Volume 4, Issue 3, March 2014 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

A Review on Testcase Prioritization Techniques

I. Jenifer
PG Scholar
Dept of Computer Science and Engineering
Sona College of Technology
Salem, India

A.K.Ilavarasi
Assistant Professor
Dept of Computer science and Engineering
Sona College of Technology
Salem, India

***Abstract*— Regression testing is the most important part in software development life cycle. Regression testing means if any modification made on the program lines it does not affect the other part of the software. During the regression testing a set of test cases and existing test cases are reused. But the regression testing is the time and cost consuming process. To overcome this problem the test cases are prioritized. This test case prioritization is done using several methods. This paper presents a survey on various test case prioritization techniques.**

***Keywords*— Test case, Prioritization, Regression Testing, Genetic algorithm, APFD, Code coverage**

I. INTRODUCTION

Test cases are referred as the collection of test specification, test procedures and test programs, that is developed by software engineers before designing and writing a piece of code. Here prioritization of the test case is important for regression testing. Because of prioritization selective test cases gets executed. Prem et al [9]presents the regression testing is one of the ways, in order to make sure that the modification made on the program lines does not affect the other parts of the software. During the regression testing all the test cases are again tested. Fig 1 shows that Regression testing has four methods that are(1) Rest all (2) Regression test selection (3) Test suite reduction (4) Test case prioritization.

.

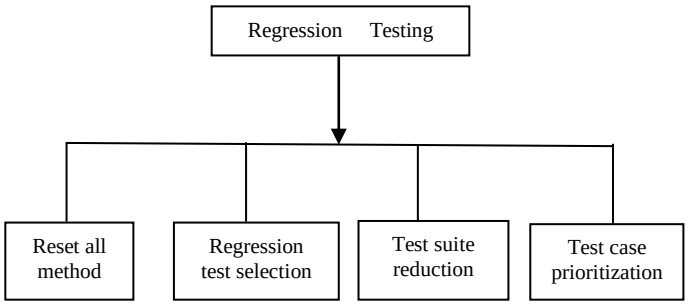

Fig1. Regression Testing

Reset all means the test cases that no longer apply to modified version of the programs are discarded and remaining set of test cases are used to reset the modified program. Regression test selection technique used to select the test cases for testing based on the information about modified program. Test suite reduction is different from regression test selection. This technique has the information about the program and removes the test cases. It is not permanently remove the test cases. But select which test cases are required. Hence it will be a time consuming process and expensive. To overcome this problem the developers incorporate prioritization of test cases for selective testing. Prioritization means scheduling (or) changing the order of test case execution, based on the factors (or) methods. Prioritization is used to identify the defects early and improve the effectiveness of regression testing. In this paper we reviewed the number of factors (or) methods used in test case prioritization in detail.

Sujatha et al [1] presents a technique, genetic algorithm is proposed to prioritize the test cases. The source code (or) binary code is used to prioritize the test cases, which is not available. The paper prioritizes the test cases completely based on requirements of the system and technique considers the maximum requirement coverage and potential of fault detection during the testing process. Severe fault test cases are executed first which are covering highly prioritized requirements. This approach is used to find the more severe faults, early in the testing process than the other random as well as fault based approach.

Test case selection and prioritization [2] provides an effective selection and prioritization of test cases based on the code coverage. This approach is used to reduce the cost and time for the regression testing. This approach has three techniques (1) Test case minimization (2) test case selection (3) Test case prioritization TCS algorithm for Test case selection and TCP algorithm for test case prioritization. In TCS algorithm test cases are grouped into 3 clusters (1) Out dated (2) Required (3) Surplus.

Out dated means the test cases which are covered by any statement, after removal of $SDEL_i$ from $TCC_{ij.}$ SDE_{Li} represents statement deleted in p. TCC_{ij} represents it is a metrics and test cases and their statement covered. It will be added in to the outdated. Modified statements which are not covered by the test cases, can be removed from TCC_{ij} and then can be added into surplus cluster. Rest of the test cases are into required cluster. Finally the required TCC_{ij} is smaller than the original TCC_{ij}. For prioritization, the output of the test case selection will be the input of the prioritization algorithm TCP_i. In this algorithm the number of statement is covered by the test cases are counted from the new (or) required TCC_{ij}. The highest numbers of statements that are covered by test cases are added into the test case prioritization vector $TCP_{i.}$ TCP_i vectors are removed from TCCij. Those steps are repeated until the statements are deleted.

II. RELATED WORK

The test case prioritizations are shown in fig 2.Test case prioritization techniques has [3] many approaches to schedule (or) ranking the test cases. But those approaches are failed to prioritize the multiple test suites, with same priority values. This paper proposes two different prioritization methods, to overcome these issues. The first method is used to prioritize the test cases with same priority (or) weights. The second method is used to effectively prioritize the multiple test suites. MTSP represents the method for multiple test cases. This system is used to improve the capability weighting algorithm. MTSPM proposed to prioritizes the test cases based on the factors. The factors are (1) defects (2) time (3) cost (4) complex factor (5) randomly. It represent the method for prioritizes the many test suites and a test case it is also prioritizes the test cases. It is also prioritizes the test cases based on the factors like (1) time &cost factors (2) randomly. In this paper the high priority reserve the effectiveness, size of the acceptable test cases is improved.

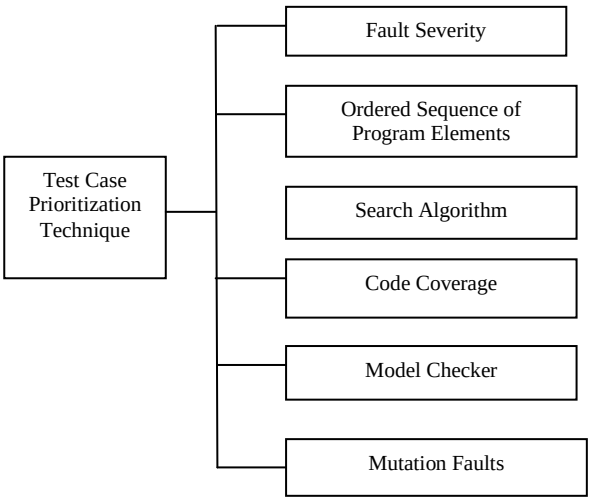

Fig 2. Test case prioritization techniques

Test case prioritization presents the [5] average faults per minute are calculated, based on the fault detection rate the test cases are prioritized. This paper presents a new algorithm to prioritize the test cases. In this algorithm average fault per minute is calculated as follows,

$$AF/m = \frac{F}{T_{cost}} - - - -(1)$$

Where F represents fault and T_{cost} represents as time. Based on that value the algorithm prioritizes the test case.

Value based regression test case prioritization [10] is used to detect the faults in early. It proposed an algorithm to prioritize the system test cases based on six factors like

(1) Customer priority
(2) Changes in requirement
(3) Implementation complexity
(4) Requirement traceability
(5) Execution time
(6) Fault impact.

The PSO (Particle Swarm Optimization) algorithm is used to assigning the values for factors and comparing the factors values. The highest score is maximum value and lowest score is minimum value. Add the factor value for all the test cases. If two test cases having the same factor value means the decision is taken by comparing requirement value of those test cases. So the PSO algorithm is more effective than the greedy algorithm in terms of time and cost.

Kumar et al [11] presents the paper based on the requirement the prioritization is done for the base of severity of faults. The total severity of faults detected (TSFD) is the summation of severity measures of all faults identified for a product is given in the below equation

$$TSFD = \sum_{i=1}^{i=n} SM - - -(7)$$

Where n represents the total number of faults identified for the product. Value driven test case prioritization for the requirement is used to improve the rate of fault detection [12].

Krishnamoorthi et al [17] Proposed new method to prioritize test case from software requirement specification to improve the user satisfaction based customer priority, changes in requirement, implementation complexity, completeness, traceability and fault impact.

System test case prioritization [18], [19] is proposed PORT approach used to prioritize system test cases based on customer priority, implementation complexity, fault proneness, and requirements volatility. The result shows that the PORT method improves the TCP than the random method.

Cognizant test case prioritization [23],[24],[25] is used to proposed historical value based approach for cost cognizant test case prioritization in which a historical value model is used to calculate historical value of test cases based on the historical test case cost and the historical fault severities.

Chunrong et al [14] presents the ordered sequence of program elements done in the basis of execution frequencies, with the help of this technique can detect the bugs in quickly. Farthest first algorithm selects the next test case which is farthest away from the already selected test suite. And this process will be repeated until all the test cases are reordered.

Kavitha et al [6] proposed to prioritize the test case based on the rate of fault detection and fault impact. The new algorithm is used to identify the severe faults at early stage. The proposed algorithm calculates the test case weight age.

$$Tcw = RFT + FI ----(2)$$

Where, Tcw represents test case weight age.

$$Tcw_i = RFT_i + FI_i ----(3)$$

Where, RTFi represents rate of fault detection average number of faults per minute by the test case is called rate of faults detection.

$$RTFi = \left[\left(number\ of \frac{faults}{time}\right) * 10\right] -- (4)$$

Where, FIi represents fault impact.

$$FIi = \left(\frac{Si}{\max(s)}\right) * 10 ----(5)$$

Where, Si represents value of the test case.

$$Si = \sum_{j=1}^{t} SV ----(6)$$

Max(s) represents high severity value. Based on the Tcw (Test Case weightage) the algorithm prioritizes the test cases. Results proved that the proposed prioritization technique is effective.

Test case prioritization using genetic algorithm [7] proposed the testing involves identifying the test case which can find the errors in program. It is difficult and time consuming process. This paper proposes a new technique to prioritize the test cases according to their capability of finding errors. More likely errors have been assigned a high priority. Less likely errors has been assigned low priority. This order will be generated by genetic algorithm. A genetic operator (selection, crossover, mutation) is applied on chromosome in order to find the fittest chromosome. Steps in genetic algorithm (1) generate population (2) Evaluate the fitness of generated population. (3) Apply selection for individual (4) Apply crossover and mutation (5) evaluation and reproduce the chromosome. For large number of time we will get a nearly optimized solution.

Suman et al [8] presents a new genetic algorithm is used to prioritize the regression test suite on the basis of complete code coverage. The main concept in genetic algorithm is (1) crossover & mutation. In this paper proposed two selection algorithms are (1) PMX (Partially Mapped Crossover) (2) Cyclic crossover. In this Cyclic crossover is more effective in terms of test cost optimization but less effective in terms of time. Genetic algorithm effectively used in terms of large number test cases and large number of possible test sequence.

Optimization [9] of the test cases is prioritized based on genetic algorithm to avoid the 80% of maintenance cost. In this test case which covers the most number of modified lines has the lowest priority and is executed last. This genetic algorithm is takes the input as information from regression testing and it produce the sequence of test case.

Amanjatain et al [4] mainly discussed various test case prioritization techniques for regression testing. The techniques are (1) customer requirement based technique (2) coverage based technique (3) cost-effective based technique (4) chronographic history based technique.

Customer requirement based technique referred as requirement factors are taken from customer and provide some weights. Based on the weights the test cases will be prioritizing. The coverage based technique us referred as quantity of the code covered by the test case. The amount of coverage is evaluated and it is used to prioritize the test case.ost effective based technique is referred as it prioritizes the test cases based on the cost factor. The test cases are prioritizes based on the prior execution of test case history is called chorographic history based technique.

Prakash et al [16] proposed a new potentially weighted method for test case prioritization. This technique prioritizes the test cases based on its coverage potential like as faults coverage, code coverage, branch coverage, path coverage, and

function coverage. And it is assigning weights for criterion. The test cases are prioritized based on its weight value from maximum to minimum. The main objective of code coverage testing is that every statement in the code has been executed at least once. The criteria value T_i^{cd} is calculated as

$$T_i^{cd} = \left(\frac{N_{cd}}{M_{cd}}\right) * 10 - - - -(8)$$

Where, N_{cd} represents the number of codes covered by the test case T_i. M_{cd} represents the maximum number of codes covered by any test case Ti.

The effective of the function coverage testing is that every function in the code has been executed at least once. Function coverage is calculated as

$$T_i^{fn} = \left(\frac{N_{fn}}{M_{fn}}\right) * 10 - - - -(9)$$

Where, N_{fn} represents the number of functions covered by the test case Ti. M_{fn} represents the maximum number of functions covered by any test cases T_i. This method is used to improve the effectiveness and reduce the time and cost.

The authors [20],[21],[22] provided the variety of test case prioritization techniques. Most of these techniques prioritize the test cases based on their coverage information such as statement coverage, branch coverage, loop coverage, function coverage condition coverage and fault detected.

Test case prioritization with [15] model checker has four prioritization methods like, coverage prioritization, FEP prioritization, property prioritization, and optimal prioritization. It is very important to re create test cases as models, which allows analysis, with concern to certain properties. It is easily done by basing the transition relation of all variables on a special state counting variable.

Do, Hyunsook et al [13] discussed the mutation testing comes from the idea of swapping the faults. This technique evaluates the capability of a test code changes into the software program and checking whether the test suite can detect those changes in the code.

III. THE METRIC FOR QUANTIFIES PRIORITIZATION

Many of the metrics are used to measure the prioritization of test cases. APFD (Average Percentage of Fault Detection) metric is one of important metric to measure the test cases prioritization based on the fault detection. Because the main aim of the prioritization is to early identifies the fault.

$$APFD = 1 - TF1 + TF2 + TF3 \ldots \frac{TFm}{m * n} + \frac{1}{2n} - (10)$$

Where, M represents number of faults represents number of test cases. TF_i represents the position of the first test in T. The APFD metrics provides better results than the other metrics. The result of the APFD is that prioritized test cases more effective than the non prioritized test cases.

IV. CONCLUSION

The paper provides a detailed review of various regression testing techniques, which mainly focuses on prioritization of test cases. Prioritization means to schedule (or) order the execution of the test cases. The various prioritization techniques are reviewed in detail. It improves the software quality and Rate of fault detection on the basis of requirement specification and changes in program code. Prioritization is used to reduce the time, cost and to improve the effectiveness on the basis of code coverage and factors value. Test case prioritization technique will be chosen based on the user needs.

REFERENCES

[1] Sujata, Mohitkumar, Varumkumar "Requirement based test case prioritization using genetic algorithm" IJCST vol .1, Issue 2, December 2010.

[2] R.Beena, S.Sarala "code coverage test case selection and prioritization" IJSEA vol.4, no.6, November 2013.

[3] Siripong Roongruangsuwan, Jirapum Daengdej, "Test case prioritization techniques" Journal of theoretical and applied information technology, 2005-2010.

[4] Amanjatain, garima Sharma "A Sytamatic review of techniques for test case prioritization", International journal of computer applications(0975-8887),Volume 68-no.2,april 2013.

[5] Praveen Ranjan Srivastava "Test case prioritization", Journal of theoretical and applied information technology, 2005-2008.

[6] R.Kavitha, N.Sureshkumar "Test case prioritization for regression testing based on severity of fault", IJCSE vol.02, no.05 2010, 1462-1466.

[7] Ruchika malhotar, abhishek bharadwaj "Test case prioritization using genetic algorithm" IJCSI: 2231-5292, vol-2, Issue-3, 2012.

[8] Suman, Seema "A genetic algorithm for regression test sequence optimization" IJARCCE vol.1, Issue 7, September 2012.

[9] T.prem Jacob and T.avi "optimization of test cases by prioritization" journal of computer science9 (8):972-980 2013.

[10] E.Ashraf, A.Rauf and K.Mahmoat "Value based regression test case prioritization"WCECS 2012, October 24-26 2012.

[11] kumar, Dr Varun,"Sujata and M.kumar,"Test case prioritization using fault severity""IJCST 1, no.1 (2010):67-71.
[12] Srikanth, hema, Laurie Williams, and Jason Osborne, "System test case prioritization of new and regression test case". In empirical software engineering, 2005.2005 international symposium on pp, 10-pp IEEE, 2005.
[13] Do, Hyunsook, and Gregg Rothermel,"on the use of mutation faults in empirical assessments of test case prioritization technique "Software engineering, IEEE transaction on 32, no.9 (2006):733-752.
[14] Wu, Kun, chunrong fang, zhenyu Chen, and zhihong Zhao. "Test case prioritization in corporating ordered sequence of program elements "in automation of software test (AST), 2012 7 the international workshop on, pp 124-130.IEEE, 2012.
[15] Fraser, Gordon, and franz wotawa. "Test case prioritization with model-checkers". In 25th conference on IASTED international.2007.
[16] Prakash.N, Rangaswamy "Potentially weighted method for test case prioritization" JCIS 9:18(2013) 7147-715
[17] Krishnamurthy Rand Sahaaya Arul Mary S.A, 2009, "Factor Oriented Requirement Coverage Based System test case prioritization of new and regression test cases", International journal of information and software technology, Elsevier, no.53, pp.7999-808.
[18] Hema Srikanth, Laurie Williams, and Joson Osborne, 2005, "System test case prioritization of new and Regression testcases", IEEE.
[19] Hema Srikanth and Sean Banerjee, 2012, Improving test Efficiency through System test prioritization, the journal of systems and software, Elsevier, vol., no., pp.1176-1187.
[20] G.Rothermal, H.Roland.H Untch, Mary Jeean Harrold,"Prioritizing test cases for Regression Testing', IEEE Transaction on softer engineering, vol.27, no.10, October2001.
[21] S.Elbaum, a.Malishevsky, G.Rothermel, "Test case prioritization: A family of Empirical Studies" IEEE transaction on software engineering, vol.28, no.2, February2002.
[22] G.Rothermel, R.Untch, C.chu, M.harrold, "Test case prioritization: an empirical study," pric. International conference on software maintenance, pp.179-188, Aug.1999.
[23] H.park,H.Ryu,J.Baik, "Historical value based approach for cost cognizant test case prioritization to improve the effectiveness of regression testing", 2nd international conference on secure system integration and reliability improvement,pp.39-46,july2008.
[24] P.Srivastava "permance evaluation of cost cognizant test case prioritization", UACEE international journal of computer science and its applications, volume 1, issue 1.
[25] A.Khalilian, M.A.Azgomi, Y.Fazlalizadeh,"An improved method for test case prioritization by incorporating historical test case data, "Science of computer programming, volume 78, issue 1, 1 November 2012, pages 93-116.

