'FUTURIZM-2014, (14TH& 15TH March 2014), 'P.D.E.As College of Engineering, Manjari[Bk]', 'IIP Cell,'Manjari Road,Hadapsar,Pune-412307. India.', 'Chief Editor:Dr.K.R.Harne', Editors: Prof R V Patil, Prof Niraja Jain

**International Journal of Innovative Technology & Adaptive Management (IJITAM)**
**ISSN: 2347-3622, Volume-1, Issue-6 , March, 2014**

# AN EFFICIENT TEST CASE PRIORITIZATION IN REGRESSION TESTING TEST CASES THROUGH GENETIC ALGORITHM.

Surendra Mahajan[1], Dr. S. D. Joshi [2],Dr. V. Khanaa[3]
1 Research Scholar, Department Of Computer Science & Engineering, Bharat University, Chennai-73, sa_mahajan@yahoo.com
2Professor, Department Of Computer Engineering, BharathiVidyapeeth University College Of Engineering, Pune-43,sdj@live.in
3Dean-Center for Information, Bharat University, Chennai-73,
khanaakrishna@gmail.com

**1. ABSTRACT**

Software maintenance is defined as activities performed on a software product following to its release for use. These activities control the changes that are often crucial during this phase of the software life cycle. It is important to retest in order to authenticate that these modifications or changes do not have unintended effects and, therefore, the system still satisfies with its specified requirements. Due to time and resource constraints it is not possible to execute all the test cases. The selective retesting of the system or component is called regression testing. Regression testing establishes the confidence in the modified program. It may account for as much as half of the cost of software maintenance. The importance of regression testing can be understood from the fact that the single most costly bug in software history could have been revealed by regression testing. Regression testing is a kind of testing which requires maximum effort, time and cost. In fact, it might be hard to run the whole application unattended and to simulate any asynchronous input (e.g., interactive inputs) the application may receive. In such cases, regression testing can last days or weeks and can involve substantial human effort. Hence a technique like Test case prioritization has to be devised which will lead to early fault detection maintenance. The importance of regression testing can be understood from the fact that the single most costly bug in software history could have been revealed by regression testing. Regression testing is a kind of testing which requires maximum effort, time and cost. In fact, it might be hard to run the whole application unattended and to simulate any asynchronous input (e.g., interactive inputs) the application may receive. In such cases, regression testing can last days or weeks and can involve substantial human effort. Hence a technique like Test case prioritization has to be devised which will lead to early fault detection.

Software developers often save the test

**International Journal of Innovative Technology & Adaptive Management (IJITAM)**
**ISSN: 2347-3622, Volume-1, Issue-6 , March, 2014**

suites they develop for their software so that they can reuse those test suites later as the software evolves. Such test suite reuse, in the form of regression testing, is pervasive in the software industry and, together with other regression testing activities, has been estimated to account for as much as one-half of the cost of software maintenance.

Running all of the test cases in a test suite, however, can require a large amount of effort. For example, if one of the product is having 20,000 lines of code, the entire test suite requires roughly seven weeks to run. For this reason, researchers have considered various techniques for reducing the cost of regression testing, including regression test selection and test suite minimization techniques. Regression test selection techniques reduce the cost of regression testing by selecting an appropriate subset of the existing test suite, based on information about the program, modified version, and test suite. Test suite minimization techniques lower costs by reducing a test suite to a minimal subset that maintains equivalent coverage of the original test suite with respect to a particular test adequacy criterion. Test case prioritization techniques schedule test cases for regression testing in an order that attempts to maximize some objective function. For example, testers might wish to schedule test cases in an order that achieves code coverage at the fastest rate possible, exercises features in order of expected frequency of use, or exercises subsystems in an order that reflects their historical propensity to fail. When the time required to execute all test cases in a test suite is short, test case prioritization may not be cost effective and it may be most expedient simply to schedule test cases in any order. When the time required to run all test cases in the test suite is sufficiently long, however, test case prioritization may be beneficial.

When the time required to re-execute an entire test suite is short, test case prioritization may not be cost-effective - it may be sufficient simply to schedule test cases in any order. When the time required to execute an entire test suite is sufficiently long, however, test case prioritization may be beneficial, because in this case, meeting testing goals earlier can yield meaningful benefits.

Because test case prioritization techniques do not themselves discard test cases, they can avoid the drawbacks that can occur when regression test selection and test suite minimization discard test cases. Alternatively, in cases where the discarding of test cases is acceptable, test case prioritization can be used in conjunction with regression test selection or test suite minimization techniques to prioritize the test cases in the selected or minimized test suite. Further, test case prioritization can increase the likelihood that, if regression testing activities are unexpectedly terminated, testing time will have been spent more beneficially than if test cases were not prioritized.

**Keywords:** Test Case Prioritization, Test Prioritization Methods, Test Prioritization Comparison, Multiple Test Prioritization and Test Suite Prioritization, Genetic Algorithm.

**International Journal of Innovative Technology & Adaptive Management (IJITAM)**
**ISSN: 2347-3622, Volume-1, Issue-6 , March, 2014**

## 2. INTRODUCTION

Regression testing is the process of validating modifications introduced in a system during software maintenance. Regression testing is an expensive Process used to validate modified software. As the test suite size is very large, system retesting consumes large amount of time and computing resources. This issue of retesting of software systems can be handled using a good test case prioritization technique. A prioritization technique schedules the test cases for execution so that the test cases with higher priority executed before lower priority. The objective of test case prioritization is to detect fault as early as possible. Test case prioritization becomes a challenge in Component-based Software System (CBSS) which facilitates development of complex systems by integrating the reusable components. CBSS has emerged as an approach that offers rapid development of system using fewer resources and effort. The core idea of reuse and reducing the development costs can be achieved if the components offer reliable services. Thus, integration of components and testing become an important phase in CBSS. Integration of components involves understanding communication and coordination between the components. Developers do not provide the sufficient information on these components. As a result of this, understanding of component interactions while integrating these components becomes a challenge. Testing components is a challenging area of research. There have been troubles integrating the components. This in turn affects the quality and reliability of the software.

In the recent past, Component Based Software System (CBSS) has gained a very high importance. This is attributed to the reduction of cost and time in building the software using reusable components. A component is executable software having a published interface. The identified advantages of CBSS: Reduced lead time, enhanced quality. Developers are not provided with sufficient information on these components.

Test case prioritization techniques improve the cost-effectiveness of regression testing by ordering test cases such that those that are more important are run earlier in the testing process. Many prioritization techniques have been proposed and evidence shows that they can be beneficial. It has been suggested, however, that the time constraints that can be imposed on regression testing by various software development processes can strongly affect the behaviour of prioritization techniques. If this is correct, a better understanding of the effects of time constraints could lead to improved prioritization techniques, and improved maintenance and testing processes. Regression testing is almost universally employed by software organizations. It is important for software quality, but it can also be prohibitively expensive. For example, for a product, a regression test suite containing over 30,000 functional test cases that require over 1000 machine hours to execute. Hundreds of hours of engineer time are also needed to

**International Journal of Innovative Technology & Adaptive Management (IJITAM)**
**ISSN: 2347-3622, Volume-1, Issue-6 , March, 2014**

oversee this regression testing process, set up test runs, monitor testing results, and maintain testing resources such as test cases, oracles, and automation utilities.

An improved rate of fault detection during regression testing can let software engineers begin their debugging activities earlier than might otherwise be possible, speeding the release of the software. An improved rate of fault detection can also provide faster feedback on the system under test and provide earlier evidence when quality goals have not been met, thus allowing strategic decisions about release schedules to be made earlier than might otherwise be possible. Test case prioritization techniques improve the cost-effectiveness of regression testing by ordering test cases such that those that are more important are run earlier in the testing process. Prioritization can provide earlier feedback to testers and management, and allow engineers to begin debugging earlier. It can also increase the probability that if testing ends prematurely, important test cases have been run.

Below are some factors to consider in prioritizing test cases:

Mission-critical components (take advice from Requirements/Customer)

Complex features (take advice from development team/Requirements) Wherefailures would be most visible (take advice from development team)

Features that undergo frequent changes (take advice from development team) 5 Areas with past histories of problems (take advice from development team)Areas with complex coding (where were the developers most challenged)

Areas of most frequent use Major functionalities rather than going into detail New functionality.

### 3. PROBLEM STATEMENT

The aim is to have retesting of the system or component i.e. regression testing. Regression testing requires maximum effort, time and cost. Regression testing can last days or weeks and can involve substantial human effort. So the problem in hand is how to develop a technique like Test case prioritization which will lead to early fault detection. Further the question in hand is -

1) How to reduce the cost of regression testing, including regression test selection and test suite minimization techniques.

2) How can be test cases prioritized for execution so that the test cases with higher priority executed before lower priority with the help of genetic algorithm.

Software testing is an important activity of the software development process, and automated test case generation contributes to reduce cost and time efforts. The research on software testing problems has cantered mostly on software test optimization and intelligent software testing techniques are extensively proposed to solve software test

**International Journal of Innovative Technology & Adaptive Management (IJITAM)**
**ISSN: 2347-3622, Volume-1, Issue-6 , March, 2014**

optimization problems.

Several meta-heuristic search techniques such as applying rules, metaheuristics (like Genetic Algorithm (GA), Ant Colony Optimization (ACO), Tabu Search, Simulated Annealing, Bacteriologic Algorithm (BA), etc.), Fuzzy logic and Neural Networks (NN), Hybrid Genetic Algorithm (HGA) and other approximation methods have been proposed for some specific types of problems in test optimization and there is still a need for a more effective solution approach to more general problems such as test suite optimization and there is a need for developing models and efficient algorithms to achieve optimization by satisfying the specified test adequacy criteria.

#### 4. OBJECTIVES

Regression testing is the process of validating modifications introduced in a system during software maintenance. Regression testing is an expensive process used to validate modified software. As the test suite size is very large, system retesting consumes large amount of time and computing resources. This issue of retesting of software systems can be handled using a good test case prioritization technique. A prioritization technique schedules the test cases for execution so that the test cases with higher priority executed before lower priority. The objective of test case prioritization is –

1) To detect fault as early as possible.

2) Test case prioritization becomes a challenge in Component-based Software System (CBSS) which facilitates development of complex systems by integrating the reusable components.

3) Improve the Test Case prioritization techniques and the cost-effectiveness of regression testing by ordering test cases such that those that are more important are run earlier in the testing process.

4) Provide earlier feedback to testers and management, and allow engineers to begin debugging earlier through Prioritization.

5) To increase the probability that if testing ends prematurely, important test cases have been run.

6) The primary intension of this research work is to prioritize the regression testing test cases.

**7)** Focusing only on the particular test cases based on the prioritization will reduce the computation cost and time. The proposed method will effectively prioritize the regression test cases.

#### 5. PROPOSED APPROACH

Regression testing may take minutes to weeks to months of time depending on the size of the test suite and how long each test case takes to run. However, through the use of an effective prioritization technique, testers can reorder the test cases to obtain an increased rate of fault detection in the system, allowing corrections to be made earlier and raising overall confidence that the software has been adequately tested. If execution needs to be halted after some time

**International Journal of Innovative Technology & Adaptive Management (IJITAM)**
**ISSN: 2347-3622, Volume-1, Issue-6 , March, 2014**

period, a prioritized test suite is more likely to be more effective during that time period than would have been achieved using a random ordering. However, if the desired execution time to run the test cases is known in advance, a better test case ordering may be possible.

The work proposes, the genetic algorithm (GA) reordered the test cases using established criteria without knowledge of the faults in the system. The ordering was determined using test adequacy criteria, which predict how likely errors are to be found rather than how many errors the test actually finds or how many faults exist in the code. Focusing only on the particular test cases based on the prioritization will reduce the computation cost and time. The proposed method will effectively prioritize the regression test cases. Once the GA is completed, the new prioritized sub sequences of the provided unit test suites were executed on select Java programs. The "goodness" of these orderings was measured using an evaluation metric called An Average Percentage of Faults Detected (APFD) that will also be calculated.

**6. PROPOSED TOOLS**

- Rational Functional Tester
- Coverage analysis tool
- Genetic algorithm (GA)
- Evaluation metric called Average Percentage of Faults Detected (APFD)

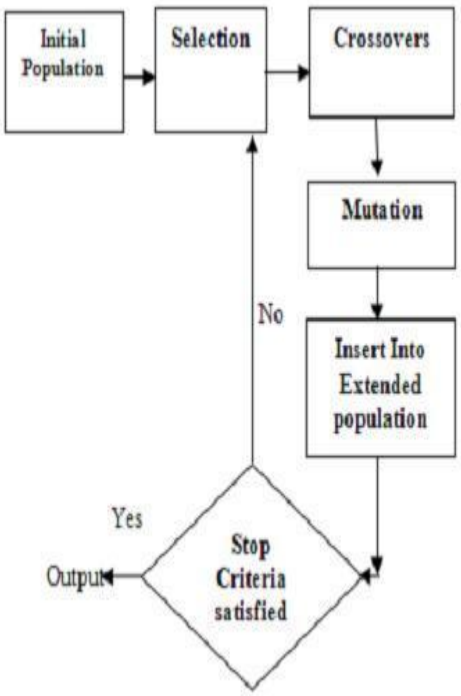
**7. GENETIC ALGORITHM**

The primary intension of the work is to prioritize the regression testing test cases.

In order to prioritize the test cases, some of the factors to be calculated. These factors will be used in the prioritization algorithm.

The factors are (1) customer assigned priority of requirements, (2) developer-perceived code implementation complexity, (3) changes in requirements, (4) fault impact of requirements, (5) completeness and (6) traceability (7) Execution time etc,. Then the Requirement Factor Value will be calculated based on these Based on these factors, a weight age will be assigned to each test case in the software. According to the weight age assigned, the test cases are prioritized. The prioritization is based on genetic algorithm (GA). The requirement factor values will be optimized by using the GA. The fitness can be calculated by using the factor values and weight age assigned. The test case with maximum fitness value will be elected as the high priority test case.

**International Journal of Innovative Technology & Adaptive Management (IJITAM)**
**ISSN: 2347-3622, Volume-1, Issue-6 , March, 2014**




Regression testing is one of the most critical activities of software development and maintenance. Whenever software is modified, a set of tests are run and the comparison of new outputs is done with the older ones to avoid unwanted changes. If new output and old output match it implies the modifications made in one part of the software didn't affect the remaining software. It is not appropriate to re-execute every test case for a program if changes occur. The problem of regression test selection can be solved by prioritizing test cases. Regression test prioritization techniques reorder the execution of a test suite in an attempt to ensure that defects are revealed earlier in the test execution phase.

For this reason, researchers have considered various techniques for reducing the cost of regression testing, including regression test selection and test suite minimization techniques. Regression test selection techniques reduce the cost of regression testing by selecting an appropriate subset of the existing test suite, based on information about the program, modified version, and test suite. Test suite minimization techniques lower costs by reducing a test suite to a minimal subset that maintains equivalent coverage of the original test suite with respect to a particular test adequacy criterion.

## 8. CONCLUSION

Test case prioritization is a method to prioritize and schedule test cases. The technique is developed in order to run test cases of higher priority in order to minimize time, cost and effort during software testing phase. The review shows that many researchers propose many methods to prioritize and reduce the effort, time and cost in the software testing phase, such as test case prioritization methods, regression selection techniques and test case reduction approaches. This paper concentrates on test case prioritization techniques only. This paper shows that there are many prioritization techniques researched between 1998 and 2008. With the existing techniques, this study introduces a new "4C" type of test case prioritization techniques, which are: (a) customer requirement-based techniques, (b) coverage-based techniques, (c) cost effective-based techniques and (d)

**International Journal of Innovative Technology & Adaptive Management (IJITAM)**
**ISSN: 2347-3622, Volume-1, Issue-6 , March, 2014**

chronographic history-based techniques. First, the customer requirement-based techniques are methods to directly prioritize test cases from requirement specifications. Second, the coverage based techniques are structural white-box testing techniques. They compare test program behaviour against the apparent intention of the source code.

However, this paper suggests the following future works to improve the capability of those two methods: (a) apply the practical prioritization weight values for commercial systems (b)improve the ability to automatically find duplicate test cases with the same values (c) improve the ability to automatically prioritize multiple large test suites with real commercial data.

## 9. REFERENCES

*[1] K. Onoma, W-T. Tsai, M. Poonawala, and H. Suganuma. Regression testing in an industrial environment.Communications of the ACM, Vol. 41, No.5, pp.81-86, May 1988.*

*[2] H. K. N. Leung and L. White. Insights Into Regression Testing. In Proceedings of the Conference on Software Maintenance, pages 60-69, October 1989.*

*[3] B. Beizer. Software Testing Techniques. Van Nostrand Reinhold, New York, NY, 1990.*

*[4] H. K. N. Leung and L. J. White. A study of integration testing and software regression at the integration level. In Proceedings of the Conference on Software Maintenance, pp. 290-300, November 1990.*

*[5]M. J. Harrold, R. Gupta, and M. L. Soa.A methodology for controlling the size of a test suite.ACM Transactions on Software Engineering and Methodology, Vol. 2, No.3, pp.270-285, July 1993.*

*[6] Y. F. Chen, D. S. Rosenblum, and K. P. Vo. TestTube: A system for selective regression testing. In Proceedings of the 16th International Conference on Software Engineering, pp. 211-222, May 1994.*

*[7] D. Binkley. Semantics guided regression test cost reduction. IEEE Transactions on Software Engineering, Vol. 23, No. 8, pp. 498-516, August 1997.*

*[8] T. Y. Chen and M. F. Lau. Dividing strategies for the optimization of a test suite. Information Processing Letters, Vol. 60, No. 3, pp. 135-141, March 1996.*

*[9] G. Rothermel and M. J. Harrold. Analyzing regression test selection techniques. IEEE Transactions on Software Engineering, Vol. 22, No. 8, pp. 529-551, August 1996.*

*[10] G. Rothermel and M. J. Harrold. A safe, efficient regression test selection technique. ACM Transactions on Software Engineering and Methodology, Vol. 6, No. 2, pp.173-210, April 1997.*

*[13] F. I. Vokolos and P. G. Frankl. Pythia: a regression test selection tool based on textual differencing. In Proceedings of the 3rd International Conference on Reliability, Quality & Safety of Software-Intensive Systems (ENCRESS '97) , May 1997.*

*[14] W. E.Wong, J. R. Horgan, A. P. Mathur, and A. Pasquini. Test set size minimization and fault detection effectiveness: A case study in a space application. In Proceedings of the 21st Annual International Computer Software & Applications Conference, pp. 522-528, August 1997.*

*[15] W. E. Wong, J. R. Horgan, S. London, and H. Agrawal. A study of effective regression testing in practice. In Proceedings of the Eighth International Symposium on Software Reliability Engineering, pp 230-238, November 1997.*

*[16] T. Ball. On the limit of control flow analysis for regression test selection. In Proceedings of the ACM International Symposium on Software Testing and Analysis, pp. 134-142, March 1998.*

*[17] W. E. Wong, J. R. Horgan, S. London, and A. P. Mathur. Effect of test set minimization on fault detection effectiveness. Software-Practice and Experience, Vol. 28, No.4, pp. 347-369, April 1998.*

*[18] G. Rothermel, M. J. Harrold, J. Ostrin, and C. Hong. An empirical study of the effects of minimization on the fault detection capabilities of test suites. In Proceedings of the International Conference on Software Maintenance, pp. 34-43, November 1998.*

*[19] G. Rothermel, R. H. Untch, C. Chu, and M. J. Harrold. Test case prioritization: An empirical study. In Proceedings of the International Conference on Software Maintenance, pp. 179-188, August 1999.*

*[20] R. V. Binder, Testing object-oriented systems Reading, Mass.: Addison Wesley, 2000.*

*[21] Praveen RanjanSrivastava, "TEST CASE PRIORITIZATION", Journal of Theoretical and Applied Information Technology, pp. 178-181, 2008.*

*[20] K.K. Aggarwal, Y. Singh, Software engineering programs documentation, operating procedures, third edition, New Age International Publishers, New Delhi, 2008.*

*[21] R.Krishnamoorthi and S.A.Sahaaya Arul Mary, "Regression Test Suite Prioritization using Genetic Algorithms", International Journal of Hybrid Information Technology Vol.2, No.3,pp. 35-52, July, 2009.*

*[22] Yogesh Singh, ArvinderKaur, BhartiSuri, "Empirical Validation of Variable based Test Case Prioritization/Selection Technique", International Journal of Digital Content Technology and its Applications Vol. 3, No. 3,*

**International Journal of Innovative Technology & Adaptive Management (IJITAM)**
**ISSN: 2347-3622, Volume-1, Issue-6 , March, 2014**

*September 2009.*

*[23] Yogesh Singh, ArvinderKaur, BhartiSuri, "A Hybrid Approach for Regression Testing in InterproceduralProgram", Journal of Information Processing Systems, Vol. 6, No.1, March 2010.*

*[24] RuchikaMalhotra, ArvinderKaur, Yogesh Singh, "A Regression Test Selection and Prioritization Technique", Journal of Information Processing Systems, Vol.6, No.2, June 2010.*

*[25] Sujata, Mohit Kumar, Dr.Varun Kumar, "Requirements based Test Case Prioritization using Genetic Algorithm", IJCST Vol. 1, No. 2, pp.189-191, December 2010.*

*[26] Arup AbhinnaAcharya, Durga Prasad Mohapatra, Namita Panda, "Model Based Test Case Prioritization for Testing Component Dependency in CBSD Using UML Sequence Diagram", International Journal of Advanced Computer Science and Applications, Vol. 1, No. 6,pp. 108-113, December 2010.*

*[27] Amrita Jyoti, Yogesh Kumar Sharma, AshishBagla, D. Pandey, "Recent Priority Algorithm In Regression Testing", International Journal of Information Technology and Knowledge Management, Vol. 2, No. 2, pp. 391-394, July-December 2010.*