

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320-088X
IMPACT FACTOR: 6.017



IJCSMC, Vol. 6, Issue. 4, April 2017, pg.216 – 221

Review of Prioritization Techniques in Regression Testing

Neeraj Kumar Saklani

Department of Computer Science & Engineering
Baddi University
Baddi, H.P., India

neeraj.mtcse.15pcs101@baddiuniv.ac.in

Parulpreet Singh

Department of Computer Science & Engineering
Baddi University
Baddi, H.P., India

parul.preet@baddiuniv.ac.in

Abstract- Due to the rapidly changing business and operating environment, software evolution is bound to happen. As software evolution goes on, the system requires to be retested again to test its validity and thus regression testing helps in achieving this goal. Regression testing can be categorized into different research areas and one of its important areas is test case prioritizations which arrange test cases on defined benchmarks with respect to the available resources and make sure that the most important or critical test cases are tested first. In this paper, we study the different test case prioritization techniques and rise out the different problems. This study will give an understanding of the problem currently occurring in test case prioritization and thus guide for future research work on test case prioritization.

Keywords: Regression testing, test cases, prioritization techniques.

I. INTRODUCTION

A change in software system that might be environmental change or simply generating higher version, they must be retested to ensure quality. Regression testing mainly focuses on change in software and performs testing on the improved version of the software [1]. The main objective of regression testing is to ensure that these changes or modification to the software do not affect the other modules of the software [2]. As regression testing holds 80% of the testing budget [3].

Regression testing is one of the most expensive and repetitive processes in software testing. To maintain quality, software system must be retested as new features or changes are made constantly. The major issue with regression testing is its cost [4]. Sometimes regression testing consumes 50% of testing cost [8]. Another issue with

regression testing to detect maximum fault and validate software under test [10]. The main focus of regression testing is to increase fault detection and decrease their cost of execution [2]. These goals are achieved by test case selection methods [6], test suite reduction [7], test case prioritization [8]. From the test suit first test cases are selected and then either they are ordered based on their priority or reduced by eliminating the unnecessary test cases based on some survey or expert advice [9]. In regression testing, researcher mainly focuses on the test case selection and test case prioritization to reduce software execution time and improve the fault detection rate [2,10,11].

II. REGRESSION TESTING

Regression testing can be divided into several sub categories. There are four main types of regression testing:

- a) Test case selection
- b) Test case prioritization
- c) Test suite reduction/minimization

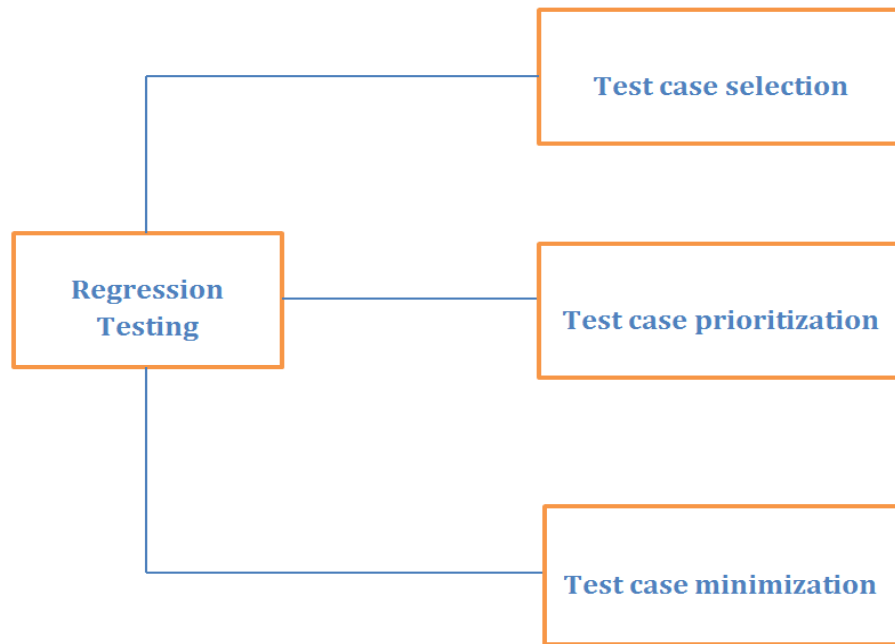


Figure 2.1: Regression Testing

- i) **Test case selection:** Test case selection process is able to select the subset of test cases from the test suite that has the potential to detect error initiated from a change. Thus, this process is used to identify and select these test cases that are relevant based on particular criteria [12]. To understand test case selection suppose we have P as the first program, take P' as a new version of program P and take I as an iteration of the program P. The main goal of regression test selection is to build I' which is a subset of I to be used for P'. This technique will select all the test cases that can produce errors in the new modified version of program P' [13].
- ii) **Test case prioritization:** Test case prioritization process considers ordering of test cases for detection of faults at the earliest. Test case prioritization helps in enabling the execution of the most critical test case at earlier

stage [14]. It is important to execute the test cases based on their priority to optimize the efficiency and execution time.

- iii) **Test suit minimization:** Test suit minimization/reduction is carried out to eliminate the redundant test cases. As software development goes on, there is an addition of new test cases to validate the new features needs to be tested to discover new errors. This creates a form of redundancy with test cases covering a single test requirement.

A. Analysis of Regression Testing Techniques

It is reported that test suite minimization and test case selection techniques encountered with some problems. These techniques help in reducing the number of test cases in a test suite but these problem decrease the popularity of these techniques. These problems are:

- 1) Test case selection techniques is not suitable for code enhancements and functionality upgrades as it emphasis on modified part of software code.
- 2) Test cases are removed permanently from the test suites.

On the other hand, test case prioritization mainly focuses on complete test suite and discovers the efficient ordering of test cases by covering the fault detection rate. Prioritization methods follow the code coverage which helps in providing the early feedback to software testing team. A test case prioritization technique uses the following method to prioritize the test cases from the test suite:

- a) **Code coverage data:** Test cases are prioritized with the information collected by coverage data like code coverage, function coverage and conditional coverage.
- b) **Code modification information:** Information collected from code enhancement can be used for prioritization of test cases.

B. Test case prioritization categories

Test case prioritization (TCP) methods are categorized into many sub categories. A survey explores 47 TCP studies published between 1999 & 2009. TCP methods are further grouped as:

- i. **Coverage based TCP:** These TCP methods use maximum coverage of information to identify the more faults during testing. TCP methods use coverage information for prioritizing test cases.
- ii. **Probability based TCP:** In these methods probability of each test case is calculated based on the results of previous test cases.
- iii. **Human based TCP:** These TCP methods prioritize the test cases based on human evaluation.
- iv. **History based TCP:** These TCP methods use history of previous test cases to prioritize.
- v. **Requirement based TCP:** Test cases are prioritize based on the critical requirement properties that need to be executed first like customer priority and change impact analysis.

C. Analysis of Test case Prioritization Techniques

A test case prioritization technique is used to reorder the test cases so that the most critical or highest priority test cases can be executed first. The priority of these test cases is set on the basis of information collected from different TCP methods discussed earlier. Code coverage and fault detection provide linear relationship between them [15]. It is reported that these two methods are commonly used by test case prioritization techniques. The overlapping of these methods does not mean that they are fully dependent on each other.

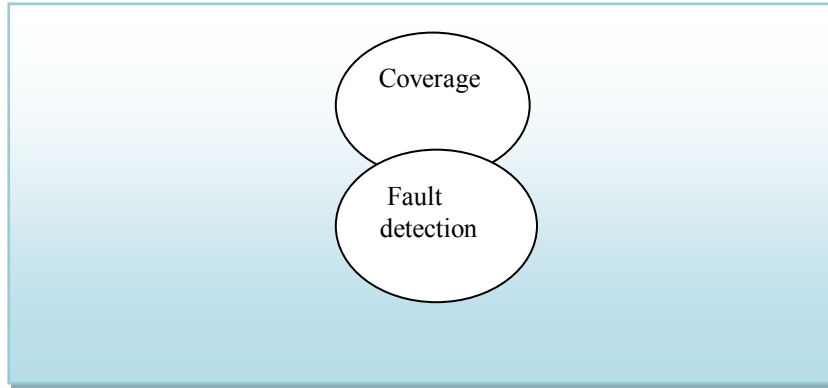


Figure 2.2: Basic attributes of TCP methods

III. GROWTH OF TCP METHODS

S.No	Topic	Problem Raised	Proposed Solution	Year
1	An empirical study of the effect of time constraints on the cost benefits of regression testing [16].	Does time constraints and limit play a role in the effectiveness of prioritization and if so, is the impact significant.	Time constraints have a strong behavioral impact on the cost benefits of test case prioritization and regression testing.	2008
2	Automated test case prioritization with reactive GRASP [17].	Does metaheuristic search based algorithm or search based prioritization techniques Performance depends on size of the program.	As size of the program & test suit grows the performance of many search based techniques drastically change. A technique was introduced that could stay as effective however it was observed to consume more execution time than other compared techniques.	2010
3	Developing a single model & test prioritization strategies for event-driven software (EDS) [18].	Conventional prioritization techniques don't work well with GUI and EDS mainly because they produce a very large amount of states and events. Also there is no abstract model to portray the event driven nature of these applications.	Developed a single abstract model for GUI & web testing alongside a shared prioritization function based on the developed model and a shared prioritization criterion.	2011
4	A history-based cost cognizant test case prioritization technique in regression testing [19].	Most prioritization technique assume that all test case cost & faults severities are equal. However, they usually vary & testers may not obtain source code before prioritization.	Proposed a cost cognizant prioritization technique that prioritizes test cases based on their historical data. It also takes into account cost of test & fault severities based without the need for source code analysis.	2012

5	Bridging the gap between the total & additional test case prioritization strategies [20].	The total & addition prioritization strategies each have weakness of their own & thus are not considered complete when used alone as each can have a downside.	By combing both strategies, each will mask the weakness of the other & the results indicate that it out performs both strategies in effectiveness	2013
6	Towards statistical prioritization for software product lines testing [21].	Software product lines (SPL) are inherently difficult to test due to the sheer amount of product that need to be tested, thus prioritization is a describe option. However, current techniques do not take into account the product behaviour.	Proposed an approach that are to be tested according to criteria based on the actual product behaviour and thus increasing the relevance of the testing & reducing the risk of not detecting error in many products.	2014
7	Coverage based regression test case selection, minimization & prioritization: a case study on industrial system [22].	Across all coverage based prioritization techniques does coarser grained coverage perform better or the finer grained coverage.	For different coverage based techniques, it was discovered that techniques based on additional coverage (block basic block decision) outperformed all other techniques used in the study.	2015

IV. CONCLUSION

In this study different testing prioritization techniques of regression testing are reviewed. The main purpose of this review is to understand the different techniques and their support in test case prioritization. It is observed that primary focus in prioritization techniques is changed from code analysis to history based because of better fault detection rate.

Most of the researches performed related to test case prioritization, researchers show their keen interest towards artificial intelligence as compare to code coverage. In future, research is needed to compare both artificial based techniques and code coverage based techniques on large scale software development to know which technique is more efficient.

REFERENCES

- [1] K. K. Ranga, "Analysis and Design of Test Case Prioritization Technique for Regression Testing", International Journal for Innovative Research in Science and Technology, vol. 2, (2015), pp. 248-252.
- [2] S. Yoo and M. Harman, "Regression testing minimization, selection and prioritization: a survey", Software Testing, Verification and Reliability, vol. 22, (2012), pp. 67-120.
- [3] T. Hall, S. Beecham, D. Bowes, D. Gray and S. Counsell, "A systematic literature review on fault prediction performance in software engineering", Software Engineering, IEEE Transactions on, vol. 38, (2012), pp. 1276-1304.
- [4] A. Srivastava and J. Thiagarajan, "Effectively prioritizing tests in development environment", in ACM SIGSOFT Software Engineering Notes, (2002), pp. 97-106.
- [5] D. Marijan, A. Gotlieb and S. Sen, "Test case prioritization for continuous regression testing: An industrial case study", in Software Maintenance (ICSM), 2013 29th IEEE International Conference on, (2013), pp. 540-543.
- [6] A. Pravin and S. Srinivasan, "Effective Test Case Selection and Prioritization in Regression Testing", Journal of Computer Science, vol. 9, (2013), p. 654.
- [7] M. Harman, P. McMinn, M. Shahbaz and S. Yoo, "A comprehensive survey of trends in oracles for software testing", University of Sheffield, Department of Computer Science, Tech. Rep. CS-13-01, (2013).
- [8] K. Zhai, B. Jiang and W. Chan, "Prioritizing test cases for regression testing of location-based services: metrics, techniques and case study", Services Computing, IEEE Transactions on, vol. 7, (2014), pp. 54-67.

- [9] J. Petke, S. Yoo, M. B. Cohen and M. Harman, "Efficiency and early fault detection with lower and higher strength combinatorial interaction testing", in Proceedings of the 2013 9th Joint Meeting on Foundations of Software Engineering, **(2013)**, pp. 26-36.
- [10] A. Orso and G. Rothermel, "Software testing: a research travelogue (2000–2014)," in Proceedings of the on Future of Software Engineering, **(2014)**, pp. 117-132.
- [11] M. J. Harrold, "Testing: a roadmap", in Proceedings of the Conference on the Future of Software Engineering, **(2000)**, pp. 61-72.
- [12] M. Grindal, B. Lindström, J. Offutt and S. F. Andler, "An evaluation of combination strategies for test case selection", Empirical Software Engineering, vol. 11, **(2006)**, pp. 583-611.
- [13] G. Rothermel, R. H. Untch, C. Chu and M. J. Harrold, "Prioritizing test cases for regression testing", Software Engineering, IEEE Transactions on, vol. 27, **(2001)**, pp. 929-948.
- [14] S. Elbaum, A. G. Malishevsky and G. Rothermel, "Test case prioritization: A family of empirical studies", Software Engineering, IEEE Transactions on, vol. 28, **(2002)**, pp. 159-182.
- [15] M. Gligoric, A. Groce, C. Zhang, R. Sharma, M. A. ALIPOUR and D. Marinov, "Guidelines for coverage-based comparisons of non-adequate test suites", Space, vol. 6, **(2014)**, pp. 1,142.
- [16] H. Do, S. Mirarab, L. Tahvildari and G. Rothermel, "An empirical study of the effect of time constraints on the cost-benefits of regression testing", in Proceedings of the 16th ACM SIGSOFT International Symposium on Foundations of software engineering, **(2008)**, pp. 71-82.
- [17] C. L. B. Maia, R. A. F. do Carmo, F. G. de Freitas, G. A. L. de Campos and J. T. de Souza, "Automated test case prioritization with reactive GRASP", Advances in Software Engineering, **(2010)**, vol. 2010.
- [18] R. C. Bryce, S. Sampath and A. M. Memon, "Developing a single model and test prioritization strategies for event-driven software", Software Engineering, IEEE Transactions on, vol. 37, **(2011)**, pp. 48-64.
- [19] Y.-C. Huang, K.-L. Peng and C.-Y. Huang, "A history-based cost-cognizant test case prioritization technique in regression testing", Journal of Systems and Software, vol. 85, **(2012)**, pp. 626-637.
- [20] L. Zhang, D. Hao, L. Zhang, G. Rothermel and H. Mei, "Bridging the gap between the total and additional test-case prioritization strategies", in Software Engineering (ICSE), 2013 35th International Conference on, **(2013)**, pp. 192-201.
- [21] X. Devroey, G. Perrouin, M. Cordy, P.-Y. Schobbens, A. Legay and P. Heymans, "Towards statistical prioritization for software product lines testing", in Proceedings of the Eighth International Workshop on Variability Modelling of Software-Intensive Systems, **(2014)**, p. 10.
- [22] D. D. Nardo, N. Alshahwan, L. Briand and Y. Labiche, "Coverage-based regression test case selection, minimization and prioritization: a case study on an industrial system", Software Testing, Verification and Reliability, vol. 25, **(2015)**, pp. 371-396.