ISSN:0975-9646

Literature Review of Model Based Test case Prioritization

Shweta A. Joshi [1], Prof. B.S. Tiple[2]

[1]*Master of Computer Engineering student, MAEER's MIT, University of Pune- 411038, Maharashtra, India*

[2]*Associate Professor, MAEER's MIT, University of Pune- 411038, Maharashtra, India*

***Abstract*— Software testing always "Performing Verification and Validation of the Software Product" for its correctness and accuracy of working. Every time it is not possible to perform each and every test case. Hence it is important to decide test case prioritization.**

The major goal of Test case prioritization is to prioritize the test case sequences and finding the faults as early as possible to improve efficiency of testing. In Component Based Software Development (CBSD) software are implemented using reusable components either they may be in-built components or third party components. By gathering all these different components final software product is constructed. In CBSD, this whole process takes less time and works rapidly. As tester is having limited access to source code of reusable components so tester can face many difficulties. In such situations test cases are ordered as per priority which can improve performance of software.

***Keywords*— CBSD, Test Case, Test Case Prioritization.**

I. INTRODUCTION

As software time to market becomes shorter day by day, there is significant increase in the use of commercial off the shelf components. Component-Based Software System (CBSS) can run properly and effectively. Component-Based Software Development (CBSD) approach builds software systems by assembling pre-existing components under well-defined architecture which brings high reusability and easy maintainability to the component, and reduces its time-to-market. Therefore the productivity of software systems is improved and the development cost is also reduced. This paper is organized as follows. In section I, it gives an introduction to the Component-Based Software Development process. Section II contains the related research papers used for the survey process. Section III concludes the paper and the future work is present in the last section.

II. LITERATURE REVIEW

In last few years there were many publications discussed the concept of component based testing. In this section all these work is discussed and relates to the proposed work later on.

A. Code Based Test Case Prioritization

Test case prioritization based on system's source code is also known as code based prioritization.

1. Test Case Prioritization [1]

This paper defines how to prioritize the test cases according to the use of APFD (Average percentage of Faults detected) value. He suggested a new technique which is able to calculate the average number of faults found per minute by a test case and using this value it sorts the test cases in decreasing order. As Stated, APFD value is determined for both the prioritized and non-prioritized test suite and it is shown that the APFD value of prioritized test suite is comparatively higher than the non-prioritized test suite. Problem occurred with this technique was calculation of APFD is only possible when prior knowledge of faults is available.

2. Test Case Prioritization: An Empirical Study [2]

Whereas Rothermel et al. worked with the test case prioritization by investigating the nine prioritization techniques. These techniques stated as:

T1: No prioritization.
T2: Random prioritization.
T3: Optimal prioritization.
T4: Total branch coverage prioritization.
T5: Additional branch coverage prioritization.
T6: Total fault-exposing-potential (FEP) prioritization.
T7: Additional fault-exposing-potential (FEP) prioritization.
T8: Total statement coverage prioritization.
T9: Additional statement coverage prioritization.

The analytical results shown that the prioritization techniques can increase the fault detection rate of test cases and finally observed their relative work to perform fault detection quickly. While FEP-based prioritization techniques are not much cost-effective.

3. Efficient Test case Prioritization in Regression Testing [3]

After working on suggested issues Prashant et al. [3], presented his work to provide the regression test framework which sequences the test cases based on code coverage. First the test suite is selected then sequencing is done based on some criteria. By using this approach Testers marked the affected portion of code with help of business analyst and development team and select the test cases.

The study shows that the prioritized test cases achieve greater coverage in earlier execution phase than the non-prioritized test cases but percentage code coverage decreases as the execution moves in case prioritized test suite. In non-prioritized test suite execution, it varies as per the period and depends on the test cases executed during that period. Also results may vary if prioritization is done using different criteria like methods, block, classes or its combination.

4. Search Algorithms for Regression Test Case Prioritization [4]

Performing some outcomes, Li et al, focused on test case prioritization techniques for code coverage, which involves block coverage, decision (branch) coverage, and lastly statement coverage. Finally each code based test prioritization must have knowledge of

- Total Statement Coverage
- Total Function Coverage
- Additional Statement Coverage
- Additional Function Coverage

B. Model Based Test Case Prioritization

5. Experimental Comparison of Code Based and Model Based Test prioritization [5]

After discussing problems faced by code based prioritization techniques Korel et al. performed a small evaluation test to verify efficiency of both simple code-based and model-based test prioritization techniques. The target of this experiment was to evaluate these methods to check performance of early fault detection in the modified system. This analysis result set shown that model based test prioritization may improve the early fault detection as compared to the code based test prioritization because the execution of the model is very fast as compared to the execution of the actual system. Therefore, execution of the model for the whole test suite is cheaper as compared with code based test case prioritization.

6. Application of System Models in Regression Test Suite Prioritization [6]

After performing some experiments again Korel et al, prioritized the test cases by using several model-based test prioritization heuristics. It had few issues that selective model based prioritization considers only the number of executions of marked transitions which does not have a significant influence on the improvement of the early fault detection.

C. Requirement Based Test Case Prioritization

7. Requirements-Based Test Case Prioritization [7]

After facing some problems with model based prioritization Srikanth et al.[7], invented a new prioritization scheme with three main studies: defining faults as early as possible, to increase the software quality and to derive and utilize the minimum dataset of PORT(Prioritization of Requirements for Testing) PFs (Prioritization Factors) which can be used to efficiently for test case prioritization. It focuses on variety of issues like;

(a) If requirements have high complexity then it leads to maximum number of faults,

(b) If requirements volatility problem is there, then it needs re-design, or some kind of modifications of relative requirements, which tends to project risks, also increased bugs that results in project failures.

8. Component Interaction Graph: A new approach to test component composition [8]

In this paper Acharya et al. generated invented test kind of cases for testing components in component composition technology. Acharya defined one methodology by which components interact with each other using a Component Interaction Graph (CIG). For representing CIG it is mandatory to declare state chart diagram as input nothing but it needs to work with behavioral aspect of the system.

9. Techniques for testing component-based software [9]

After performing some experiments Wu et al. proposed that testing is realized using a Component Interaction Graph (CIG) by which the interactions and the dependence relationships among components can be elaborated. Major problem occurred with model dependence- based test prioritization that it requires multiple state transitions while doing interactions among all components in system.

10. A Model based prioritization technique for component based software retesting using UML state chart diagram [10]

This paper proposes new technique of prioritizing test cases which takes CIG as input along with the old test cases and also author have to consider the total number of state changes and total number of database access i.e. direct and indirect access encountered due to each test case. The existing model based test prioritization methods can only be used when models are modified during system maintenance.

III. CONCLUSION

For prioritizing test cases there are many algorithms which present a new approach for testing the component-based software systems. The studies show that testing of component-based software systems is necessary yet expensive. Here more importance is given to component interactions because maximum defect occur when components are going to interact with each other. This approach is mainly applicable to test the component

composition in case of component based software maintenance.

Finally our algorithm is found to be very effective in

- Maximizing the objective function.
- Minimizing the cost of system retesting.

Future Challenges

In future we can extend this work as; the system can use regression testing it can be done by testing interconnected test cases (group of paths) instead of testing every single test case , which might prove little complex but less time consuming.

IV. ACKNOWLEDGEMENT

I feel immense pleasure while presenting this work and am very thankful to my guide ***Prof. B.S. Tiple***, Dept. of Computer Engineering of *Maharashtra Institute of Technology, Pune* for her valuable suggestions and support.

I would like to express my sincere thanks to our senior professor and M.E. Coordinator ***Prof. Mangesh Bedekar*** for teaching me the fine points which are helpful for this completing work. I am also thankful to our Principal of MIT, Pune **Dr. L. *K. Kshirsagar*** for his consent to go forward with this topic.

VII. REFERENCES

[1] P. R. Srivastava, "Test Case Prioritization," Journal of Theoretical and Applied Information Technology, JATIT 2008.

[2] G. Rothermel, R.H.Untch, C.Chu, M.J.Harrold, "Test Case Prioritization: An Emperical Study," in Proceedings of the 24th IEEE International Conference Software Maintenance (ICSM '1999) Oxford, U.K, September 1999.

[3] P. Malangave, D. B. Kulkarni, "Efficient Test case Prioritization in Regression Testinng".

[4] Z.Li, M. Harman, and R. M. Hierons," Search Algorithms for Regression Test Case Prioritization," IEEE Transactions on Software Engineering, Vol. 33, No. 4, April 2007.

[5] B. Korel, G. Koutsogiannakis, "Experimental Comparsion of Code Based and Model model Based Test prioritization," IEEE 2009.

[6] B. Korel, G. Koutsogiannakis, and L.H.Tahat, "Application of System Models in Regression Test Suite Prioritization," in Proceedings of the 24thIEEE International Conference Software Maintenance (ICSM '08) pp.247- 256, 2008.

[7] Hema Srikanth and Laurie Williams, "Requirements-Based Test Case Prioritization," North Carolina State University, ACM SIGSOFT Software Engineering, pages 1-3, 2005.

[8] A. A. Acharya, S. K. Jena, "Component Interaction Graph: A new approach to test component composition," Journal of Computer Science and Engineering, Volume 1, Issue 1, May 2010.

[9] Y. Wu, D. Pan and M. Chen, "Techniques for testing component-based software," In Proceedings of the 7th IEEE International Conference on Engineering of Complex Computer Systems. pp. 222-232, 2001.

[10] Sanjukta Mohanty, Arup Abhinna Acharya, Durga Prasad Mohapatra, "A Model based prioritization technique for Component based software retesting using UML state chart diagram". 2011 IEEE Third Int'l Conf. on Electronics Computer Technology.

[11] Sanjukta Mohanty, Arup Abhinna Acharya, Durga Prasad Mohapatra, "A survey on model based test case prioritization" International Journal of Computer Science and Information Technologies, Vol. 2 (3) , 2011, 1042-1047.

[12] G. Rothermel, R. Untch, M. Harrol, "Prioritizing Test Cases For Regression Testing," IEEE Transactions on Software Engineering, volume 27, No. 10, pp. 929-948, 2001.

[13] Arup Abhinna Acharya and Sisir Kumar Jena, **"**Component Interaction Graph: A new approach to test component composition" Vol 2(3), 2011, 1042-1047.