the end. So to overcome this problem a new algorithm will be implement which will prioritize according to functions importance.

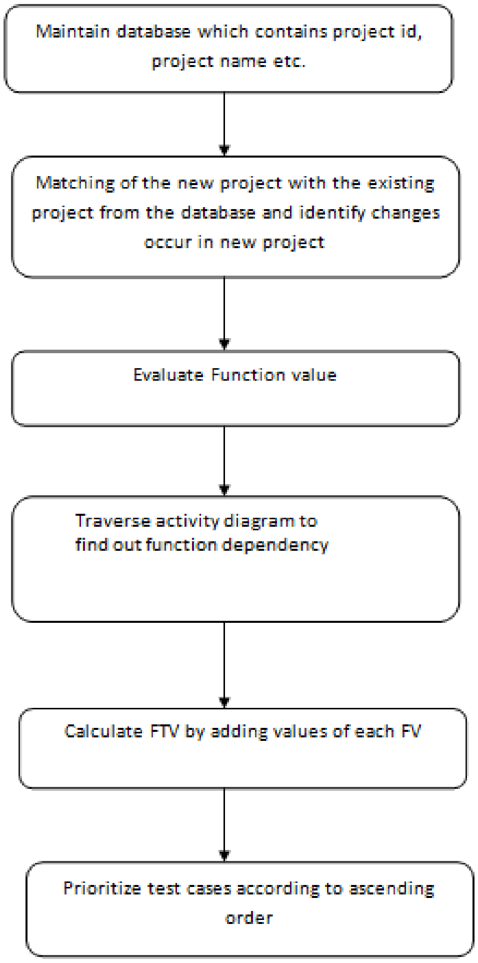
Maintain database which contains project id, project name etc.
Matching of the new project with the existing project from the database and identify changes occur in new project
Evaluate Function value
Traverse activity diagram to find out function dependency
Calculate FTV by adding values of each FV
Prioritize test cases according to ascending order

Fig.1: Flow chart of proposed technique

Algorithm: Steps of proposed algorithm are as follow:

1. Maintain a database which contains all the related information of the project i.e. project id, no. of changes made and affected functions due to changes.

2. Match new projects with database and find out that functions which are affected due to changes in the functionality.

3. Evaluate Function Values (FV) of each function according to formula of Function value.

$$\text{Function Value} = \frac{\text{number of times function encounter}}{\text{Total no of functions being affected}} \quad (2)$$

4. Traverse activity diagram to find out function dependency.

5. Find out the Function Test value (FTV) by adding values of each FV.

6. Prioritize the test cases according to the ascending order of each test case

VII. RESULTS

Test Cases	FTV
T2	3.46
T3	3.8
T4	6.56
T1	7.07

Table 1: Shows test cases according to the increasing order of FTV

Here the performance of proposed system can be evaluated by using the APFD metric.

Test Cases Fault	T1	T2	T3	T4
F1	X		X	
F2		X	X	
F3				X
F4			X	X
F5		X		

Table 2: Faults detected by the test suites

Number of test cases is 4,i.e,{T1,T2,T3,T4},and the number of faults occur during the regression testing is 5,i.e,{F1,F2,F3,F4,F5}.The prioritized test suites with test sequence {T2,T3,T4,T1},then APFD metric after prioritization is :-

$$\text{APFD}= 1-\frac{(3+3+1+1+1)}{4*5} + \frac{1}{2*4} \quad = 0.675$$

The APFD metric before prioritization is:-

$$\text{APFD}= 1-\frac{(1+1+1+4+5)}{4*5} + \frac{1}{2*4} \quad = 0.525$$

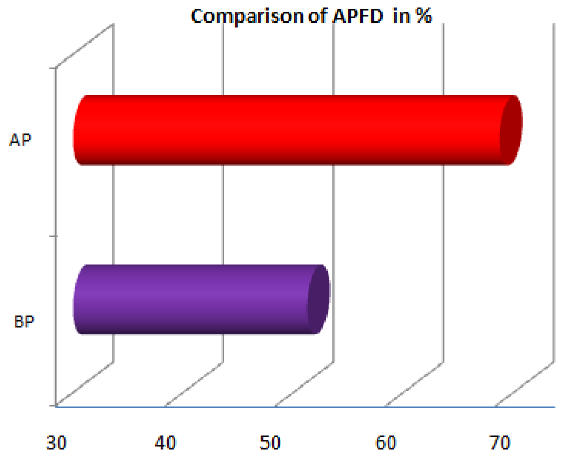

Fig.2: APFD metric for proposed system

VIII. APPLICATION DOMAIN

Regression Testing increases our chances of detecting bugs caused by changes to a software and application- either enhancements or defect fixes. Also keep one thing in mind that we also don't give any sort of guarantee that for performing regression testing here are never any sort of side effects also. Regression testing also detects undesirable side caused always by changing the operating environment. The set about regression test is much useful for a new way about doing integration testing. This new mode is quite faster and little confusing than the old way about doing integration testing- but you always need a some sort of set about regression test to do it.

IX. CONCLUSIONS AND FUTURE WORK

The process of verifying the modified software in the maintenance phase is known as Regression testing. Time and budget constraints are its major disadvantage due to complex process. To remove this disadvantage a new technique will propose which depends upon the importance of the functionality. In the existing work test cases are prioritized on the basis of timing and number of faults find out. This process has problem that it gives higher priority to least important function and low priority to most important function. So to avoid this problem a new technique will propose which gives priority to the importance of the function not test cases. In this work, we use the log files for taking the values of number of functions encounter. On the basis of these values the FV value is calculated. To extract the value of function encounter, we need to analysis the log files. To implement the proposed algorithm in the real times, we implement the certain log file analyzing algorithm like "random forest algorithm" for analyzing log files.

ACKNOWLEDGEMENT

I would like to express my gratitude to Ms. Isha Sharma and Ms. Jasleen Kaur, faculty members of Computer Science & Engineering Department of GNI, Mullana for their guidance and help in completing this work successfully.

REFERENCES

[1] S.Raju, G.V.Uma, "Factors Oriented Test Case Prioritization Technique in Regression Testing Using Genetic Algorithm",European Journal of Scientific Research,2012.

[2] Corina S, Pasareanu ,Willem Visser , "A survey of new trends in symbolic execution for software testing and analysis", pp 339-353, 2011.

[3] Praveen Ranjan Srivastava, Sirish Kumar, A.P. Singh, G. Raghurama, "Software testing Effort: An Accessment Through Fuzzy Criteria Approch. *Journal of Uncertain Systems* , *5*, 183-203, 2011.

[4] Swarnendu Biswas and Rajib Mall, "Regression Test Selection Techniques: A Survey", Informatica 35 ,289–321, April 2011

[5] Gregory T. Daich. , "Defining a software stratergy" . *STSC, 2009* .

[6] Khan, M. E, "Different Forms of Software Testing Techniques for Finding Errors", IJCSI International Journal of Computer Science Issues , Volume 7, 2009.

[7] Lijun Mei, Zhenyu Zhang, "Test Case Prioritization for Regression Testing of Service-Oriented Business Applications", Proceedings of the 18th International Conference on World Wide Web (WWW 2009), ACM Press, New York, NY (2009).

[8] Dr. Arvinder Kaur, Shivangi Goyal, "A Bee Colony Optimization Algorithm for Fault Coverage Based Regression Test Suite Prioritization", International Journal of Advanced Science and Technology, issue 29 , 2010.

[9] Mangal, B. S, "Analyzing Test Case Selection using Proposed Hybrid Technique based on BCO and Genetic Algorithm and a Comparison with ACO*", IJCA, 2012*.

[10] Ghiduk, A. S , "A New Software Data-Flow Testing Approach via Ant Colony Algorithms", Universal Journal of Computer Science and Engineering Technology , 1, 64-72 , 2011.

[11] Bohringer, Christoph, Rutherford, Thomas F, " Integrating Bottom-Up into Top-Down:A Mixed Complementary Approach", 2003.

[12] S.Raju,G.V.Uma," Factors Oriented Test Case Prioritization Technique in Regression Testing using Genetic Algorithm", European Journal of Scientific Research, 2012.

[13] W. Eric Wong, J. R. Horgan, Saul London, Hira Aggarwal, "A Study of Effective Regression Testing in Practice*", IEEE*, 2008.

[14] R.Pradeepa, K.VimalDevi, " Effectiveness of Testcase Prioritization using APFD Metric:Survey",ICRTCT,2013.