A Module Based Code Coverage approach using Regression Technique for Software Testing

Ajay Kumar Bansal¹, Pallavi B. Vyas², Ashu Gupta³, Manmohan Sharma⁴
¹,²,⁴School of Computer Science & Engineering, Lovely Professional University
³Prince Sultan University, Riyadh, Kingdom of Saudi Arabia

Abstract—Regression testing is an important type of software testing which used to test the whole software to ensure the accuracy of the software after performing some modification on its different modules or part. To reduce the regression testing time, development team uses the prioritization technique to select and prioritize the test cases. Code coverage is one of the prioritization benchmarks for selecting the test cases. In Literature many author proposed their work on code coverage prioritization regression testing. A proposed modular approach is used centered on statements of the program to maximize the code coverage. Our approach works in two levels: First, prioritize test cases for each module; second, find the test cases for whole program by combining the first level prioritized test cases.

Keywords—Regression testing, Code Coverage, Modular program, Test Case Prioritization

I. INTRODUCTION

Software testing plays a vigorous part to provide reliable, bug-free and fault tolerant software to the user. Testing is performed throughout the development and maintenance practice: in Analysis part, Design part and Coding part. In Software Maintenance phase, a development team is supposed to do task which includes improvement of competencies, removal of aptitudes, error corrections, and expansion. These modification tasks may cause the generation of new error in the old modules of software and causes software to work incorrectly. Regression Testing is the process which is performed during the software maintenance to re-verify and re-validate the modified parts of software. To reduce the regression testing time cost, different types of Regression testing techniques used are given below:

To lessen the budget and time of regression testing, there are many prioritization techniques built on some coverage measures like requirement, code, fault, etc. and test case having major exposure is designated by comparing with all other test cases [8]. The author proposed a methodology which is based on maximum code coverage using statements of the code. We have tried to use that one for code coverage instead of fault, and uses statements to select the optimized test cases. In this paper we have explained our code coverage approach theoretically. The rest of the paper is systematized as follows. Section 2 defines the related work. Sections 3 represent major code coverage work in tabular form by different authors. Section 4 presents our theoretical modular approach for code coverage. Finally in section 5, we have discussed conclusions and future work of ourpaper.
II. RELATED WORK

Siripong et al [5] provide idea of only test case prioritization techniques. This study introduces four categories of test case prioritization methods, like: chronographic history techniques, coverage- procedures, cost effective techniques, customer requirement-based techniques and chronographic history-based techniques. Study shows that many research challenges and breaks are found in the test case ordering area. In the field of Regression Testing, these challenges and gaps can give the direction of research.

Gaurav et al [4] have overview of Regression Testing approaches and algorithms. This paper further classified each one of them which are explained Regression Test Choice and Test Case Ordering in detail with various Search Algorithms for Test Case Prioritization. The paper represents full organization of Regression Testing, possible areas of Regression Testing which helps the researchers to explore regression testing importance and scope.

N. Prakash et al [11] proposed test case prioritization approach based on modular for regression testing. It has been conducted in two levels: (1) Local test case, (2) Global test case. A new proposed algorithm is used for this new methodology, which when compare with other algorithm (greedy, additional algorithm) give best results in terms of finding the average fault detection. This methodology applied in three applications USMS, HMS, and IOPS. The existing work is grounded on fault coverage.

Parveen et al [8] proposed a test case prioritization algorithm for prioritizing the test cases which will assign a value for the test case that is to be calculated depending upon the change request that is given by the customer. Code coverage algorithm is based on considering results obtained by executing the test cases which covers modified lines of code. Empirical study performed on two projects and effectiveness of our proposed prioritization algorithm is best as compared to random algorithm.

Navleen Kaur [15] has given discussion on the Prioritized Test Cases which cover multiple criteria that is stronger than single coverage criteria. Proposed algorithm aims to cover multiple criteria such as statement coverage, branch coverage, fault coverage and path coverage using executing single set of Test Cases. According to Naveen optimized proposed work they have implemented Bank Application in java and C++ language. Results are compared Before Prioritization and after multiple criterion prioritizations.

Thillaikarasi Muthusamy [13] proposed an algorithm based on fault coverage. The proposed work is based on 6 factors which are: customer allocated priority, developer pragmatic code execution complexity, variations in necessities, fault effect, inclusiveness. Each of the test cases are assigned weights agreeing to the factors and the the test cases are ordered according to values given to the test cases. APFD metric is used to find the percentage of fault detection in the system.

Md. Imrul Kayes [6] proposed an algorithm based on fault coverage. Fault detection is applied on both prioritized and non-prioritized test cases. Test suitcases are implemented to detect dependency of fault among the test cases. APFDD metric has been used to measure the usefulness of prioritizing test cases. Prioritization technique is used to advance the rate of reliance detection in regression testing.

Muhammad Shahid [7] made survey of different research papers of test coverage and associated different reporting based apparatuses and seen on diverse systems available for producing test cases to placate test handling criteria.

III. REVIEW ON LATEST CODE COVERAGE BASED REGRESSION TESTING ALGORITHM

Till now many code coverage based prioritization techniques and algorithm has been discussed by many researchers. Code coverage has been chosen on some code criteria like statements, loops, path, branch, methods etc. some of the important work in the area of code coverage regression testing approaches has been discussed following:
<table>
<thead>
<tr>
<th>NAME</th>
<th>BASIS</th>
<th>METHODOLOGY</th>
<th>RESULT</th>
</tr>
</thead>
<tbody>
<tr>
<td>“Code Handling Based Test Case Selection and Prioritization”[13]</td>
<td>Statement Coverage</td>
<td>Make three clusters of test cases (Outdated, required, surplus) By using matrix of test cases and statements of program</td>
<td>Among 14 test cases only 3 prioritized TC cover all code.</td>
</tr>
<tr>
<td>“A New Code Constructed Test Case Prioritization Technique”[14]</td>
<td>Method Coverage</td>
<td>Collects the selected test cases centered on the material of all the approaches sheltered by test case.</td>
<td>From prioritized TC array first TC cover 75% methods</td>
</tr>
<tr>
<td>“Method Based on weighted For Coverage Based Test Case Prioritization”[12]</td>
<td>Statement, Path, Branch, Function Fault Coverage</td>
<td>The weight is considered for every test case and for each coverage standards built on the coverage data</td>
<td>Results are good as Compared to 2-optimal algorithm</td>
</tr>
<tr>
<td>“Regression Testing with multiple Criteria based Test case Prioritization”[15]</td>
<td>Statement, Branch Path, Faults</td>
<td>A matrix is used in which one side writes all the Test cases and on the top write all statements, branch no., faults and path. The Test Cases which cover maximum criteria, highest priority is assigned to that test cases.</td>
<td>Values of APFD, APCD, and APSD found better for proposed work.</td>
</tr>
<tr>
<td>“An Well-organized System for Reducing the Test Cases which is Used for Performing Regression Testing”[10]</td>
<td>modified functionality and Test case that cover code</td>
<td>Algorithm which will assign a value for the test case and the value will be calculated depending upon on the change request that is given by the customer.</td>
<td>Better result as compared with random prioritization techniques</td>
</tr>
<tr>
<td>“Efficiency Of Test Case Ordering Techniques Based On Regression Testing”[17]</td>
<td>Fault Coverage</td>
<td>Algorithm use 6 factors (customer allocated priority, developer pragmatic code execution complexity, changes in requirements, fault impact, inclusiveness) to find faults using APFD metric. Prioritize test cases by assigning weights to each test case.</td>
<td>Gives 88% fault when 75% test cases are used and 40% Fault when 30% test cases are used.</td>
</tr>
<tr>
<td>“Test Case Prioritization for System Testing Based on Burden Dependency”[6]</td>
<td>Fault Coverage</td>
<td>APFDD metric has been used to quantify the effectiveness of prioritizing test cases. Prioritization technique is used to recover the degree of dependency detection in regression testing. Test cases are implemented to detect the fault dependency</td>
<td>APFDD metric gives 77% fault detection on prioritized test cases.</td>
</tr>
<tr>
<td>“A Revision on Test Exposure in Software Testing”[7]</td>
<td>Test Coverage</td>
<td>survey was conducted on research papers of test coverage classified on the basis of 6 groups i.e. survey and study of, frameworks, test coverage items, methods/procedures, software</td>
<td>Survey shows that breaks and naked quantity of Test Coverage can be explored further</td>
</tr>
</tbody>
</table>
IV. PROPOSED APPROACH

Our proposed Module based code coverage approach applies on the programs which are divided into modules and each module has its own specific functionality. This methodology arranges the test cases on the base of statement coverage criteria. There are two level of prioritization of test cases in proposed our approach.

Fig2. Diagrammatic Representation of Proposed Methodology

First Level: On the basis of statement coverage, prioritize the test case for individual module and will find optimum testsuit.

Second Level: In Second level we will consider two groups of test suitcases. First group is ideal test cases which are recognized during modular based test case ordering and second set will be mutual test suit for all segment of the software. Now after combining these two groups of test cases, we will order to recognize the universal optimal test cases. To behavior the regression testing more successfully these overall ideal test cases are satisfactory.

Algorithm to find optimum test cases is discussed below:

(TC_ARR - Test case array list)
(Tmcs – Maximum Coverage of Statements)

Input: - Test suite T, number of statement detected by a test case s.

Output: - Prioritized test cases.

Begin
The test suitcases T which cover first statement in Si and the test case T which has association with first consecutive supreme statement Si should be removed and added in the TC_ARR.

Begin
Select the test case Tmcs that has highest relationship with fault Si. Remove Tmcs and its covered statements Si and add it in TC_ARR.

If T has association with statement Si then repeats, otherwise
Exit. End
Categorize the TC.ARR in descendent direction built on the quantity of Statements enclosed by each test case.

End

At last, we will get an array which contains only those test cases which cover almost all statements of the program. Maximum code will be cover at early stages of testing which helps to reduce regression testing time by using these test cases.

CONCLUSION

The ultimate goal of prioritization techniques is to lessen the rate of deterioration testing in term of time and money. In our work we have try to cover all basic and important code coverage approach given by different author. Our approach select the optimum test cases based on statements to cover maximum code earlier so that any faults in testing can be find earlier as soon as possible. In this paper we have represent our work in theoretical manner. Future work can be extended to implement this approach practically on some projects either manually or using some tools.

REFERENCE


[8] A. Pravin and Dr. S Srinivasan “An Efficient Algorithm for Reducing the Test Cases which is Used for Performing Regression Testing” International Conference on Computational Techniques and Artificial Intelligence (ICCTAI’2013) March 17-18, 2013 Dubai (UAE)


