THINK INDIA JOURNAL
ISSN: 0971-1260
Vol-22-NO-30 Special Issue-December-2019

An Insightof Test Case Prioritization Techniques and its Associated Outcomes

Ashima[1], Mohit Arora[2*], Shivali Chopra[3]

School of Computer Science and Engineering

Lovely Professional University. Phagwara, Punjab, India

[1]ashigarg57452@gmail.com,[2]mohit.15980@lpu.co.in,[3]shivali.19259@lpu.co.in

* Corresponding Author

Abstract

The testing approach that helps in identifying faults in software that is launched with its new version is known as regression testing. The test cases are cases which are applied to test the software. The test cases generally cover two things which are coverage and time. The coverage represents the area of the code which is covered by the software and time is the factor which defines the time taken for the testing. The test case prioritization and test case generation are the technique of regression testing. The various techniques are designed in the previous years for the test case prioritization. In the test case prioritization, test cases of the prioritized according to the software changes. In this paper, various techniques of test case prioritization are reviewed and analyzed in terms of certain parameters. The test case prioritization will reduce the testing time of the software and also able to detect maximum number of faults from the software. The test case prioritization techniques are generally categorized into automated and manually prioritization. The automated prioritization techniques are efficient technique for the test case prioritization.

Keywords: Regression testing, Test case prioritization, Test case generation**,** Fault detection

1. Introduction

Any kinds of test suites designed by the software developers are saved by them to be reutilized in future as per the evolution of software. In the software industry, such test suite reutilization is pervasive in the form of regression testing [1]. One-half of the cost of software maintenance is accounted together with other regression testing activities. An inordinate amount of time can be consumed by running all test cases in an existing test suite. For attempting to maximize some objective function, the prioritization techniques are responsible for scheduling the test cases in order to perform regression testing. For instance, at the fastest rate possible, the code coverage

ISSN: 0971-1260
Vol-22-NO-30 Special Issue-December-2019

could be achieved by the testers that wish to schedule test cases [2]. As per the order to expected frequency of use, the features can also be exercised. In a manner that reflects the historical propensity to fail, the subsystems are exercised. In case when there is less time needed for executing test cases, the test case prioritization is not cost-effective. For scheduling the test cases in any order, this method might be the most expedient [3]. The test case prioritization can be beneficial in case when there is very longer time required for running all the test cases included within the test suite.

Based on certain criterion, the test cases are within an execution mechanism. Increasing the likelihood that the test cases will meet certain objective more closely in comparison to when they were executed in any other order, is the major objective of prioritization [4]. Huge variety of objectives can be addressed in test case prioritization among which some are:

i. The rate of fault detection might be increased by testers. The likelihood of identifying the faults present in a run of regression tests is called fault detection.

ii. For maximizing the rate of detecting high-risk faults, it is necessary to locate the faults existing in testing processes by the testers [5].

iii. The possibility of showing those errors which have relevancy to a specific code is increased by testers through regression testing mechanism.

iv. For the system under test, the testers might wish to increase the coverage of coverable code of the system.

v. The confidence in reliability of system under test can be increased by the tests at higher speed.

It is possible to present an intractable test case prioritization related problem based on the choice of objectives.

There are few important factors based on which the test cases are prioritized [6]. They are:

a. Customer Requirements: Based on the requirements of customers that are documented in the requirement specification collection phase, the test cases are arranged in these customer

ISSN: 0971-1260
Vol-22-NO-30 Special Issue-December-2019

requirement-based prioritization techniques. For the important factors, the values are accredited and a need for prioritization of test cases is shown by the high value factor.

b. Coverage Based: A source code of a program is validated and calculated based on the coverage during the testing is known as the coverage-based prioritization technique. The form of code that has been coated is referred to as coverage [7]. The coverage here can be related to any number of requirements involved. Therefore, the most important sections of main code can be tested and prioritized by the test case in this approach.

c. Cost effective: Prioritization of test cases is done depending upon the cost factor. There are certain important factors related to test cases which perform execution, validation and analysis of cost related factors for calculating this parameter. Thus, there is maximal value for the test cases demand the diminish cost.

d. History Based: The priorities are assigned to test cases based on their prior achievements in this category. It is possible to decline or increase the previous performances of test cases in order to make sure that testing is valuable for the ongoing sessions.

2. Techniques for Test Case Prioritization

The various test case prioritization techniques are shown in the comparison table 1 below. It is highly costly to perform regression testing in software development life cycle. However, all the requirements of stakeholders need to be fulfilled by the project. Around 50% of the total software cost is consumed within the testing phase. Regression testing is used by the engineers to assign the test cases preferences [8]. The test cases that have higher significance are executed here. Fault detection is the major target of test case prioritization. For prioritizing the test cases, there are several techniques invented by various researchers in software testing today. The widely used techniques by researchers these days are compared below. Every technique has its own merits and demerits.

Table 1: Test Case Prioritization techniques

Sr. No.	Technique	Key Idea	Advantage	Disadvantage
1.	Fault	The requirement	1. The software quality is	1. The induced factor of

ISSN: 0971-1260
Vol-22-NO-30 Special Issue-December-2019

	Severity	specification is the major idea based on which this technique is designed.	improved. 2. High severity faults are discovered easily here. 3. The fault detection rate is improved. 4. The most important factor is to provide requirements volatility.	requirements volatility is not removed. 2. There is limited project scope.
2.	Fault Localization	It is based on the fault tolerance's execution information.	1. The analysis approach is highly detailed. 2. The failure is exposed at higher rate.	1. The effectiveness is less. 2. It might result in affecting the subsequent fault localization.
3.	Mutation faults	The changes in the program code are the base of this technique.	1. There is improvement in fault detection rate.	1. There is no significance about the cost reduction.
4.	Ordered Sequence of Program Elements	The frequencies of program element are executed as base of this technique.	1. In the form of loops, the bugs are detected. 2. This approach is highly cost effective.	1. The efficiency of this approach is not much.
5.	APFD	The average faults identified per minute are the base of this technique.	1. At system level, the rate of faults detection is easy.	1. For fault detection, the efficiency of this technique is less.
6.	Model Checker	The functional model of program test is the key idea of this technique.	When generating the test cases, the prioritization is efficiently applied.	The factors like costs of potential faults and actual test case execution costs are not included here.
7.	Search Algorithm	The code coverage is the key idea used in this technique.	1. It is highly efficient and flexible. 2. The prioritization of test cases is no affected by the size of program.	1. Large number of test cases cannot be solved. 2. Various results are generated in some cases.

2. Related Literature

Md. Hasan Mahmood, et.al (2017) recommended a novel technique to assign priority to test cases on the basis of their ability to cover maximal working of a software system [9]. This technique was based on real time factors in software development life cycle. Identifying test cases covering imperative business performances was the main aim of prioritization. The tested outcomes demonstrated that the recommended technique successfully decreased cost and time of the software system. In addition, this technique made certain the entire performance exposure.

Faiza Farooq, et.al (2017) recommended a novel scheme to prioritize the test cases on the basis of transformation testing [10]. Different faults in the original program were introduced by

ISSN: 0971-1260
Vol-22-NO-30 Special Issue-December-2019

checking changes. Several changed copies of the program and test case that exposed maximal number of these faults was assigned with maximum priority. The recommended approach was used for the testing of cases and measured the fault discovery rates generated by the prioritized test cases. The achieved results depicted that the recommended method made improvements in the fault discovery rate of test cases.

Rubing Huang, et.al (2017) examined feasible ATCP approaches on the basis of four techniques [11]. These types included NIGP and different other variants related to this approach only. It was analyzed that the ICBP type showed more testing efficiency as compared to other types. Shockingly, it was analyzed that that NIGP could be efficient as IMBP. Also, SBP type occasionally achieved better fault detection rates in contrast to ICBP approaches.

Andrea Janes, (2017) tried to use process-mining as an approach for constructing a model. This model described the ways through which users establish communication with a system for generating test cases and assigning priority to them [12]. The process-mining could use the behavior of client to extract information about the mostly utilized system parts and their order of execution. It was possible to create test cases replicating user input and assign priority to test suites using process mining. A range of opportunities had been provided by this approach for generating test cases and assigning priority to them. However, it was required to verify the created process models for minimizing figurative partiality.

Riza Dhiman, et.al (2019) recommended a new technique for test case prioritization [13]. The recommended technique used manual slicing and automatic slicing for detecting maximal faults from the scheme. Some changes were carried out in this scheme to release the new adaptation. The recommended approach was implemented in the MATLAB tool. The simulation results depicted that the recommended approach provided the fault detection rate-based enhancements and reduction in the execution time in regression testing.

Dharmveer Kumar Yadav, et.al (2016) assigned priority to test cases for object-based program [14]. The most valuable work was to choose competent and appropriate test suites in regression testing from the test case. A prioritization method was implemented with the objective that regression testing related cost could be reduced. The priority was assigned on the basis of

different parameters using fuzzy logic. Both prioritized and non-prioritized test cases were analyzed with the help of APFD. The efficiency related outcomes had been shown through graphical representation.

Dipesh Pradhan, et.al (2018) recommended a new technique called REMAP for black-box dynamic test case prioritization [15]. This technique used rule mining and multi-objective exploration. The three main elements of this approach were identified. The performance of recommended approach was evaluated in terms of different parameters. The tested results depicted that this approach performed better than other existing techniques in most of the cases. This approach achieved better APFD (Average Percentage of Faults Detected) of 18%.

Table 2: Effectiveness of existing techniques

Authors	Techniques	Effectiveness
Md. Hasan Mahmoodet.al [9]	The tested outcomes demonstrated that the recommended technique successfully decreased cost and time of the software system	The accuracy is achieved up to 90 percent for defect prediction
Faiza Farooq et.al [10]	Maximum priority was assigned to the test cases and other changed copies of programs that were exposing maximal numbers of faults.	The maximum accuracy is achieved up to 80 percent for the defect prediction
Rubing Huang et.al [11]	These types included non-information-guided prioritization (NIGP), ICBP,IMBP and similarity-based prioritization (SBP)	The execution time is achieved upto 10 second and defect is achieved up to 80 percent
Andrea Janes et.al [12]	This model described the ways through which users establish communication with a system for generating test cases and assigning priority to them	A range of opportunities had been provided by this approach for generating test cases and assigning priority to them.
Riza Dhiman et.al [13]	The recommended technique used manual slicing and automatic slicing for detecting maximal faults from the scheme	The technique has detected the faults up to 80 to 90 percent for the fault detection
Dharmveer Kumar et.al [14]	A prioritization method was implemented with the aim of reducing the regression testing related cost. The priority was assigned on the basis of different parameters using fuzzy logic	The APFD matrix is constructed which give result upto 90 percent in terms of effectiveness
Dipesh Pradhan et.al [15]	The performance of recommended approach was evaluated in terms of different parameters. The tested results depicted that this approach performed better than other existing techniques.	This approach achievedbetter APFD value of about 18%.

The table of comparison is given in Table 3.

Table 3: Literature Review outcomes

Authors Names	Year	Description	Outcomes
Md. Hasan Mahmood, et.al [9]	2017	Recommended a novel technique to assign priority to test cases on the basis of their ability to cover maximal working of a software system.	The tested outcomes demonstrated that the recommended technique successfully decreased cost and time of the software system. In addition, this technique made certain the entire performance exposure.
Faiza Farooq, et.al [10]	2017	Recommended a novel scheme to prioritize the test cases on the basis of transformation testing.	The achieved results depicted that the recommended method made improvements in the fault discovery rate of test cases.
Rubing Huang et.al [11]	2017	Examined feasible ATCP approaches on the basis of four techniques. These types included non-information-guided prioritization (NIGP); interaction coverage-based prioritization (ICBP); input-model mutation-based prioritization (IMBP); and similarity-based prioritization (SBP).	It was analyzed that the ICBP type showed more testing efficiency as compared to other types.
Andrea Janes et.al [12]	2017	Tried to use process-mining as an approach for constructing a model. This model described the ways through which users establish communication with a system for generating test cases and assigning priority to them.	A range of opportunities had been provided by this approach for generating test cases and assigning priority to them.
Riza Dhiman, et.al [13]	2019	Recommended a new technique for test case prioritization. The recommended technique used manual slicing and automatic slicing for detecting maximal faults from the scheme.	The simulation results depicted that the recommended approach enhanced fault detection rate and reduced the execution time in regression testing.
Dharmveer Kumar et.al [14]	2016	Assigned priority to test cases for object-basedprogram. The most valuable work was to choose competent and appropriate test suites in regression testing from the test case.	APFD was used to evaluate the test cases that were prioritized as well as non-prioritized. Graphical representation was used to show the efficiency of prioritized test cases against non-prioritized test case.
Dipesh Pradhan et.al [15]	2018	Recommended a new technique called REMAPfor black-box dynamic test case prioritization. This technique used rule mining and multi-objective exploration	The tested results depicted that this approach performed better than previous techniques. This approach achievedbetter APFD of 18%.

3. Conclusion and Future scope

In this paper, various techniques of test case prioritization are reviewed and analyzed in terms of certain parameters. The technique of test case prioritization is broadly classified into manual and

automated slicing. The manual slicing technique is the technique which can manually prioritize the test case.

As a future work, optimization techniques for automated slicing in test case prioritization can be employed.

REFERENCES

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

[2] Hema Srikanth, Laurie Williams and Jason Osborne, "System Test Case Prioritization of New and Regression Test Cases", In Proceedings of the 4th International Symposium on Empirical Software Engineering (ISESE), pages 62–71. IEEE Computer Society, 2005.

[3] Hyunsook Do and Gregg Rothermel, "On the Use of Mutation Faults in Empirical Assessments of Test Case Prioritization Techniques", IEEE Transactions on Software Engineering, V. 32, No. 9, pages 733- 752, 2006.

[4] Hyunsook Do and Gregg Rothermel, "A Controlled Experiment Assessing Test Case Prioritization Techniques via Mutation Faults", Proceedings of the IEEE International Conference on Software Maintenance, pages 411-420, 2005.

[5] Sreedevi Sampath, Sara Sprenkle, Emily Gibson and Lori Pollock, "Web Application Testing with Customized Test Requirements – An Experimental Comparison Study", 17th International Symposium on Software Reliability Engineering (ISSRE'06), 2006.

[6] Xiaofang Zhang, Baowen Xu, Changhai Nie and Liang Shi, "An Approach for Optimizing Test Suite Based on Testing Requirement Reduction", Journal of Software (in Chinese), 18(4): 821-831, 2007.

[7] Xiaofang Zhang, Baowen Xu, Changhai Nie and Liang Shi, "Test Suite Optimization Based on Testing Requirements Reduction", International Journal of Electronics & Computer Science, 7(1): 9- 15, 2005.

ISSN: 0971-1260
Vol-22-NO-30 Special Issue-December-2019

[8] Xiaofang Zhang, Baowen Xu, Zhenyu Chen, Changhai Nie and Leifang Li, "An Empirical Evaluation of Test Suite Reduction for Boolean Specification-based Testing", The Eighth International Conference on Quality Software, 2008.

[9] Md. Hasan Mahmood, Md. Shazzad Hosain, "Improving test case prioritization based on practical priority factors", 2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS)

[10] Faiza Farooq, Aamer Nadeem, "A Fault Based Approach to Test Case Prioritization", 2017 International Conference on Frontiers of Information Technology (FIT)

[11] Rubing Huang, Weiwen Zong, Dave Towey, Yunan Zhou, Jinfu Chen, "An Empirical Examination of Abstract Test Case Prioritization Techniques", 2017 IEEE/ACM 39th International Conference on Software Engineering Companion (ICSE-C)

[12] Andrea Janes, "Test Case Generation and Prioritization: A Process-Mining Approach", 2017 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW)

[13] Riza Dhiman, Vinay Chopra, "Novel Approach for Test Case Prioritization Using ACO Algorithm", 2019 IEEE 2nd International Conference on Information and Computer Technologies (ICICT)

[14] Dharmveer Kumar Yadav, Sandip Dutta, "Test case prioritization technique based on early fault detection using fuzzy logic", 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)

[15] Dipesh Pradhan, Shuai Wang, Shaukat Ali, Tao Yue, Marius Liaaen, "REMAP: Using Rule Mining and Multi-objective Search for Dynamic Test Case Prioritization", 2018 IEEE 11th International Conference on Software Testing, Verification and Validation (ICST)