International Journal of Scientific Engineering and Technology (ISSN : 2277-1581)
Volume No.3 Issue No.7, pp : 856-860 1 July 2014

Test Case Prioritization Based On Faults

R.Pradeepa[1] and Dr.K.VimalaDevi[2]
[1]Department of Computer Applications, Sri Ramakrishna Engineering College,Coimbatore, Tamilnadu, India.[2]Department of Computer Science, P.S.R. Engineering College, Sivakasi, Tamilnadu, India.
Email : [1]prathikrish.lect@gmail.com, [2]kvimaladevi@gmail.com

Abstract - ***Regression testing is one of the important process during the development stage, but it is the most critical activities of software development and maintenance. Whenever software is modified, the comparison between the new and old test case has to be done from the first, to avoid unwanted changes. It is impractical to re-execute all the test case if any changes in the program. Due to this time, cost and space is wasted, this problem can be overcome by prioritization. Test Case Prioritization (TCP) is a technique to schedule and execute the test case with higher priority, according to rate of fault detection. Main goal of TCP is 1. to increase the rate of fault detection, 2. to increase the effectiveness of test suites, 3. to decrease the cost of regression testing. The severe faults can be detected on the basis of different prioritization technique as prioritization, non-prioritization, random prioritization and selection prioritization basis with the help of Average Percentage of Fault Detected (APFD) metrics.***

Keywords - Regression Testing, Test case, Test Case Prioritization (TCP), Average Percentage of Fault Detected(APFD), Rate of fault detection.

I. INTRODUCTION

Software testing is an essential phase in software engineering to determine the quality of software products and its reliability. A test case is defined as "A set of test inputs, execution, and expected results developed for a particular objective, such as to identify an independent path or to verify the problem with a specific requirements. Test cases are created based on the requirements and executed to find faults in the software. A test suite is a set of test cases, which satisfies the customers' requirements. Executing all the test case is impossible because it may lead to high testing cost in terms of time, cost and memory.

Test cases are executed in order to meet their goals, (i.e) (a) to increase the rate of fault detection, (b) to achieve code coverage, (c) to increase the effectiveness of test suite, (d) to decrease the cost of regression testing. [iii] Dennis Jeffery *et.al*, experimentally proven that test suite can be reduced by selecting the prerequisite test cases.

Test case can be ordered based on some criterion like code coverage (path, function, condition, function, statement etc), requirement, fault , cost, time, severity, weight based, history-based, etc. Testers execute the higher priority test cases first , remaining the next. TCP technique never discard any test cases, so that the drawback of minimization technique is solved.

In Rothermal[i] and Elbum[ii] explored several prioritization techniques to increase the rate of fault detection. Varieties of testing criteria have been discussed and various criteria are useful for identifying the test cases that differentiate the structural and functional elements in the program. Through this number of faults can be identified more effectively.

To determine the effectiveness of fault , we have injected the faults in the fault matrix. Section II discusses a literature review of existing TCP techniques and prioritization process. Section III describes the test case prioritization. Section IV proposes research challenges measuring the rate of fault detection and about APFD metric. Section V proposes the experimental analysis with graph. Section VI discussed about conclusion and future work regarding TCP. Section VII proposed the acknowledgement. Section VIII represents all the references used in this paper.

II. RELATED WORK

Researchers have proposed various techniques to prioritize the test cases based on the faults. A general discussion from the existing survey is presented here in this section.

Rothermal et.al [i], compared the results of different test case prioritization techniques through his experiments. Faults can be detected on the basis of different prioritization technique as prioritization, non-prioritization, random, selection, optimal, total-branch coverage, additional-branch coverage, total fault exposing potential(FEP) and additional fault exposing potential prioritization are used. From the experimental study total FEP coverage based TCP performs better than rest of the technique. Additionally they have mentioned that the TCP process is required for software testing because: (a) the regression testing phase consumes lot of time and cost to execute, (b) there is a need to decide which test case to be executed first, and (c) there is not enough time or resources to execute the entire test suite.

Elbum et.al [ii], conducted a set of empirical studies that aim to find, (1) the effectiveness of prioritization techniques to specific modified version, (2) to find the trade-off between fine granularity prioritization techniques and coarse granularity prioritization techniques. The result analysis shows that version-specific prioritization can improve the rate of fault detection significantly.

Acharya et.al [vi], to minimize the cost and time of regression testing, they proposed a method to prioritize the test case. They have proposed model based prioritization method by considering the number of object Interactions per unit time as

the objective function. Here more importance is given to the number of inter component object interactions to maximize the faults, when the components interact each other.

Praveen Ranjan Srivastav [vii] has proposed a new TCP algorithm to compute the average faults discovered per minute. Calculating the effectiveness of prioritized and non-prioritized test cases using APFD metric.

R.Kavitha et.al [viii] have proposed an algorithm that performs the rate of fault detection and fault impact based prioritization of test cases. Experimental results shows that more effective severe fault identification at earlier stage of the testing process could be obtained by the proposed algorithm for prioritized test cases compare to unprioritized ones using APFD metric.

Siripong et.al [ix] conducted a set of experimental analysis to improve the ability to prioritize a set of test cases in case that there are multiple test cases with the same priority weight values, to improve the ability to automatically find duplicate test cases with the same values.

III. REGRESSION TESTING TECHNIQUE

Regression Testing: Regression testing is either full or partial selection of previously performed test cases which are re-executed to make sure the existing functionalities work adequate. This testing is complete to make sure that new code reworks should not have side effects on the accessible functionalities.

Regression Testing Techniques: Software maintenance is supposed to do an action that contains enhancements, error or bug corrections, optimization and deletion of existing features. These modifications may cause the system to work incorrectly. Therefore, Regression Testing becomes essential and can be carried out using following techniques:

Retest All : This is one of the processes for regression testing in which all the tests in the presented test suite ought to be re-executed. This is very exclusive as it needs a huge time and resources.

Regression Test Selection : Re-executing the entire test suite, it is superior to choose part of test suite. Test cases can be classified as 1)Reusable Test Cases 2) Obsolete Test Cases. Re-usable Test Cases can be used in succeeding regression cycles. Obsolete Test Cases can't be used in succeeding cycles.

Prioritization of Test Cases : This technique applies information about program and test suite to eliminate the test cases, which have become redundant with time, as new functionality is added. It is different from Regression test selection.

IV. PROPOSED PRIORITIZATION TECHNIQUE

A failure is a deviation of the expected behaviour of a program, caused by a fault. The most common prioritization goal is to increase the rate of fault detection. It means, test cases are executed in an order such that failures occur as early as possible in the testing process. The following prioritization techniques have been proposed in order to achieve the goal proposed.

Prioritization: This technique will execute the test case based on RFT values.

Non-Prioritization: This technique will execute the test case in sequence order.

Random : This is the most straight forward prioritization criterion, which orders test case randomly. Random permutations are used to control in many prioritization experiments [ii].

Selection : This technique will execute the test cases by selecting the independent path without redundancy.

APFD Metric: TCP is aimed to improve the effectiveness of regression testing through re-ordering the test cases to increase the fault detection ability. To solve this problem Rothermal et.al [i],has introduced APFD metrics. It helps to measure the weighted average of the percentage of faults detected over the entire test suite. The APFD values ranges from 0 to 100; higher values implies better fault detection rates.

Let T be a test suite containing n test cases, and let F be a set of m faults revealed by T.

$$APFD = 1 - \frac{TF1 + TF2 + \ldots\ldots + TFm}{nm} + \frac{1}{2n} \quad -- \; 1$$

Advantage: This technique will lessen the cost of validating, executing and managing test suites over upcoming software release.

Disadvantage: (1) Main problem of TCP is, capability of fault detection is compressed by reducing the test suite size . (2) APFD is possible only when prior knowledge of faults are available, hence used only for evaluation[vii].

Rate of fault detection : The average number of faults per minute by a test case is called rate of fault detection. The rate of fault detection of test case i have been calculated using # of faults by time taken to detect faults for each test case multiplied by 10.

RFTi = ((# of faults) / time) * 10

Every factor is converted into 1 to 10 point scale. The reason in earlier work [ii] may take long time depending on the size of test suite and test case execution time. In this paper new TCP technique that prioritize the test case with the goal of giving importance of test case which have higher value for rate of fault detection.

V. EXPERIMENTATION AND ANALYSIS

The experiment was conducted using simple project, it is tested both manually and automatically using the tools to execute all the test cases and to gather its execution time. We have injected 10 faults based on severity levels. We have noted the time taken to find out the faults by each test case has been shown in the Table 1.

Table 1: Fault Matrix – No. of faults with execution time.

Test case/ Fault	TC 1	TC 2	TC 3	TC 4	TC 5	TC 6	TC 7	TC 8	TC 9	TC 10
F1						*				*
F2	*			*				*		
F3			*				*		*	
F4				*				*		
F5						*				
F6						*				*
F7	*		*		*					
F8		*					*			
F9			*				*		*	
F10		*				*				
No. of Faults	2	2	3	2	1	4	3	2	2	2
Execution Time	12	9	9	14	11	10	14	8	13	10

Rate of fault detection of test cases TC1, TC2, …….TC10 respectively.

RFT1 = (2/12)*10 = 1.66

RFT2 = (2/9)*10 = 2.22

RFT3 = (3/9)*10 = 3.33

RFT4 = (2/14)*10 = 1.42

RFT5 = (1/11)*10 = 0.90

RFT6 = (4/10)*10 = 4.0

RFT7 = (3/14)*10 = 2.14

RFT8 = (2/8)*10 = 2.5

RFT9 = (2/13)*10 = 1.53

RFT10 = (2/10)*10 = 2.0

The comparison between the results of prioritized (using RFT values), non-prioritized, random and selection cases is done based on the results of APFD metric. APFD is the standardized metric that is used to find the degree of faults detected[4]. The comparative analysis is shown below in table 3 and Prioritize order of test cases based on the rate of fault detection values are,

TC6,TC3,TC8,TC2,TC7,TC10,TC1,TC9,TC4,TC5.

$$APFD = 1 - \frac{1+7+2+9+1+1+7+4+2+4}{10*10} + \frac{1}{2*10}$$

$$= 1-38/100+1/2$$
$$= 0.57$$

The Non – Prioritized order of test cases are
TC1,TC2,TC3,TC4,TC5,TC6,TC7,TC8,TC9,TC10

$$APFD = 1 - \frac{6+1+3+4+6+6+1+2+3+2}{10*10} + \frac{1}{2*10}$$

$$= 1-34/100+1/20$$
$$= 0.61$$

The Random order of test cases are
TC5,TC1,TC10,TC3,TC9,TC2,TC4,TC6,TC8,TC7

$$APFD = 1 - \frac{8+2+4+7+8+8+2+6+4+6}{10*10} + \frac{1}{2*10}$$

$$= 1-55/100+1/20$$
$$= 0.40$$

Computation Procedure for selecting test cases

Step 1: Select test case having maximum # of faults

TC6 : F1,F5,F6,F10 - TC6 is selected

Step2 : Compare the selected test case with other test case

TC10: F1,F6 - All faults covered within TC6, hence

TC10 is rejected

TC2 : F8,F10 - Fault F10 is covered within TC6, hence TC2 is

rejected

Step3 : Select the next test case having maximum # of faults – TC3

& TC7

TC3 : F3,F7,F9 - TC3 is selected based on minimum execution time

TC5 : F7- All faults covered within TC3, hence TC5 is rejected

TC10 : F3,F9 - All faults covered within TC3, hence TC10 is

rejected

TC7 : F3,F8,F9 - Fault F3 & F9 is covered within TC3, hence

TC7 is rejected

TC1 : F2,F7 - Fault F7 is covered within TC3, hence TC1 is rejected

Step4 : Select the next test case having maximum # of faults –

TC2,TC8 & TC4

TC8 : F2 - TC8 is selected based on minimum execution time

TC4 : F2,F4 - All faults covered within TC8, hence TC4 is rejected

TC1 : F2 - All faults covered within TC8, hence TC1 is rejected

Step5 : Select the next test case having maximum # of faults – TC2 & TC7

TC2 : F8 - TC2 is selected based on minimum execution time

TC7 : F8 - All faults covered within TC2, hence TC7 is rejected

Selected test case : TC2,TC8,TC3,TC6

Rejected test case : TC4,TC1,TC7,TC5,TC10

Table 2 : After the selection of test case

Test case/ Fault	TC2	TC8	TC3	TC6
F1				*
F2		*		
F3			*	
F4		*		
F5				*
F6				*
F7			*	
F8	*			
F9			*	
F10	*			*
No. of Faults	2	2	3	4

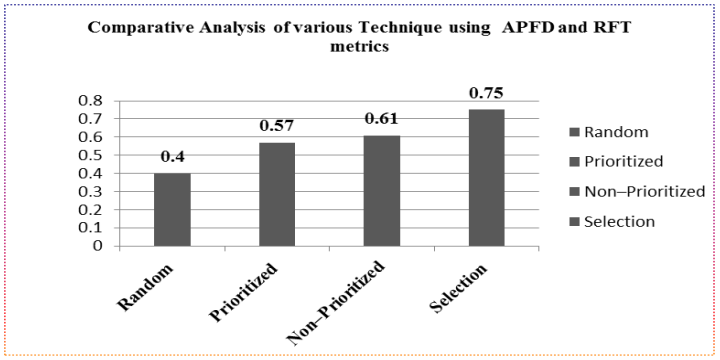
The selection order of test cases are
TC2,TC8,TC3,TC6

$$APFD = 1 - \frac{4+0+3+0+4+4+0+1+3+1}{10*10} + \frac{1}{2*10}$$
$$= 1-20/100+1/20$$
$$= 0.75$$

Table 3 : Comparative analysis

S. No.	Methods/ Techniques	Test case order	APFD values
1.	Random	TC5,TC1,TC10,TC3,TC9,TC2,TC4,TC6,TC8,TC7	0.40
2.	Prioritized	TC6,TC3,TC8,TC2,TC7,TC10,TC1,TC9,TC4,TC5	0.57
3.	Non–Prioritized	TC1,TC2,TC3,TC4,TC5,TC6,TC7,TC8,TC9,TC10	0.61
4.	Selection	TC2,TC8,TC3,TC6	0.75

Figure 1: APFD is higher for selection that reveal most faults early than others



Thus the selection based test case yields better fault detection than the other.

VI. CONCLUSION AND FUTURE WORK

This paper proposed an approach for TCP in order to improve regression testing time and cost. Comparative analysis is done for prioritized, non-prioritized, random and selection cases with the help of APFD metric. It is proven that selection of test case result is more efficient while comparing to the other techniques. In future TCP can be done by applying ANOVA and ANCOVA for more effectiveness.

VII. ACKNOWLEDGEMENTS

We are most grateful to all the people who helped us to compile and complete this work.. We would like to thank almighty for giving us strength to pull throughout this task and all the experts who gave their best impact towards this field of research.

VIII. REFERENCES

i. *Rothermal, Roland H. Untch, Chengyun Chu and Mary Jean Harrold, "Prioritizing Test Case for Regression Testing", IEEE Transactions on Software Engineering, 2001.*

ii. *Sebestian Elbum, Alexey G.Malishevsky and Gregg Rothermal, "Test Case Prioritization : A Family of Empirical Studies," IEEE Transactions on Software Engineering, 2002.*

iii. *Dennis Jeffery and Neelam Gupta, "Improving Fault Detection Capability by Selectively Retaining Test Cases during Test Suite Reduction", IEEE Transactions on Software Engineering, VOL 33 No.2,February 2007.*

iv. *Sahil Gupta, Haimanshi Raperia, Eshan Kapur, Harshpreet Singh and Aseem Kumar, "A Novel Approaches For Test Case Prioritization", International Journal of Computer Science, Engineering and Applications, Vol.2, No.3, June 2012.*

v. *Sanjukta Mohanty, Arup Abinna Acharya, Durga Prasad Mohapatra, "A Survey on Model Based Test Case Prioritization", International Journal of Computer Science, Information Technology, Vol.2(3), 2011, 1042-1047.*

vi. *Rimmi Saini, Satinder Saini, Deepa Gupta, Ajay Ranna, " Reduction in Test Case using Regression Testing approach and cost effective test prioritization testing techniques – APFD measures", International Journal of Science and Research Publications, Vol 2, Issue 7, July 2012.*

vii. *Praveen Ranjan Srivastava,"Test Case Prioritization", Journal of Theoritical and Applied Information Technology IEEE, 2008.*

viii. *R.Kavitha and N.SureshKumar, "Test Case Prioritization for Regression Testing based on Severity of Fault", International Journal of Computer Science, Engineering, Vol.2, No.5, 2010, 1462-1466.*

ix. *Siripong Roonguanguwan, Jirapun Daengdej, "Test Case Prioritization Techniques", Journal of Theoritical and Applied Information Technology, 2005-2010.*