Various Techniques Used For Prioritization of Test Cases

Ekta Khandelwal[*], **Madhulika Bhadauria**[**]

[*] Amity School of Engineering and Technology, India

Abstract- The Software testing phase of the development life cycle model of software plays a very important role in deciding the quality and functionality of the software. The Testing of the software is done to detect the fault in its functionality and on the basis of its performance further quality of software is improved. In the software testing phase there is a task involved which is prioritization of Test cases. In this paper we have discussed about test case prioritization, what is the need of test cases prioritization? And the various techniques proposed by researchers for performing the test case prioritization.

Index Terms- Test Case, Test Suite, Test Case prioritization, fault, APFD

I. Introduction

In general, a Software Development Life Cycle consists of main four Requirement gathering, design and analysis, Development of prototype and then testing and maintenance.
As the name reveals in the first phase requirement from the point of view of the users as well as other stakeholder are collected, in the second phase designing is done, in the third phase implementation is done and the fourth phase is Testing and Maintenance, the testing include the execution of the program or an application with an intent of finding software errors and faults in the application executed. The testing is also known as the process of validating and verifying that software meets the business and technical requirements that guided its design and developments [3], so that it works according to the need of the stakeholders.
Now, on further discussion there is various type of testing that software tester prefer to use according to their need. These can be Mutation Testing, Regression Testing, white Box Testing, black box testing mainly.
Depending upon the type of the testing the tester designs the test Cases and the collection or set of many test cases is known as Test Suite. So during the testing phase to perform the testing the tester decides the test cases, then executes the software with those test cases and then check and verifies the results produced by those executions. After the verification the faults are detected in the application and tell the developer about the improvements required means validation is followed by the verification of the application. The testing id performed to ensure the stakeholders about the quality and functionality of the applications.
The regression testing one of the testing techniques is the testing in which testing is performed on the modified application using the same previously defined sets of Test cases. When an application is developed and it is tested for the first time a set of test cases means test suite is designed to verify and validate its functionality. The tester keeps this test suite with them for further use. When a modification is done in the application then these previously designed test suites are used by testers to ensure that no new errors have been introduced in the previously tested code [4].
But to execute all the test cases to check a small modification is not worth full. It is impractical and inefficient to re execute every test for every program function if once modification occurs. Further it becomes very expensive technique also by executing the all test for a small change. Therefore to reduce the cost of the regression technique and to make it more effective the concept of test case prioritization was introduced by the researchers.
In the test case prioritization the test cases are prioritized and scheduled in order that attempts to maximize some objective function. To decide the priority of the test cases the various factors depending upon the need are decided then the priority is assigned to the test cases. Test case prioritization provides a way to schedule and run test cases, which have the highest priority in order to provide earlier detect faults. Furthermore,
In [3] it is mentioned that Gregg Rothermel has proven that prioritization and scheduling test cases are one of the most critical task during the software testing process as he has given an example of industrial collaborators reports, which shows that there are approx 20,000 lines of code, running the entire test cases require seven weeks. In this situation prioritization of test cases plays a vital role to save the time.
In addition to this [3] have given three reasons that are why we need to do the prioritization of the test cases. These reasons are as below:

The regression testing phase is time as well as cost consuming
Lack of time and resources to run all the test cases again
Need to decide which test cases to run first
Thus in the paper we have discussed the various proposed techniques for prioritization of the test cases.

Further sections are organized as: description of classes on the basis of which various prioritization techniques are classified, description of Prioritization technique based on APFD, Prioritization technique based on Fault Severity, Prioritization in Regression testing in section II, III, IV, and V respectively.

Finally the future work and a conclusion of the discussion are provided in last section.

II. FACTORS FOR CLASSIFICATION OF THE PRIORITIZATION TECHNIQUES

It has been previously said that the techniques for the prioritization of the test cases depends upon various factors. In this section we have discussed the various factors that classify the prioritization techniques in the different classes.

1) Customer requirements –

In the customer requirements based prioritization techniques test cases are prioritized on the factors decided on keeping in mind the requirements of customers documented during the phase of requirements gathering.
Some of the factors proposed to rank the test cases for this approach are customer assigned priority (CP), requirement complexity (RC) and requirements volatility (RV). These factors are assigned a value and higher factor values indicate a need for prioritization of test case related to that requirement.

2) Coverage Based-

In the prioritization techniques based on coverage the prioritization of the test cases are on the quantity of the source code of a program that has been exercised during testing [3].
In the context of testing the word coverage means the part of code that has been covered during the process of the testing or it can be requirement coverage, total requirement coverage, additional requirement coverage; therefore in this approach the test cases having the capability of testing the major part of code are prioritized first and so on.

3) Cost Effective-

In this approach the test cases are prioritized on the basis of the cost factor. The cost can be the cost of requirement gathering, cost of regression testing, cost of execution and validating test cases, cost of analyses to select and support a test case, cost of prioritization of test cases. Thus, the test cases requiring the less cost will get the higher priority.

4) History based-

In this approach the test cases are prioritized based upon the history of the test case itself it means priority of test case depends upon its previous performance. The execution history of the test case increases or decreases the likelihood that it will be used in the current testing session [3].

For the detailed study of these four approaches discussed you can refer to [1].

III. PRIORITIZATION BASED ON APFD

In this section we have discussed a test case prioritization technique which is based on the rate at which fault is detected in the code under test. In the [1], they have proposed a technique in which two criteria are used to decide a factor on the basis of which the priorities to the test cases are assigned.

The first criterion is the number of the code lines or statement covered by a test case. The first is symbolized by St

Statement coverage (St) = (No. of statements covered /total number of statements) * 100

The second criterion is number of function calls in the covered statement. This is symbolized by Fc.

No. of Function calls (Fc) = Local functions + nested functions + private functions + overloaded methods

Both the criteria are counted in the numerical value.

The factor used to prioritize the test cases is the product of these two numerical values and this factor is symbolized by P.

Product (P) = Statement coverage (St) * Number of the Function Calls (Fc)

Thus the value of P for each test case of a test suite is calculated using this formula and the test case having higher value is given the higher priority.

In [1], they have used a metric known as APFD (Average Percentage of Fault detected) to compare the fault detection rate of the prioritized and non prioritized test cases. The APFD for both the test suites is calculated and then compared. The formula used to get the value of APFD is given below

$$APFD = 1- \{(TF1 + TF2 + TF3 \ldots\ldots TFm) / mn \} + (1/2n)$$

Where,

n is the number of test cases in a test suite
m is the number of faults
TFi is the position of first test in T that exposes fault i.

In [1] they have proved that the value obtained for the test suite having prioritized test cases has the higher value for the APFD. For further detailed study of this approach you can refer to [1].

IV. PRIORITIZATION USING FAULT SEVERITY

In this section we are going to discuss a technique which is based on prioritization of technique in which priority to the test cases are assigned depending upon the priority of the requirement to be tested. The requirement to be tested first is assigned the higher weight and the test cases covering that requirement are given higher priority of execution. In this approaches the requirements considered are based on fault severity that is the number of times the fault can occur in the code that code is required to be tested first thus given the higher weight.

The four factors are proposed by [2], to decide the weight of the requirement. These four factors are described in brief below

1) Business value Measure-

This is the factor in which the requirements are assigned rank according to their importance. Most critical requirement is assigned higher number (10) and least important requirement is assigned the lowest number (1).

2) Project Change Volatility-

This factor depends upon the times consumer is modifying the project requirements during the software development cycle. The biggest causes of the project failure happen to be the lack of user inputs and the changing and incomplete requirements [4].

3) Development Complexity-

Development efforts, technology used for development, environmental constraints and the time consumed or required decides the complexity of the development phase.

4) Fault Proneness of Requirements-

This is the very direct factor for assigning weight to requirement. This factor considers those requirements which are error prone according to the historical data such as requirement failure reported by the customer.

Thus on the basis of this weight is assigned to each requirement and depending upon this the weighted prioritization factor that measures the importance of testing a requirement earlier is calculated and the test cases are assigned priority. This is the first step of the proposed technique.

The second step is to assign severity measure to the each fault. The range of severity measure varies as below:

Complex (Severity 1): SM value of 9-10
Moderate (Severity 2): SM of 6
Low (Severity 3) :SM of 4
Very Low (Severity 4): SM of 2

This is done to calculate the total severity of the Faults Detected (TSFD). This is the summation of the measures of the all faults identified.
This is how the test cases are prioritized first requirements are prioritized then test cases against each requirements is mapped , then executed according to the priority assigned and the results are analyzed based on fault severity.

V. PRIORITIZATION IN CASE OF REGRESSION TESTING

In this section prioritization of test cases in case of regression testing will be discussed. In [5], authors have presented nine techniques which could be used for this purpose. These techniques are discussed in brief as follows:

1) No prioritization

In this case no techniques are implemented and is been used as a untreated test suit and it serves as a control.

2) Random prioritization

This is applied to have an additional control in studies where the test cases are ordered randomly in the test suite.

3) Optimal prioritization

In this technique known faults are been used so that its results can be used to measure the effects of other prioritization techniques which are to be used

4) Total statement coverage prioritization

This technique instrument the program with any test cases and finds out that which statements were covered by the test cases; then these test cases can be prioritized on the bases of number of statements they covered. If more than one tests case covers equal number of statements then we have use some additional rules or we can order them randomly.

5) Additional statement coverage prioritization

This method covers the shortcomings of total statement coverage technique and iteratively selects a test case that gives the highest statement coverage and then adjusts the coverage information on rest of the test cases to find out their statements not yet covered. This whole process is repeated till at least one test case covers all the statements.

6) Total branch coverage prioritization

This technique is same as statement coverage prioritization but it just uses test coverage measured in form of program branches as there was the case of statements in other technique.

7) Additional branch coverage prioritization

This technique is same as additional statement coverage prioritization but the only difference is that it uses test coverage measured in term of program branches and not in statements.

8) Total fault-exposing potential prioritization

Other method like statement and branch coverage takes in context that whether a statement or branch is been reached by some test cases or not and do not takes in context the case that some faults are more easily seen as compared to others. As some test cases can expose faults more easily as compared to others, so it is called as fault-exposing potential of a test case.

9) Additional fault-exposing potential prioritization

As additions were made in total coverage and branch coverage prioritization, extensions were made in total fault-exposing potential prioritization and as a result this technique is created where we extend total FEP to create additional FEP prioritizations.

VI. CONCLUSION AND FUTURE WORK

In this paper we have briefly discussed and reviewed the various techniques for the test cases prioritization and the various factors responsible for classification of the techniques. All the algorithms have their own field of use.

Further research will be focused on studying the other prioritization techniques and on implementing the proposed algorithms in order to get the good results in testing phase of the application.

REFERENCES

[1] Sahil Gupta, Himanshi Rapria, Eshan Kapur, Harshpreet Singh, And Aseen kumar " A Novel Approach for Test Case Prioritization " at IJCSEA, Vol. 2, No.3, June 2012

[2] Dr. Varun Kumar, Sujata, and Mohit Kumar "test Case prioritization Using Fault Severity" at IJCST, September 2010

[3] Sripong Roongruangsuwan, and Jirapun Daengdej "Test Case Prioritization Techniques" at JATIT & LLS, 2010

[4] Praveen Ranjan Srivastava "Test Case Prioritization" at JATIT, 2008

[5] Gregg Rothermel, Roland H. Untch, Chengyun Chu and Mary Jean Harrold "Prioritizing Test Cases For regression Testing" at IEEE, 2001

[6] Gaurav Duggal, Mrs. Bharti Suri "Understanding Regression Testing Techniques" 2006

AUTHORS

First Author – Ekta Khandelwal, M.tech, Amity School of engineering and technology.

Second Author –Mrs. Madhulika Bhadauria ,Assistant Professor and currently pursuing P.hd from , Amity School of engineering and technology.