A study on the analysis of the effect of Testcase Prioritization Methods

Dr. Priti[1], Kavita Dahiya[2]
Assistant Professor, Computer Science and Application, MDU, Rohtak
Research Scholar, Computer Science and Application, MDU, Rohtak

Abstract : Software testing is the important process of software development. Software testing is performed to analyze the functional capabilities and limitation of software system. Software testing is performed by performing various testcases in a specific sequence. Test sequence generation is the method to decide the sequence of testcase execution. Prioritization method decides the aspect, objective and significance of software testing. In this paper, the effect of different type of prioritization methods is provided on software testing. The elements and the functional description of fault, dependency and history based methods are provided in this paper.

Keywords : Testcase Prioritization, Blackbox Testing, Regression Testing, Sequence Generation

I. INTRODUCTION

The prioritization is the important process used for the optimization of testing process. The efficiency, resource utilization and reliability of the software testing is improved using prioritization methods. The prioritization methods are associated with blackbox testing to identify the order of test case implementation that can give the maximum benefit. The benefit is in terms of software reliability, time and cost effectiveness. Once the test-cases are prioritized, it becomes easy to set the order of testcase execution. The prioritization is basically dependent on some aspect or prospective such as fault, module-dependency, history etc. After setting the objective and aspect, the next work is define some measure to perform the estimation. The evaluation and computation of the test cases is performed based on these measures. Based on these measures, the method is defined to identify the sequence of test case execution[1][2]. The role of prioritization within the testing and test sequence generation is shown in Figure 1.

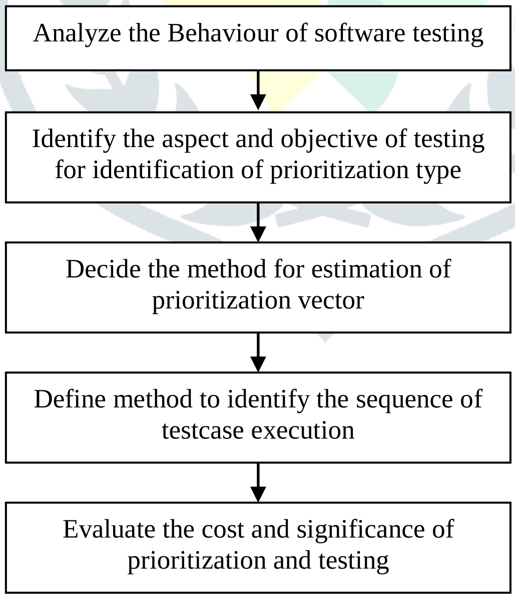
Analyze the Behaviour of software testing
Identify the aspect and objective of testing for identification of prioritization type
Decide the method for estimation of prioritization vector
Define method to identify the sequence of testcase execution
Evaluate the cost and significance of prioritization and testing

Figure 1 : Prioritization in Test Sequence Generation

Figure 1 shows that the prioritization is used effectively as the inclusive stage in test sequence generation. For, this at first the testing behavior is analyzed. The type of testing, software modules and environment are identified before applying the prioritization. The objective and requirement is also identified based on the application and environment. Based on these factors, the prioritization aspect is decided. After deciding the main aspect, the particular measure is defined for computation of that factor. After taking the decision of prioritization measure, the method is applied for generation of test sequence. Later, the analysis is done to identify the significance and effectiveness of prioritization method.

In regression testing, the prioritization methods are sensitive respective to the identification of the changes occur in the module. The prioritization aspect and measures are also decided while mapping the testcases to the modules. The dependency of the modified modules also considered as the effective prioritization method. The effect of the change module on other module is also identified while deciding the priority of the test cases. The relation between the module, changes in the module and the test cases is an important factor in regression testing[3]. The regression testing is considered as the continuous process that always happen some changes in the software system are performed. The generation of new version or the module level changes are identified as the continuous regression testing[4]. The prioritization methods are required each time when the regression testing is performed to handle these continuous changes in the software. The continuous regression testing and prioritization is the important functional behavior used in the industry. In industry, the continuous observation is carried out to identify the test cases with good prioritization factor. The selection of the effective prioritization method is also a challenge. The prioritization method can be manual or automated. The decision on the prioritization aspect, measure and the combination is also a challenge.

In this paper, a study on the prioritization methods is provided with its significance on testing behavior. Various prioritization techniques with their impact are discussed in this paper. In this section, the scope and behavior of prioritization method is provided. The role of prioritization in test sequence generation is provided in this section. In section II, the work of the researchers is defined on test sequence generation and prioritization methods. In section III, some of the common prioritization methods and behaviours are defined. In section IV, the conclusion of work is provided.

II. RELATED WORK

Regression Testing is a model based testing technique that reduces the cost and time of software development. Regression testing techniques are applied in structured and stage based form. These stages include test case selection, test data selection, test case prioritization and test sequence generation. Earlier researchers worked on each of the stages separately in with different constraint, method and attribute to optimize the regression test procedure. In this section, the contribution of earlier researchers is discussed.

Dini et al.[21] has explored the effect of test suits on regression testing. Author designed two methods for test case generation called feedback directed technique and search based technique. The dependency tracked method was considered by the author to optimize the testing behavior. Priyanka et al.[22] verified the effectiveness of testing under domain accountability. A novel method called Revised Analytical Hierarchy Process was applied by the author prioritizes the test case under verification aspect. The control case and coverage case prioritization was applied by the author for effective test case selection. Dhareula et al.[23] provided a exploratory study on test case selection method under code based and requirement based techniques. Author also identified the various attributes for test case reusability in these two categories of testing. Akimoto et al.[24] used the extended differential control flow graph for test case selection. The complexity based evaluation was applied by the author for promising test case selection. Dahiya et al.[25] setup a tool to categorize the test cases under reusable, retestable, obsolete and new test case categories. The activity diagrams were observed under semantic checks to relate the test cases. The sequence diagram based validation was applied by the author to improve the accuracy of the work process. Fawzy et al.[26] used the code coverage analysis for test case selection by avoiding the dead lock situation. The transition time observation was taken by the author to reduce the processing time.

Kandil et al.[27] presented an agile based automated regression testing process to expose the number of test cases. Author also applied the weighed test parameters for test case prioritization. The user aspect involvement was considered by the author to reduce the cost and to improve the capability rate. Mohapatra et al.[28] applied a study to identify the redundant test cases to improve the performance of regression testing. Author used the genetic technique with varying chromosome length to reduce the test cases in the test suit. Gao et al.[29] proposed a prioritization assisted ACO method to generate the fault avoided test sequence. The test case prioritization was done based on fault severity, execution time and fault count parameters. The metric regulation based ACO method was applied to generate the preserved test sequence. Ekelund et al.[30] developed a difference engine to analyze the outcome of previous runs and based on the recommendation the regression test case is selected. In this tool, the history of the tool, its access and impact were analyzed. The software system correctness and the results were analyzed to identify the effective test cases. Chauhan et al.[31] presented a test case selection method based on object dependency analysis. The activity graph processing was made to evaluate the dependency and code changes.

Konsaard et al.[32] used the prioritization based test case selection to achieve the maximum code coverage. The concern is defined to achieve the fault preserved and code coverage adaptive test case selection. This prioritization method was applied on genetic model for test sequence generation. Saha et al.[33] applied a likelihood dynamic mapping to assign the priorities to test cases based on fault analysis. A query featured model was generated by the author to minimize the score between the program and test cases. Akimoto et al.[34] applied a Differential control flow graph technique for setting up the structural relation between the modules and the test cases. The dependency metrics and the complexity evaluation were considered to assign the priority. The code change observation was also considered as other criteria for test case prioritization. Dobuneh et al.[35] defined a hybrid criteria for assigning the priority to test cases and to reduce the fault detection rate.

III. PRIORITIZATION METHODS

Various prioritization methods[14] were used by the researchers to optimize the sequence of testcases. Multiple criterias[17] were defined by the authors for improving the prioritization and the test sequence generation. In this section, the different aspects and types of prioritization methods are defined with relative impact on test sequence generation.

A) Fault based Prioritization

Fault is one of the effective measure that defines the quality and reliability of software. The fault occurrence, its frequency and the criticality are the major associated aspects that affect the reliability of the software system. Kayes et al.[5] used the fault information for prioritization of test cases for regression testing. The fault dependency and fault-rate were observed by the author within the regression testing. The method is capable to detect the dependency between the faults. Author analyzed the work based on the ratio of the faults exist in the software system and observed the fault dependency. The metric was able to observe the fault severity and analyze the execution time of test cases. The author showed that the fault and fault dependency can impact the reduction in software reliability and increase the execution time of test cases. Farooq et al.[6] used the fault based prioritization within the regression testing and mutation testing. Author applied the mutation testing to assign the priorities to the test cases. The fault rate was used by the author as effective measure for deciding the sequence of test case execution. The resource level and coverage based method was applied for optimizing the test sequence. The mutation based prioritization method assigned the higher priorities to the test cases wither higher faults and coverage. Yadav et al.[11] used the fuzzy logic for detection of software fault for prioritization of test cases. This method improved the fault detection rate and reduced the execution time and improved the requirement coverage. Wang et al.[12] used the fault severity as the measure for prioritization of test cases. The method categorized the fatal, serious and general faults in the system. The analysis was performed to identify the fault existence in software system and detection in test cases effectively. The method improved the efficiency of regression testing.

B) Module Coverage and Dependency based Prioritization

The coverage and module dependency is another aspect of test-case prioritization. The coverage information includes statement, branch and method level coverage. The static and dynamic methods are available to estimate the coverage of the testcases. The coverage is the important factor to identify the priority of the testcases. The cost span was also analyzed by the author based on the dynamic and static coverage. The coverage criteria based analysis was performed by the author for evaluating the test method and class. Author identified that the coverage information captured in early version is less effective in prioritization stage[7]. Huang et al.[9] defined the abstract prioritization method in four categories. These categories are interaction coverage, input=model mutation, non-information and similarity based prioritization method. Author improved the fault detection rate as compared to other prioritization methods. Kaushik et al.[10] observed the coverage based dynamic prioritization methods for effective estimation of testcases. Author performed the software artifact based analysis with constraint specification. The industrial changes were also defined by the author. The test sequence scheduling with reordering method was defined for improving the reliability of software system. nARDO ET AL.[18] provided the coverage based prioritization method for real-world system. Author analyzed the various coverage criteria respective to the industry. The finer grained coverage criteria method not enhances the fault detection rate.

D) Requirement based Prioritization

Requirement is the factor that defines the analysis respective to the user and customer prospective. The application, process and industry level requirements are observed in this prioritization method for estimation of effective test sequence. Kavitha et al.[8] used the requirement based analysis for prioritization of testcases. Author used the user and customer satisfaction as the measure for software quality. Author used three main factors called customer priority, requirement change and implementation complexity. The results show that the method has improved the reliability and fault detection ratio in the software. Salem et al.[15] proposed the requirement analysis model for improving the effectiveness of test sequence generation and modeling of test coverage. Author identified the functional requirement with behavior tree representation and derived the prioritized test cases. The method was defined to satisfy the industry need and achieved the sensitive accuracy. Arafeen et al.[20] used the requirement based clustering method for prioritization of testcases. The requirement information was analysis with fault specification for observing the earlier code. The clustering based method improved the detection of faults and improved the reliability in test sequence generation.

C) History based Prioritization

History based prioritization methods are defined by the researchers to provide the reference based prioritization. It means, some software is already prioritized and the testing is performed over it. In a new version or software, same approach can be applied. Lin et al.[13] provided a study on version specific ongoing software development and its prioritization in the real environment. Author proposed the version aware prioritization approach for taking the observations from existing software. Author achieved the better fault detection rate and improved the prioritization of the test suite. This method improved the accuracy of fault detection in the system and improved the software reliability. Noguchi et al.[16] used the ant colony based analysis method to collect the history information and used it to improve the effectiveness of test sequence generation. Diverse feature based analysis was defined for reducing the cost of software testing. Cho et al.[19] used the fault history based analysis method for prioritizing the test cases. The correlation data based analysis method is defined for improving the effectiveness of test sequence generation.

IV. CONCLUSION

Software testing is responsible to identify the software errors and to deliver robust and reliable software. Software testing is performed by generating the effective and significant testcases. After generating the testcases, the sequence of test case execution is another challenge. The prioritization methods are responsible to identify the sequence of test case execution. In this paper, various prioritization methods are defined and their scope and impact is discussed on software testing and regression testing.

REFERENCES

[1] C. Malz, N. Jazdi and P. Gohner, "Prioritization of Test Cases Using Software Agents and Fuzzy Logic," 2012 IEEE Fifth International Conference on Software Testing, Verification and Validation, Montreal, QC, 2012, pp. 483-486.
[2] W. Liu, X. Wu, W. Zhang and Y. Xu, "The research of the test case prioritization algorithm for black box testing," 2014 IEEE 5th International Conference on Software Engineering and Service Science, Beijing, 2014, pp. 37-40.
[3] M. H. Mahmood and M. S. Hosain, "Improving test case prioritization based on practical priority factors," 2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS), Beijing, 2017, pp. 899-902.
[4] D. Marijan, A. Gotlieb and S. Sen, "Test Case Prioritization for Continuous Regression Testing: An Industrial Case Study," 2013 IEEE International Conference on Software Maintenance, Eindhoven, 2013, pp. 540-543.
[5] M. I. Kayes, "Test case prioritization for regression testing based on fault dependency," 2011 3rd International Conference on Electronics Computer Technology, Kanyakumari, 2011, pp. 48-52.
[6] F. Farooq and A. Nadeem, "A Fault Based Approach to Test Case Prioritization," 2017 International Conference on Frontiers of Information Technology (FIT), Islamabad, 2017, pp. 52-57.
[7] J. Zhou and D. Hao, "Impact of Static and Dynamic Coverage on Test-Case Prioritization: An Empirical Study," 2017 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW), Tokyo, 2017, pp. 392-394.
[8] R. Kavitha, V. R. Kavitha and N. Suresh Kumar, "Requirement based test case prioritization," 2010 INTERNATIONAL CONFERENCE ON COMMUNICATION CONTROL AND COMPUTING TECHNOLOGIES, Ramanathapuram, 2010, pp. 826-829.
[9] R. Huang, W. Zong, D. Towey, Y. Zhou and J. Chen, "An Empirical Examination of Abstract Test Case Prioritization Techniques," 2017 IEEE/ACM 39th International Conference on Software Engineering Companion (ICSE-C), Buenos Aires, 2017, pp. 141-143.
[10] N. Kaushik, M. Salehie, L. Tahvildari, S. Li and M. Moore, "Dynamic Prioritization in Regression Testing," 2011 IEEE Fourth International Conference on Software Testing, Verification and Validation Workshops, Berlin, 2011, pp. 135-138.
[11] D. K. Yadav and S. Dutta, "Test case prioritization technique based on early fault detection using fuzzy logic," 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, 2016, pp. 1033-1036.
[12] Y. Wang, X. Zhao and X. Ding, "An effective test case prioritization method based on fault severity," 2015 6th IEEE International Conference on Software Engineering and Service Science (ICSESS), Beijing, 2015, pp. 737-741.
[13] C. Lin, C. Chen, C. Tsai and G. M. Kapfhammer, "History-Based Test Case Prioritization with Software Version Awareness," 2013 18th International Conference on Engineering of Complex Computer Systems, Singapore, 2013, pp. 171-172.
[14] C. Malz and P. Göhner, "Agent-Based Test Case Prioritization," 2011 IEEE Fourth International Conference on Software Testing, Verification and Validation Workshops, Berlin, 2011, pp. 149-152.
[15] Y. I. Salem and R. Hassan, "Requirement-based test case generation and prioritization," 2010 International Computer Engineering Conference (ICENCO), Giza, 2010, pp. 152-157.
[16] T. Noguchi, H. Washizaki, Y. Fukazawa, A. Sato and K. Ota, "History-Based Test Case Prioritization for Black Box Testing Using Ant Colony Optimization," 2015 IEEE 8th International Conference on Software Testing, Verification and Validation (ICST), Graz, 2015, pp. 1-2.
[17] R. Abid and A. Nadeem, "A novel approach to multiple criteria based test case prioritization," 2017 13th International Conference on Emerging Technologies (ICET), Islamabad, 2017, pp. 1-6.
[18] D. Di Nardo, N. Alshahwan, L. Briand and Y. Labiche, "Coverage-Based Test Case Prioritisation: An Industrial Case Study," 2013 IEEE Sixth International Conference on Software Testing, Verification and Validation, Luxembourg, 2013, pp. 302-311.
[19] Y. Cho, J. Kim and E. Lee, "History-Based Test Case Prioritization for Failure Information," 2016 23rd Asia-Pacific Software Engineering Conference (APSEC), Hamilton, 2016, pp. 385-388.
[20] M. J. Arafeen and H. Do, "Test Case Prioritization Using Requirements-Based Clustering," 2013 IEEE Sixth International Conference on Software Testing, Verification and Validation, Luxembourg, 2013, pp. 312-321.
[21] N. Dini, A. Sullivan, M. Gligoric and G. Rothermel, "The Effect of Test Suite Type on Regression Test Selection," 2016 IEEE 27th International Symposium on Software Reliability Engineering (ISSRE), Ottawa, ON, 2016, pp. 47-58
[22] Priyanka, H. Kumar and N. Chauhan, "A novel approach for selecting an effective regression testing technique," 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, 2016, pp. 1122-1125
[23] P. Dhareula and A. Ganpati, "Identification of attributes for test case reusability in regression test selection techniques," 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, 2016, pp. 1144-1147.

[24] S. Akimoto, S. Nakanishi, R. Yaegashi and T. Takagi, "Extended differential control flow graphs for the selection of test cases in regression testing," 2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS), Okayama, 2016, pp. 1-2.

[25] S. Dahiya, R. K. Bhatia and D. Rattan, "Regression test selection using class, sequence and activity diagrams," in IET Software, vol. 10, no. 3, pp. 72-80, 6 2016

[26] M. M. Fawzy, M. S. El-Mahallawy and H. El-Deeb, "Enhanced code coverage approach for regression testing," 2015 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT), Kumaracoil, 2015, pp. 438-442

[27] P. Kandil, S. Moussa and N. Badr, "A methodology for regression testing reduction and prioritization of agile releases," 2015 5th International Conference on Information & Communication Technology and Accessibility (ICTA), Marrakech, 2015, pp. 1-6.

[28] S. K. Mohapatra and M. Pradhan, "Finding representative test suit for test case reduction in regression testing," 2015 International Conference on Computer, Communication and Control (IC4), Indore, 2015, pp. 1-6.

[29] D. Gao, X. Guo and L. Zhao, "Test case prioritization for regression testing based on ant colony optimization," 2015 6th IEEE International Conference on Software Engineering and Service Science (ICSESS), Beijing, 2015, pp. 275-279

[30] E. D. Ekelund and E. Engström, "Efficient regression testing based on test history: An industrial evaluation," 2015 IEEE International Conference on Software Maintenance and Evolution (ICSME), Bremen, 2015, pp. 449-457

[31] N. Chauhan, M. Dutta and M. Singh, "A Program Model Based Regression Test Selection Technique for Object-Oriented Programs," 2015 Fifth International Conference on Communication Systems and Network Technologies, Gwalior, 2015, pp. 918-924.

[32] P. Konsaard and L. Ramingwong, "Total coverage based regression test case prioritization using genetic algorithm," 2015 12th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), Hua Hin, 2015, pp. 1-6

[33] R. K. Saha, L. Zhang, S. Khurshid and D. E. Perry, "An Information Retrieval Approach for Regression Test Prioritization Based on Program Changes," 2015 IEEE/ACM 37th IEEE International Conference on Software Engineering, Florence, 2015, pp. 268-279.

[34] S. Akimoto, R. Yaegashi and T. Takagi, "Test case selection technique for regression testing using differential control flow graphs," 2015 IEEE/ACIS 16th International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), Takamatsu, 2015, pp. 1-3

[35] M. R. Nejad Dobuneh, D. N. A. Jawawi, M. Ghazali and M. V. Malakooti, "Development test case prioritization technique in regression testing based on hybrid criteria," 2014 8th. Malaysian Software Engineering Conference (MySEC), Langkawi, 2014, pp. 301-305.