

AN APPROACH FOR TEST CASE PRIORITIZATION BASED UPON VARYING REQUIREMENTS

ASEEM KUMAR¹, SAHIL GUPTA², HIMANSHI REPARIA³, HARSHPREET SINGH⁴

¹Research scholar, Department of Computer Engineering, Lovely Professional University, Phagwara, India
aseem.kumar6@gmail.com

²Research scholar, Department of Computer Engineering, Lovely Professional University, Phagwara, India
sahil.gupta2688@gmail.com

³Assistant Professor, Department of Computer Engineering, Lovely Professional University, Phagwara, India
himanshi19.rpr@gmail.com

⁴Assistant Professor, Department of Computer Engineering, Lovely Professional University, Phagwara, India
harshpreet.15790@lpu.co.in

ABSTRACT

Software testing is a process continuously performed by the development team during the life cycle of the software with the motive to detect the faults as early as possible. Regressing testing is the most suitable technique for this in which we test number of test cases. As the number of test cases can be very large it is always preferable to prioritize test cases based upon certain criterions. In this paper prioritization strategy is proposed which prioritize test cases based on requirements analysis. By regressing testing if the requirements will vary in future, the software will be modified in such a manner that it will not affect the remaining parts of the software. The proposed system improves the testing process and its efficiency to achieve goals regarding quality, cost, and effort as well user satisfaction and the result of the proposed method evaluated with the help of performance evaluation metric.

KEYWORDS

Regression testing, Test case prioritization, test cases, Test case weightage, requirement analysis

1. INTRODUCTION

Basically Software is the process in which series of instructions for the computer that performs a particular task called a Program. System software and application software are the two major categories of software we have in the current scenario. Any program that processes data for the

user is known as application software for example word processor, payroll, Application software etc. [1].

Generally a software product should only be released after it has gone through a proper process of development, testing schemes and bug fixing. Areas such as performance, stability and error handling by setting up test scenarios under controlled conditions and assessing the results comes under software testing. This is exact reason why software has to be tested. The main purpose of software is mainly tested to check that it meets the customer's needs, expectation etc. It is a usual norm that software is considered of good quality if it meets the all the requirements of user. [9]

Whenever software is under the process of modification or under the process of updating some test cases needs to run and the comparison of new outputs based upon test cases is done with the older one to detect and to curtail any unnecessary changes. As a result of these comparisons if the proper matching of both new and old output (after modification) is there then it leads to the conclusion that the amendments in one part of the software would no longer be effective to the other parts of the software. [1] Changes in the requirements or amendments in maintenance or evolution purposes may break existing working features, or may violate the requirements established in the previous software releases. If the requirements keep on varying it leads to the effect the existing working of the software. To resolve the above problems Regression testing is essential. Regression testing is one of the most crucial and critical activities of software development and maintenance purpose [9]. Regression testing has been used during the development and maintenance of a software product to assist software-testing activities and guarantee the attainment of adequate quality through various versions of the software product [2]. Regression testing permits to test modified software to provide confidence that no new errors are introduced into previously tested code. Regression test selection techniques attempt to reduce the cost of regression testing by selecting and running only a subset of the test cases in existing test suite [3]. Whenever software is modified with regression testing execution of number of time consuming test cases required. As In regression testing large number of test cases needs to be run therefore it is an impractical approach to run all the possible test cases. In order to make it much simpler prioritization technique of the possible test cases proposed, in which test cases having high priority according to some criterion, executed first then those having lower priority so that to enhance the performance.

2. TEST CASE PRIORITIZATION

In last few years, several researchers have done good amount of research on the test case prioritization problem and presented techniques for the same. Research has shown that at least 50% of the total cost of software development consists of testing activities [4]. which is very high as compare to the other cost consuming activities. Proper behaviour of the software can be ensured and absence of unexpected faults also called as regression faults in the introduced changes can be confirmed by employing regression testing to the software [5]. Rate of regression fault detection can be increased up to greater extent during software development by adopting test case prioritization techniques [6]. In prioritization techniques test suites are prioritized according to some factors. Very less amount of researchers inclined towards requirement analysis phase using this technique.

3. PROPOSED TEST CASE PRIORITIZATION STRATEGY

In order to cover the changing requirements and their impacts over other modules weightage is calculated of test cases on basis two requirements based factor Which are considered as *Rfactors* and requirement priority (*RPriority*) computed from those two *Reactor's R-value and Rweight*. In the proposed strategy an algorithm is proposed which prioritized the test cases according to the test case weightage (*TCW*).

3.1. Requirement Factors (*RFactors*)

On the basis of requirement analysis the two factors are considered and later on *RValue* of requirement factor calculated for individual factor. The *Rfactors* considered are as follows:

3.1.1. Requirement Modification Impact Localization (*RMIL*)

It is the measure of how much impact produced by the change in any requirement during the development phase from its original state i.e. change of state from the first time when requirements are gathered from the user. It is a measure assigned by the developer initially between the ranges of 1 to 10, which is the *RWeight* of this *Rfactor*. It is computed from the number of times that particular requirement changed till present time from the first time requirements gathered from the user in the requirement analysis phase of software development. The values for all the requirements are assigned on a 10-point scale which states that a requirement is altered more than 10 times. *RValue of RMIL* is calculated by dividing the number of changes for any requirement R divided to the maximum number of changes to be made as a result of change in that requirement R among entire project requirements. If the requirement R is changed P times and Q is the maximum number of requirements the *RValue* of *RMIL* of R is calculated as

$$RValue\ of\ RMIL = (P/Q) * 10$$

To calculate *RValue* on a 10 point scale we multiply with 10 in above equation.

3.2.2. Degree of Coupling (*DC_p*)

It is the Degree of coupling of the particular requirement with other requirements. It suggests the how much one requirement is interlinked with other requirements, and the number of levels of changes to be made in other requirements with the change of a particular requirement. Its weight is assigned developer initially by the development between the ranges of 1 to 10. And this is the *RWeight* of this *Rfactor*. In this paper we have discuss three levels of coupling to calculate its *RValue* which are as follows:

3.2.2.1. Level 1

This level of coupling states that change of one requirement leads to change in one requirement. It means one particular requirement is interlinked with 1 another requirement. The Relation among the those two requirements is 1:1

3.2.2.2. Level 2

This level of coupling states that change in one requirement leads to the change in 2 or more requirements. It means one particular requirement is interlinked with 2 another requirement Relation among the requirements is 1: N where N is ≥ 2

3.2.2.3. Level 3

This level of coupling states that change in one requirement leads to the change in 3 or more requirements. It means one particular requirement is interlinked with 3another requirement Relation among the requirements is 1: M where M is ≥ 3

To understand this concept more consider a practical example of a student who takes admission in a university under a specific course name, and registration no. And allotment of hostel and food facility in hostel also has done with these credentials of the student. If later on student changes his course then his course name also changes and as a result of this necessary changes should be made in the records of hostel and changes made in hostel allotment records leads to the changes in the records of food facility of hostel. Thus this change in student course name leads to the changes up to two levels of requirements.

Logically if the course name is requirement R_1 during admission of student it is R_2, R_3 while allotment of hostel and food facility respectively
Then

$$R_1 R_2 \text{ (Level 1)}$$

$$R_2 R_3 \text{ (Level 2)}$$

From the equation it is clear that changes in R_1 leads to changes in R_2 and changes in R_2 leads to changes in R_3 i.e. Requirement R_1 is interlinked with two requirement and changes in R_1 leads to the changes up to two levels. This is an example of Level 2 type of coupling.

Therefore DC_{p} i.e. Degree of coupling for of R_1 is 2. Which means its $RValue$ is 2

3.2. Prioritization Technique

In this paper the proposed work goes in two different steps, each step has its own importance and impact. Those two steps are as follows

3.2.1. Computation Of Requirement Priority ($RPriority$)

Value of Requirement Priority ($RPriority$) states that priority the test case that consists of a particular requirement. $RPriority$ for each of the requirement r is calculated from the above stated two factors. Higher the $RPriority$ higher the need of testing the test case containing that requirement.

$$RPriority = \sum_{n=1}^K (R.factor\ Value \times R\ Factor\ Weight) \dots\dots\dots (1)$$

Where k is the no of *Rfactors* considered. We consider two *Rfactors* , k =2

3.2.2. Test Case Weightage (TCW)

Every test case will be weighted before prioritization based upon certain criteria so that they can be tested accordingly. TCW in the proposed technique is based on *RPriority*. Test cases consists of high priority requirement calculated from equation 1 higher the priority of that test case.

3.3. Proposed Prioritization Algorithm

After computing the test case weightage for each of the test case, on the basis of TCW each test case will be prioritize before testing with the help of proposed algorithm in the proposed prioritization strategy. The algorithm gets input as test suite T, test case weightage of each test case is computed and the output of the algorithm is the final prioritized test case order.

Algorithm:

1. Begin
2. Set T' empty
3. for each test case t ∈ T
4. Calculate $RPriority = \sum_{n=1}^K (R.factor\ Value \times R\ Factor\ Weight)$
5. Deduce TCW from *RPriority*
6. end for
7. Sort T in descending order on the value of test case weightage
8. Let T' be T
9. End

4. PRIORITISED TEST SUITE EFFECTIVENESS

After ordering the various test cases using prioritization strategy the performance of test suite also needs to be evaluated. It may be measured using various methods. It is always on a positive side to assess the effectiveness of the ordering of the test suite. Effectiveness is being measured by the rate of faults detected. We deduced a metrics M_{RP} from APFD metric that is used to calculate the level of effectiveness

4.1 M_{RP} Metric

The metric which was developed by Elbaum et al [8] that measures the rate of fault detection per percentage of test suite execution known as APFD metric.

APFD can be calculated as:

$$APFD = 1 - (Tf1+Tf2+Tf3 \dots\dots\dots + Tfm) / nm + 1/2n$$

Where n is the number of test cases and m is the number of faults detected .With the help of APFD metrics (a performance analysis matric for test cases) to measure the effectiveness of proposed prioritizationstrategy APFD metric can be deduced following metric based the requirements and its priority[10]

$$M_{RP} = 1 - (TR1+TR2+TR3 \dots\dots\dots + TRm) / nm + 1/2n$$

Where n is the number of test cases and m is the number of requirements.

5 AN EXAMPLE

To understand to working of the proposed technique suppose we have requirement of the project as R= (R₁, R₂, R₃, R₄), Test Suite T= (T₁, T₂, T₃, T₄), RValue and Rweight also suggested in the table below. We can calculate the RPriority and TCW for each test case.

Table1. Test Cases Containing Requirements

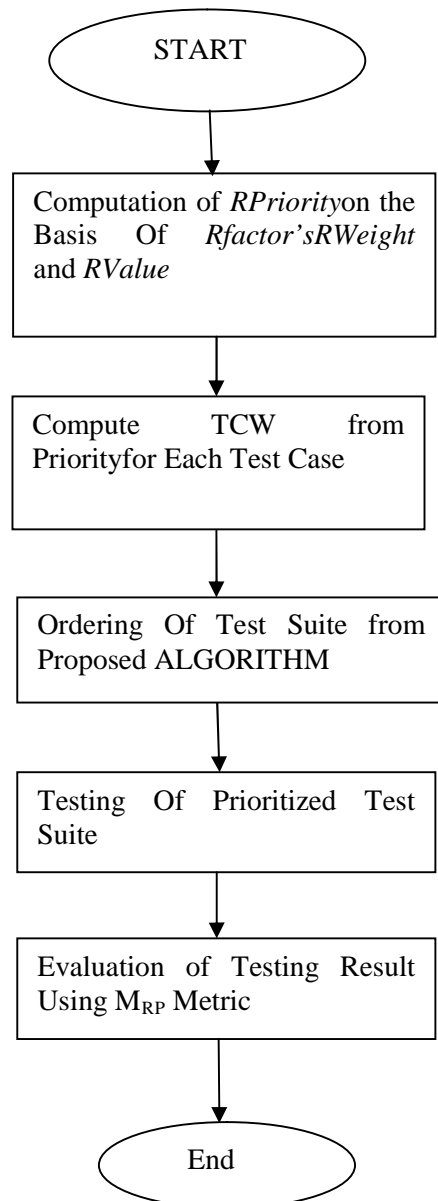
R T	R ₁	R ₂	R ₃	R ₄
T1	X	X		
T2	X			X
T3	X		X	
T4	X	X		

Table 2. RWeight and RPriority

RFactor Weight (RWeight)	R RFactor	R ₁	R ₂	R ₃	R ₄
2	RMIL	3	6	5	2
2	DCp	3	5	4	5
RPriority		12	22	18	14

From the Table 2. It is clear that $RPriority R_2 > RPriority R_3 > RPriority R_4 > RPriority R_1$. Thus the order of execution of test cases weightage (TCW) computed from algorithm will be $T_1 > T_3 > T_2 > T_4$. Test cases will be executed in this order and finally result will be evaluated with the M_{RP} Metric to compute the effectiveness of proposed strategy.

6. WORK FLOW OF PROPOSED PRIORITIZATION STRATEGY



7. CONCLUSION

In this paper we describe requirement based test case prioritization technique. This proposed technique is highly useful to identify and evaluate various issues arises while working with varying requirement environment. The proposed prioritization technique used most efficient factors to prioritize test suite because the errors introduced in the requirement phase is approximately 50% of all faults detected in the entire project. The change in requirements is the major factor attributable to the failure of the project so we prioritize test cases according to requirement priority and requirement factors.

ACKNOWLEDGEMENTS

We would like to express my gratitude to god and all the people who helped us in compilation of this work and devote their precious time in this work.

REFERENCES

- [1] Glenford J.Mayers, 2004,"The Art of Software Testing", John Wiley & Sons publication
- [2] Hyuncheol Park, HoyeonRyu, JongmoonBaik, (2008) "Historical Value-Based Approach for Cost-cognizant Test Case Prioritization to Improve the Effectiveness of Regression Testing", The Second International Conference on Secure System Integration and Reliability Improvement, pp39-46
- [3] Henry Muccini, Marcio Dias, Debra J. Richardson,(2006)"Software Architecture-based Regression Testing", Journal of Systems and Software, Vol 79,No 10, pp 1379-1396
- [4] M. Harrold, (2000), testinga roadmap, International Conference on Software Engineering,Limerick, Ireland, 2000, pp. 61–72.
- [5] Alessandro Orso, Nanjuan Shi and Mary Jean Harrold, (2004) "Scaling regression testing to large software systems"(2004) 12th ACM SIGSOFT twelfth international symposium on Foundations of software engineering, Vol.29, No.6,
- [6] Sujatha, Mohit Kumar and Varun Kumar, (2010) "Requirements based Test Case Prioritization using Genetic Algorithm", International Journal of Computer Science and Technology, Vol.1, No, 2, pp.189-191.
- [7] R. Kavitha and N.Suresh Kumar, "Factors Oriented Test Case Prioritization Technique in Regression Testing "European Journal of Scientific Research ,ISSN 1450-216X Vol.55 No.2 (2011), pp.261-274
- [8] S. Elbaum, A. G. Malishevsky and G. Rothermel,(2001), "Incorporating varying test costs and fault severities into test case prioritization" ,23rd International Conference of Software Engineering, pages 329-338
- [9] C.Padmini 2004 "Beginner Guide to Software Testing".
- [10] Xiaofang Zhang ,Changhai Nie,Baowen Xu,Bo Qu (2007) "Test Case Prioritization based on Varying Testing Requirement Priorities and Test Case Costs" 17th International Conference on Quality Software (QSIC 2007)

Authors

Aseem Kumar received Bachelor of Technology from Lovely Professional University, Punjab, India and Currently Pursuing Masters in Technology from Lovely Professional University. His areas of interest include Software Testing, Test Case Prioritization, and Requirement Prioritization



Sahil Gupta received Bachelor of Technology from Lovely Professional University, Punjab, India and Currently Pursuing Masters in Technology from Lovely Professional University. His areas of interest include Software Engineering, Test Case Prioritization, and Functional Testing



Himanshi Reperia is Post Graduate in Computer Science and is currently working as Assistant Professor at Lovely Professional University. She has a teaching Experience of around 1 year. Her areas of interest are Software Engineering and Testing



Harshpreet Singh is Post Graduate in Computer Science and is currently working as Assistant Professor at Lovely Professional University. He has a teaching Experience of around 1 year. His areas of interest include Software Engineering, Software Testing, and distributed systems.

