Classification Model for Test Case Prioritization Techniques

Sujata
Dept. of Computer Science, SOE
GD Goenka University, Gurugram,
Haryana, India
sujata.bhutani@gmail.com

G.N. Purohit
Dept. of AIM & ACT,
Banasthali Vidyapeeth
Banasthali, Rajasthan, India
gn_purohitjaipur@yahoo.co.in

Abstract— **Regression Testing is mainly done in software maintenance aiming to assure that the changes made in the software have correctly been implemented and also to achieve the confidence that the modifications have not affected the other parts of the software. It is very costly and expensive technique. There are number of techniques present in literature that focus on achieving various testing objectives early in the process and hence reduces its cost. Despite of that, testers usually prefer only few already known techniques for test case prioritization. The main reason behind is the absence of guidelines for the selection of TCP techniques. Hence, this piece of research introduces a novel approach for classification of TCP techniques using fuzzy logic to support the efficient selection of test case prioritization techniques. This work is an extension of already proposed selection schema for test case prioritization techniques. To perform the validation of proposed approach results are compared with other classification techniques using Weka tool. The analysis clearly shows the effectiveness of proposed approach as compared to others in terms of its accuracy.**

Keywords— ***regression testing; test case prioritization; classification; fuzzy logic***

I. INTRODUCTION

Regression testing in maintenance phase is actually the process of retesting the updated software to ensure that new errors have not been introduced into earlier validated code. Moreover, regression testing should take up as little time as possible in the sense of executing as few test cases as possible[21]. Due to its expensive nature various techniques exist in literature that focus on cost. These are: (i) Re-execute all; (ii) Test case Minimization/ Reduction; (iii) Test Case Selection; (iv) Test Case Prioritization; (v) Hybrid approach[6,13]. This paper focuses on test case prioritization techniques.

Test Case Prioritization

Test case prioritization techniques execute test cases in an order to some criterion. The purpose of this prioritization is to increase the likelihood that they will meet the specific objective more closely than they would if they were executed in some random [8, 9]. Test case prioritization can address a wide variety of objectives. The test case prioritization can be defined as [8,9]

Given: For a given test suit S for a given program X; XS, the set of permutations of S;
f, a function from XS to the real numbers.
Problem: Find $S' \in XS$ such that $(\forall S'')$ $(S'' \in XS)$ $(S'' \neq S')$ $[f(S') \geq f(S'')]$.

In this definition, XS is the set of possible combination to prioritize the test cases of test suit S, and f is an objective function. For example, testers may wish to increase the coverage of code in the software under test at a faster rate, increase or improve their confidence in the reliability of the software in lesser time, or increase the rate at which test suites detects faults in that system during regression testing.

Test case Prioritization problem was formally defined by Rothermel[8,9] and he studied nine TCP techniques mentioned in table-1. Among them four were based on code coverage and two were based on improve rate of early fault detection. These were compared with no prioritization and random prioritization techniques. Optimal category is possible only in literature practically can' be achieved[6]. The experiments show the early fault detection with Greedy and Additional Greedy algorithm in testing process for code coverage[15]. To measure the test objective Rothermel described a metric, APFD that quantify weighted average of % of detected faults over the fraction of test suit(%) executed. Its values range between 0-100 and average higher the value is better the fault detection.

Consider a test suit T having n number of test cases; F is a set of m faults detected by test suit T. TFi is the first test case in T'(one of the ordering of T) that reveal a fault i. Then APFD for T' is defined by the following equation[8,9,14]:

$$Average\ Percentage\ of\ Fault\ Detected = 1 - \left(\frac{TF1 + TF2 + \cdots TFm}{nm}\right) + \left(\frac{1}{2n}\right)$$

TABLE I. CODE BASED TCP TECHNIQUES

Type	Key	Depiction
Tech-0	*No Prioritization*	Based on control; no prioritization
Tech-1	*Random* Technique	No proper ordering; randomly ordered; test cases are chosen

		randomly
Tech-2	*Optimal* Technique	Test cases are scheduled to optimize rate of fault detection
Tech-3	*T-branch*	Test cases are prioritized and executed in an order that achieves the maximum coverage of various code branches
Tech-4	*T-add-branch*	Test cases are executed in order that maximize the coverage of those code branches that have not been covered that is based on the feedback.
Tech-5	*T-FEP*	Ordered test cases based on exposing faults ability.
Tech-6	*T-add-FEP*	Ordered test cases based on exposing faults ability including the feedbacks from previous states.
Tech-7	*T-stmnt*	Test cases are executed in an order that increases or maximizes the coverage of various code statements[10]
Tech-8	*T-add-stmnt*	Test cases are prioritized in an order that maximize the coverage of statements that have not yet been covered (feedback based)[10]

Subsequently various studies have been done on TCP techniques. **Hema Srikanth[[10]** proposed a new approach based on requirement prioritization to prioritize the test cases by considering customer priority as an important prioritization factor. **Sujata[17]** proposed another approach to prioritize the test cases by considering severity of faults. This is based on 80-20 principle. It says that most of the (80%) features/component or function of code are used only 20 times. So, faults should be detected first from that part of the software. The results, in terms of Total Severe Faults Detected, were measure through WPFD metric. Gary performed a study for specification based testing environment that improves the detection of severe faults early in the testing process. To optimize the order of test cases, metaheuristic approach are used by considering inter- case dependency[7].

The major issues of code based prioritization techniques is that they focus only on number of faults detected and, hence, treats all faults equally. In practice, all faults can't be treated same. Moreover Testing does not assure about the absence of faults it is the process of validate the software against users' specifications and requirements. Therefore, there may be some cases where existence of fault is not so important but its requirement coverage is . Requirement based test case prioritization address such issues by assigning the priority to test cases on requirement coverage based. Major Issue with specification based and requirement based Test Case Prioritization is that there is no efficient way to measure the effectiveness of selected Test Suits.

Mark Harman formally defined the efficiency of meta-heuristic and evolutionary algorithms for code based test case prioritization techniques. The study clearly mentions the effectiveness of evolutionary algorithms over the Greedy and Additional Greedy despite of the coverage criteria that has been chosen[12]. However the size of program may increase the complexity of computing fitness value. Hence, Genetic Algorithm outperform the cases in which entire ordering is to be considered due to complex nature of the projects.

The use of fuzzy logic in the field of testing specially in regression testing is not the new. Various authors have proposed various techniques in this regard. Some has used fuzzy logic for the classification, some to improve the effectiveness and, some for optimization.

Pedro Almir Oliveria *et al* (2016) proposed a new technique for performing test case selection proceeded by prioritization and, further test minimization, at the same time, as a new hybrid approach. This research is the fusion of two intelligence techniques - fuzzy inference and meta-heuristics[5].

Ali M. Alakeel proposed a novel test case prioritization technique using fuzzy logic to improve the effectiveness of a given test suit by violating assertions and which further subjected for prioritization of test cases. The best of this research is that it can be used in both environment either White box testing or black box testing environment. The aim of this technique is to reorder the test cases according to rate of violating assertions from the source code[1].

Anwar et al performed a comparative study of various test case prioritization technique and used fuzzy logic for optimization of test suit and hence regression testing process[20].

Risk estimation of software requirements is a major factor to improve the quality of the software. **Charitha et al[4]** proposed a new approach for risk estimation using fuzzy expert system to improve the effectiveness of TCP in regression testing process. The proposed approach is a systematic study that reduces the theoretical aspects and moves the research towards pragmatic environment.

All these studies show the generation of test case prioritization techniques. As the time evolves the nature of these studies also evolve. Although a number of techniques have been proposed but most of the techniques are limited to code coverage based mechanism and focus on finding maximum number of faults. According to survey performed by **Catal(2013)** on Test Case Prioritization Techniques maximum number of techniques proposed till date are Coverage (code) based (40%)and least importance is given to cost aware(2%) and distribution based(2%) techniques.

A number of TCP techniques are there in the literature but no guidelines are available for the practitioners for their selection. From the past decade study has been moved from code based to requirement based, single objective to multi-objective and finally one evolutionary algorithm like Genetic to some optimized one (like Fuzzy Optimization) but still testers are lacking with the knowledge which specific technique should be used for a particular scenario. Due to this lack of knowledge in this field and complex nature of regression testing, testers mostly opt already known TCP techniques based on their experience. All most all the researches have confided the efficiency of TCP techniques always depend upon the nature of project domain and characteristics of test suit. According to Harman[19], the Test Case prioritization, selection and Test Case Minimization Problems are really hard to solve efficiently. And hence, the use of public data sets is suggested for the software testing research to make them verifiable and repeatable[3].

Vegas and Basili proposed a characterization schema for selecting the testing technique based on various project characteristics[2]. This schema is aiming at providing a solution for the testers in selection of software testing technique. In this research a repository is made that depicts the various characteristics of a particular technique and also describes all the techniques over a same pattern so that a knowledge based choices can be made.

II. Classification Model for TCP Techniques

A. Intorduction

This piece of research is an extension of proposed selection schema[16] and provides a Model for selection of test case prioritization technique based on three factors: (i) requirement coverage, (ii) efforts and, (iii) complexity

In selection schema identification of relevant project attributes/features is done to identify TCP techniques covering maximum project attributes consequently requirements. Selection Schema assumes that requirement weights are uniform. Once techniques have been identified next step is to classify those techniques based on maximum coverage and minimum testing effort and, complexity. To calculate the effort this research is using the same baseline as was used by Krishnamoorthi. Hence, testing effort represents average number of test cases required by a particular technique for effective testing. Complexity of a technique plays a measure role in estimation of testing efforts. According to survey performed[16] complexity can be taken on the scale of 1-10 usually defined by the developer and analyst. So this piece of research achieves maximum coverage with minimum complexity and testing efforts.

Here, effort is measure in terms of two in two terms: Final Total effort (FTE) and Average Effort (AE). TE is the summation of all test cases run for a particular technique[11] for all available stories.

$$FTE_n = \sum_1^n TE_j \qquad (2)$$

where $\sum_{j=1}^{n} TE_j$ is the number of test cases executed for each story in a particular project[11].

If there are T number of total test cases then Average Effort (AE) for a particular technique is computed as[11]

$$AE_i = \left[\frac{FTE_i}{\sum_{j=1}^{n} T_j}\right] \qquad (3)$$

Where FTE_i is the Final Total Effort for each story from 1 to n and $\sum_{j=1}^{n} T_j$ is the total number of test cases in each story from 1 to n[11].

Relevance of various project features should not be the only criteria for selection of TCP techniques. There are some more parameters like complexity of requirements and hence of the testing techniques, efforts in terms of average number of test cases run which play an important role in selection of TCP techniques. Hence, this paper introduces a novel approach for classification of various TCP techniques that has been selected based on their maximum coverage with minimum efforts considering complexity of the techniques too. The proposed approach uses fuzzy logic system to classify the various TCP techniques. The use of fuzzy logic in the field of testing specially in regression testing is not old. Various authors have proposed various techniques in this regard. Some has used fuzzy logic for the classification, some to improve the effectiveness and, some for optimization.

B. Proposed Approach

The various stages of proposed approach are as follows:

Stage-1 Identifying project features' in terms of relevance and hence coverage of requirements[16].

Stage-2 Identify the complexity of testing techniques.

Stage-3 calculating testing effort

Stage-4 classify TCP techniques using fuzzy inference system.

Stage-1 uses the prioritization matrix that was proposed in Selection Schema. Stage-2 finds the complexity of a particular technique against the coverage of its requirements. Complexity of TCP techniques is considered on the scale of 1-10 based on the survey conducted. To calculate the effort of a particular technique above equation-2 is used. According to formulae, effort will come on the scale of 0-1. In this approach we will normalize it by multiplying with 10 to consider on equal scale as of coverage and complexity. In the final stage classification is done using Fuzzy Rule based system. The output of this system is mapped to following fuzzy sets: Select, Moderate and, Discard. Input variables as well as output variables can take the values between 1-10. In this study, triangular membership functions are used for the mapping of fuzzy sets and input variables during fuzzification and also for the mapping of output variables and fuzzy sets during defuzzification. Input variables are mapped to three fuzzy sets each: Low, Medium and High.

This proposed Fuzzy rule based systems has 17 rules in total and further reviewed many times by considering experts knowledge. These rules are based on the following experts' suggestions achieved through surveys:

1. For selection of any technique most important factor is project characteristics coverage and hence requirement coverage.
2. Efforts is decided and calculated once the requirements are final. So techniques should be selected with the maximum coverage and less efforts in achieving those characteristics.
3. Complexity also plays important role because it is only the complexity that decides the effort for a particular technique.

Hence, following these observations the rules were formed as follows:

TABLE II. RULE BASE FOR FUZZY BASED SELECTION OF TCP TECHNIQUES

Rule#	**Rule hypothesis**	**Class (Rule outcome)**
1	If (rel = low && comp ≠ high && effort = low)	Discard
2	If (rel = low && comp ≠ high && effort = medium)	Discard
3	If (rel = low && comp = high && effort ≠ high)	Discard
4	If (rel = low && comp = high && effort= high)	Discard
5	If (rel = high && comp = medium && effort = medium)	Select
6	If (rel = high && comp = low && effort = medium)	Select
7	If (rel = high && comp = low && effort = low)	Select
8	If (rel = high && comp= medium && effort = low)	Select
9	If(rel = high && comp ≠ high && effort = low)	Select
10	If (rel = medium && comp ≠ high && effort = low)	Moderate
11	If (rel = medium && comp ≠ high && effort = medium)	Moderate
12	If (rel = medium && comp ≠ high && effort = high)	Discard
13	If (rel = medium && comp = high && effort = high)	Discard
14	If (rel = medium && comp ≠ high && effort = medium)	Moderate
15	If (rel = low && comp = high && effort ≠ medium)	Discard
16	If (rel == medium && comp== high && effort == medium)	Discard
17	If (rel == high && comp == medium && effort == high)	Discard

Here rel represents the relevance of project attributes and hence coverage. Comp represents complexity of a particular technique. While forming the rules, the following observation were considered regarding select, discard and moderate for output variables:

1. technique will be selected only if relevance/coverage is high and will be discarded if rel is low irrespective of other two input variables. Hence, effort and complexity will be checked if rel is high.

2. if rel is medium that technique is moderate then other two input variables will be checked either for low or medium value to get the moderate value as the output variable. If any of the two input variables has high value that technique will be discarded.

Figure-1 and fig-2 represent the surface and the rules base view of the proposed system.

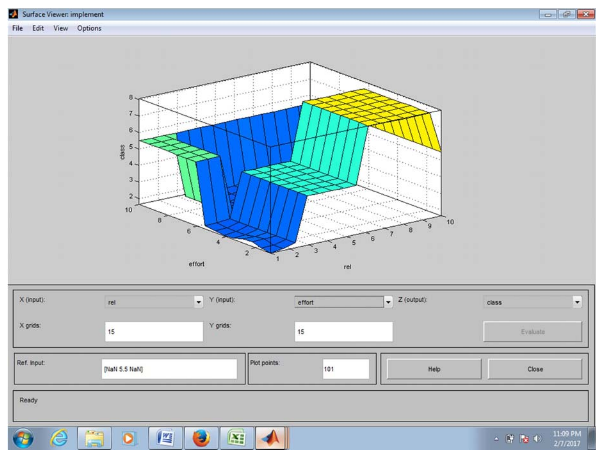

Fig. 1. Surface View of Effort and Coverage in FIS using MATLAB

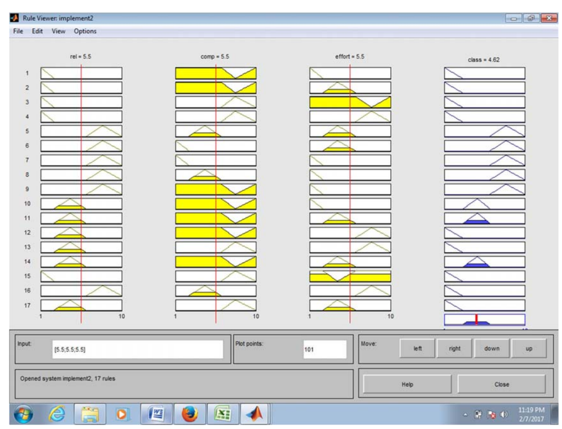

Fig. 2. Rule based View through MATLAB

The below algorithm shows the step by step procedure of the proposed approach.

Algorithm: class_TCP

Input(three inputs):
1. Relevance of selected TCP Techniques based on maximum requirement coverage.
2. Complexity of selected TCP techniques
3. Average Effort(AE)

Output:
Final_class : TCP Techniques

Begin

1. Identify input variables(linguistic variables) i.e. relevance, AEI, Complexity(initialization)
2. Mapping of fuzzy sets to input variables by constructing the membership functions (initialization)
3. Formation of rules to create the rule base (initialization)
4. Conversion of input data (fuzzification)
5. Assessment of available rules in the rule base (inference)
6. Merge all the results achieved from available rules(inference)
7. Mapping of output data (defuzzification)

End

III. EXPERIMENTAL STUDY

The validation of proposed approach is done by considering data set provided by industry people. The output is recorded and validated with the original data. Our proposed rule based fuzzy system has classified 14 out 15 instances correctly. Output of industry project data set output is shown in the figure-3. Further its accuracy is checked and compared with other classification techniques such as bagging, J48 and, KSTAR to prove the efficiency of rule formation in Fuzzy logic. Using Bagging algorithm, correctly classified instances are 11 out of 15 and hence accuracy is 73.333%. Using J48 decision tree correctly classified instances are 10 out of 15 and hence, accuracy is 66.6667% for industrial data set. Using KSTAR classifier, correctly classified instances are 7 out of 15 and hence accuracy is 46.67%.Undoubtedly, accuracy through fuzzy logic is 93.333% shows that is the best approach for classification.

>> b=readfis('implement2')	>> output=evalfis(a,b)
b =	output =
name: 'implement2'	1.8273
type: 'mamdani'	1.7962
andMethod: 'min'	8.0352
orMethod: 'max'	8.0164
defuzzMethod: 'centroid'	1.8861
impMethod: 'min'	3.6214
aggMethod: 'max'	4.6046
input: [1x3 struct]	2.8981
output: [1x1 struct]	8.0444
rule: [1x17 struct]	8.0362
	8.0287
	8.0431
	4.6180
	8.0440
	1.8282

Fig. 1. Results of FIS through MATLAB

The following line chart and bar chart clearly show the effectiveness of selection criteria for TCP technique. Weka is used as black box tool for the comparison of proposed approach to get the accuracy through confusion matrix.

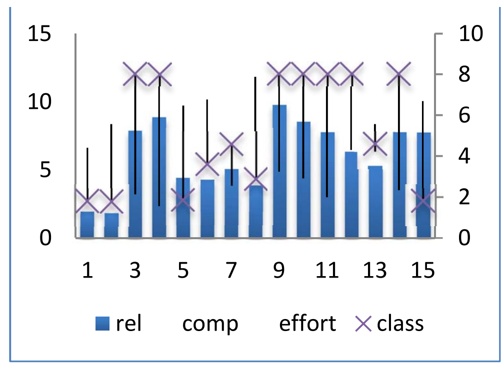

Fig. 2. Results of FIS with industry data

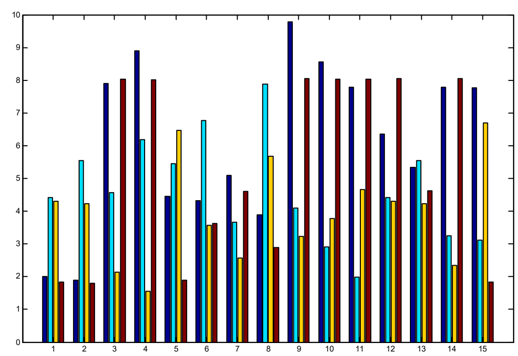

Fig. 3. Results of FIS through MATLAB

While calculating accuracy there was one point where this research has to relay on experts knowledge. There is one rule which adopted by considering experts' opinion as it always depends upon the clients. The rule is as follows:

"If rel is high and comp is medium and effort is medium class is select/moderate".

In general, it all depends on deadline and analysts' decision whether to increase the effort by increasing the cost or not. So, this rule based approach is silent for the above said rule and provides flexibility to various industries to add in later stages according to their current requirements. But for the validation purpose we are considering it as 'select' in first case and 'discard' for the second case. During the analysis of proposed approach with data set two records were found in accordance to above said rule. Hence, 12 records were considered as classified out of 15 in this case. To calculate accuracy of proposed approach with data set, two options were considered.

1. Accuracy is 80% while considering that data as misclassified and

2. Accuracy is 93.333% while considering that data as classified or ignored it.

The outputs of the proposed approach are three classes that provide the support for the selection of various existing TCP techniques.

IV. Conclusion

In this research we have proposed a novel based technique for the classification of TCP techniques using Fuzzy Logic. Fuzzy Rule based classification technique is finally compared with bagging and J48 decision tree and KSTAR classifier for its accuracy. The results over the other classification techniques clearly show that this technique is efficient in terms of accuracy as compared to other techniques.

Acknowledgment

We would like to thank Mr. Mohit Kumar (Sr. Project Manager) and Mr. Yogesh Gupta(Test Management Consultant) for providing assistance during survey and the validation of proposed approach.

References

[1] Ali M. Alakeel, "Using Fuzzy Logic in Test Case Prioritization for Regression Testing Programs with Assertions," Scientific World J, vol-2014, Article ID-316014, 2014.

[2] Arilo Claudio Dias Neto , Rosiane de Freitas Rodrigues , Guilherme Horta Travassos; "*Porantim-Opt:Optimizing the Combined Selection of Model-Based Testing Techniques*", Fourth International Conference on Software Test., Verif. and Validt. , 2011.

[3] C.Catal and D. Mishra, "Test Case Prioritization: A systematic mapping study, "Software Quality Journal, vol. 21, no. 3, pp. 445-478, Sep. 2013.

[4] Charitha Hettiarachchi, Hyunsook Do, Byoungju Choi, "Risk-based test case prioritization using a fuzzy expert system", In Journal: Information and Software Technology, Volume 69, pp 1-15, Jan. 2016.

[5] D. Silva, R. Rabelo, M. Campanhã, P. S. Neto, P. A. Oliveira and R. Britto, "A hybrid approach for test case prioritization and selection," *IEEE Congress on Evolutionary Computation (CEC)*, pp. 4508-4515, July2016.

[6] G.Duggal, B.Suri, "Understanding Regression Testing Techniques", COIT, 2008.

[7] Gary Yu-Hsin Chen, Pei-Qi Wang," Test Case Prioritization in Specification-based Environment", Journal of Software, Vol.-9,No.8, pp. 205-2064, August 2014.

[8] G. Rothermel, R.H. Untch, C. Chu, and M.J. Harrold, "Prioritizing Test Cases for Regression Testing," IEEE Trans. Software Eng., vol. 27, no. 10, pp. 929-948, 2001.

[9] G. Rothermel, R. H. Untch, C. Chu and M. J. Harrold, "Test Case Prioritization: An Empirical Study"; International Conference on Software Maintenance, Oxford, UK, IEEE, 1999. .

[10] Hema Srikanth, Laurie Williams, Jason Osborne; "*Towards the prioritization of System Test Cases"* Software Testing , Verification and Reliability 2014; 24:320-337; Wiley Online Library, June 2013.

[11] Krishnamoorthi Ramasamy, S.A.Sahaaya AruI Mary, "Incorporating varying Requirement Priorities and Costs in Test Case Prioritization for New and Regression testing", International Conference on Computing, Communication and Networking-IEEE (ICCCN 2008) 978-1-4244-3595-1/08

[12] M.Harman,S.A.Mansouri, and Y.Zhang, "Search based Software Engineering: Trends, techniques and applications," ACM Comput. Surv., vol. 45, no. 1,pp. 11:1-11:61, Dec. 2012.

[13] N. Sharma, Sujata and G. N. Purohit, "TCP techniques : an empirical study", *International Conference on High Performance Computing and Applications (ICHPCA),* Bhubaneswar, pp. 1-6, December, 2014.

[14] Praveen Ranjan Srivastava, "Test case Prioritization"; Journal of Theoretical and Applied Information Technology; pp:178-181, 2008.

[15] S. Elbaum, A. Malishevsky, and G.Rothermel, "Test case prioritization: A family of empirical studies", IEEE Transactions on Software Engineering,2002.

[16] Sujata and G. N. Purohit, "A Schema Support for Selection of Test Case Prioritization Techniques, *Fifth International Conference on Advanced Computing & Communication Technologies* (ACCT '15), pp. 547-551, 2015.

[17] Varun kumar, Sujata, Mohit Kumar, "Test case prioritization using fault severity", International Journal of Computer Science and Technology (IJCST), Vol-1, Issue-1, pp-67-71, 2010.

[18] Vegas, S.,Basili, V. , "A Characterization Schema for Software Testing Techniques", Empirical Software Engineering, v.10 n.4, p.437-466, October, 2005.

[19] Yoo, S. and Harman, M., Regression Testing Minimization, Selection and prioritization: a survey. Softw. Test. Verif. Reliab.,pp. 22: 67–120 2012.

[20] Zeeshan Anwar and Ali Ahsan. 2014, "Exploration and analysis of regression test suite optimization", SIGSOFT Softw. Eng. Notes 39,pp. 1-5, February 2014.

[21] Roger S. Pressman, Software engineering a practitioner's approach 6/e, 2005