A Hybrid Density Based K Mean for Test Case Prioritization using Prism Algorithm

Meenakumari, Shivi Sharma

Dept. of CSE, L.R.I.E.T Solan, HPTU, India

ABSTRACT: Software testing is important phase of software development life cycle which ensures the developer that the developed software works according to specifications or not. Prioritization techniques that incorporate a clustering approach and utilize code coverage, code complexity and history faults as well to increase the effectiveness of the prioritization. To make testing efficient and effective a techniques of test case prioritization are used. An efficient Test case prioritization technique reduces the cost of testing and fault detection capabilities of testing. Results showthat test case prioritization that utilizes a clustering approach can improve the effectiveness of test case prioritization techniques.

KEYWORDS: Software Testing, Regression Testing, Test case prioritization, Test Case Prioritization Technique.

I.   INTRODUCTION

Software testing is done with an attention to find out the defect in existing software. Software testing has been proven that testing , analysis and debugging costs usually consume over 50 % of the costs associates with the development of large software system.[1] Software testing is important phase of software development life cycle which ensure the developer that the developed software works according to specification or not. In other words software testing is verification and validation process of computer program or application/product to meet the requirements that guided its design and development , works as expected , can be implemented with the same characteristics , and satisfies the need of stakeholder . A defect is inconsistency among the expected and actual result. The defect's ultimate source may be traced to a fault introduced in the specification, design, or development phase. [2] Software testing is one element of a broader topicthat is often referred to as verifying and validating that a software application or program. Software testing is useful for finding the defects, fundamental weakness in the application code that must be improved or checked.

Software testing has three main purposes: **verification, validation**, and **defect:**
*   The process of verification confirms that software meets its specifications. It ensures that software correctly implemented for specific function.
*   Whereas the process of validation ensures that the software meets the business requirements. It provides the traceable activities to customers.
*   A defect is inconsistency among the expected and actual result. The defect's ultimate source may be traced to a fault introduced in the specification, design, or development phases.
*   In software testing there are many type of testing are used.The one most important type is regression testing.[3]

Regression testing concentrates on finding defect after a major code change has occurred. Specifically,it exposessoftware regression or old bugs that have reappeared.Regression testing is process of retesting the modified software and ensures that new error does not introduce into the previously tested source code due to these modifications.Regression testing is very expensive testing process. In order to decrease the cost of regression testing thesoftware tester may prioritize the test case so that the test case which are more important as run earlier during regression testing process. [2]

Considering the software development and maintenance process one of the most critical activities is regression testing. Test suites which are developed by software developers for their software have being saved by them. These test suites are then used in the form of regression testing. Reuse of test suite and other activities related to regression testing have one half of the cost of software maintenance. Regression testing is very expensive but ensures that the software

program will work according to its specification after changes have been made to it. Not only the cost these activities can consume an inordinate amount of time.

One of the approaches to overcome these problems testers runs those test cases first which have highest priority according to some criterion. This approach of regression testing is called test case prioritization.

Regression Testing Goal: Testing modified softwareto ensure that changes are correct and do not   adversely affect other parts of the software

- Make use of existing test cases developed for previous versions of the software
- May have to create new test cases as well

Why is Regression Testing Important?

- Software is buggy
- Software is modified over time
- Adding new functionality
- Improving performance

Various approaches of regression testing are:

- Retest All
- Regression Test Selection
- Test Suite Reduction
- Test Case Prioritization
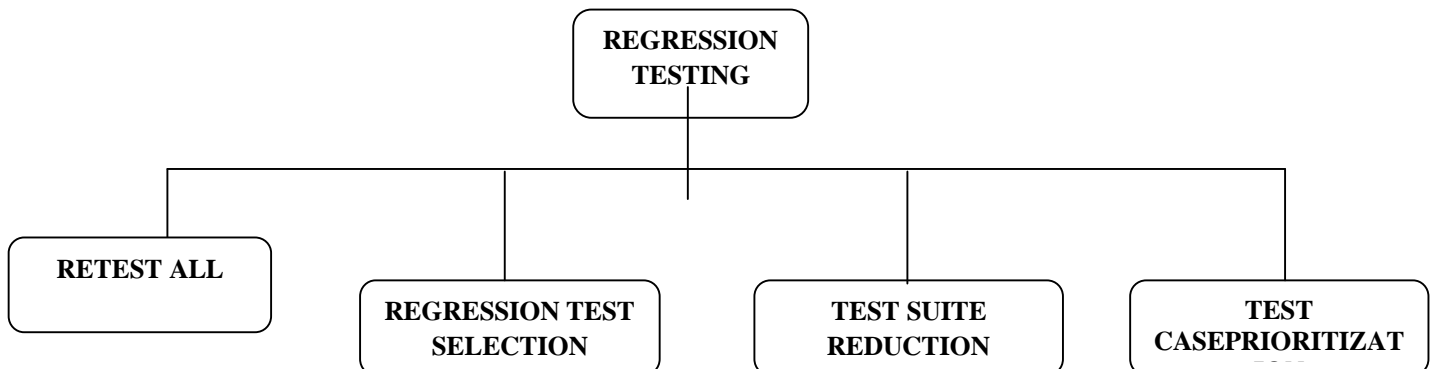
- Retest All - It is the most straightforward approach of regression testing. In this simply execute all the existing test cases in the test suite.[7]
- Regression Test Selection-Retest all technique takes time and effort as all test cases are used to test the program again, so may be quite expensive. This technique much better as it uses information about program, modified program, test cases to select subset of test cases for testing.[5]
- Test Suite Reduction – This technique uses information about program and test suite to remove the test cases, which have become redundant with time, as new functionality is added. It is different from Regression test selection as former does not permanently remove test cases but selects those that are required. Advantage of this technique is that it reduces cost of validating, executing, managing test suites over future releases of software, but the downside of this is that it might reduce the fault detection capability with the reduction of test suite size[5]
- Test Case Prioritization in this technique each test caseis assigned a priority. Priority is set according to some criterion and test cases with highest priority are scheduled first For example criterion may be that the test case which has faster code coverage gets the highest priority.Advantage to previous techniques is that it doesn't discard or permanently remove the test cases from test suite. Another criterion may be rate at which fault is detected.[5]

Test case prioritization: Test case prioritization technique prioritize and schedule test cases in an order that attempts to maximize some objective function.[1]Test case prioritization technique offer an alternative approach to improving

International Journal of Innovative Research in Computer and Communication Engineering

regression testing cost effectiveness. Test case prioritization technique to increase the chance of early fault detection using various types of information available from software artifacts.

Test case prioritization is a strategy for improving regression testing.The most important concerns in the testing phase are time and cost. Test case prioritization is a classification type of regression testing that helps us to make the process of testing a cost and time effective task. It also increase the efficiency of the testing procedure by prioritizing the test cases according to desired criteria.[3]Test cases are used to determine whether an application or software system is working correctly or not. It is difficult to re-execute every test case for a program if changes occur. To reduce the cost of regression testing,testers will prioritize the test cases. The main aim of test case prioritization is toincrease the rate of fault detection. Fault detection rate is a measure to find out, howquickly faults are detected within the testing process. An improved rate of fault detection can provide earlier feedback for earlier debugging. In order to enhance the effectiveness of the test case ordering and to improve the rate of fault detection, two approachesnamely "test case prioritization using clustering approach is proposed.[6]

There exist different types oftest case prioritization methods:

- **Code-based test caseprioritization:** Incode-based test case prioritization, source code of the systemis used to prioritize the test cases. Most of the test caseprioritization methods are code based. [4]
- **In model-based testcase:** Prioritization a system's model is used to prioritize thetest cases. System models are used to capture some aspects ofthe system behavior. The model based test case prioritizationmay improve the early fault detection as compared to thecode-based test case prioritization. Model-based testprioritization may be an inexpensive alternative to the existingcode-based test prioritization methods. However, model-basedtest case prioritization may be sensitive to the correct/incorrectinformationprovided by the testers/developers. Hence modelbasedtest case prioritization is the best one compared to codebased test case prioritization [4].
-

Test case technique:
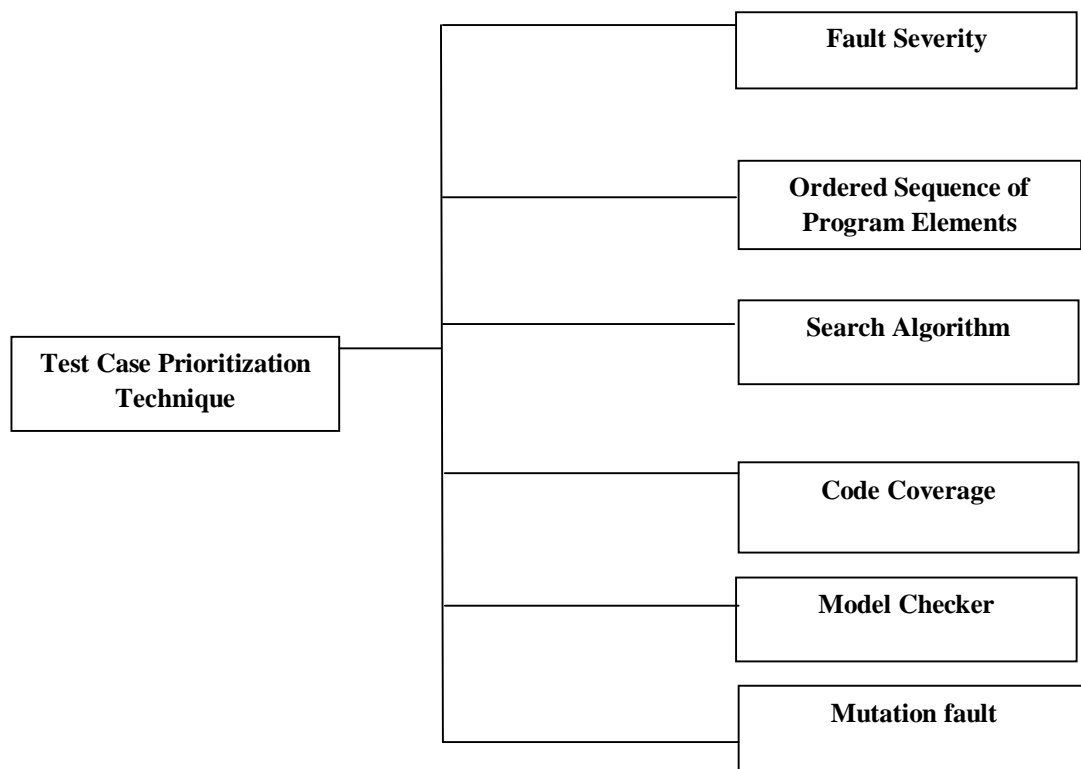
Test case prioritization presents the [3] average faults per minute are calculated, based on the fault detection rate the test cases are prioritized. This paper presents a new algorithm to prioritize the test cases. In this algorithm average fault

per minute is calculated as follows, AF/m Where F represents fault and Tcost represents as time. Based on that value the algorithm prioritizes the test case. Value based regression test case prioritization is used to detect the faults in early. It proposed an algorithm to prioritize the system test cases based on six factors like (1) Customer priority (2) Changes in requirement (3) Implementation complexity (4) Requirement traceability (5) Execution time (6) Fault impact. Classification based on the characteristics of the prioritization algorithms-

- Based on customer requirements
- Based on coverage
- Based on cost effective
- Based on chronographic history

Based on customer requirements-In these techniques various customer requirement factor are considered. Assign some weight to these factors. Test cases having high weight value are executed first and test cases with low weight value are executed later.

Based on coverage-For detecting faults earlier in testing, we have to achieve more coverage. These techniques test internal structure of data and may be consider as white box testing.

Based on cost effective-There are much kind of cost related to test cases like cost of analysis and cost of prioritization. In cost effective based techniques test cases are ordered for execution based on cost.

Leung and white propose a cost modal that compare the various regression strategies. They divide the total cost into two parts

- Direct cost
- Indirect cost

Direct cost includes
System analysis cost Ca
Test selection cost Cs
Test execution costCe
Result analysis cost Cr

Indirect cost includes
Overhead cost
Tool development cost

One disadvantage of this technique is that they did not include the cost of undetected faults.They used the cost factors like

- Ca (T) cost of analysis
- $Ce(T)$ cost of execution
- $Cc(T)$ cost of result checking
- $Cs(T)$ cost of selection
- Cm (T) cost of maintenance of the test suite

In experiment for test case prioritization they consider two factors cost of analysis and cost of prioritization Cp (T). In their work they divide the testing process in two phases one preliminary and second critical phase. These two phases have different costs. The result shows that the optimal ordering, total function coverage and additional function coverage have maximum saving.

Based on chronographic history-

In these type of prioritization techniques test execution history considered to be the main factor for prioritization oftest cases. Jung-Min-Kim and Adam Porter proposed a history based test case prioritization technique. It is for regression testing in resource constrained environment. Their main motive behind this is to show that historical information can be useful for decreasing the cost and it may be useful increasing the effectiveness of testing process. [7]

II.   CONCLUSION

To reduce this cost by prioritizing test cases and running the tests for the selective test cases as per the available time and manpower. There are a number of test cases available which can consume a lot of time and effort. A selective number of test cases needs to be selected which would be otherwise used for the same purpose. The priorities of the test cases need to be decided on the basis of several parameters. The parameters for the test case prioritization need to be

chosen and a model needs to be developed which would set priority among the test cases. First of all a data set needs to be generated which would be utilized for our proposed algorithm testing. Then the dataset needs to be pre-processed for outlier removal and redundancy removal. Then a technique for clustering of the test cases needs to be developed which would be utilized for the above mentioned problem. The parameters has to be selected such that the code coverage remains almost the same while there must be significant reduction in the number of test cases which implies a reduction in the total testing effort.

REFERENCES

[1]SiripongRoongruangsuwan, JirapunDaengdej, **"**Test Case Prioritization Techniques",*Journal of Theoretical and Applied Information Technology,2010.*

[2]MedhunHashiniD.R, "Clustering Approach to Test Case Prioritization Using Code Coverage Metric",*International Journal of Engineering And Computer Science*, Volume 3 Issue 4, 2014.

[3] Ashima Singh, " Prioritizing Test Cases in Regression testing using   Fault Based Analysis", *International Journal of Computer Science Issues*, Vol.9, Issue 6, No 1, 2012**.**

[4]SanjuktaMohanty, Arup AbhinnaAcharya, Durga Prasad Mohapatra, SanjuktaMohanty, "A Survey On Model Based Test Case Prioritization", *International Journal of Computer Science and Information Technologies*, Vol. 2 (3), 2011.

[5] Praveen RanjanSrivastava, "Test Case Prioritization"*Journal of Theoretical and Applied Information Technology, 2008.*

*[6]*K.Sandya "Test Case Prioritization Using Clustering Approach And Multiple Modular Based Approach"*International Conference* 2012.

[7]Monika, Ajmer singh "Test Case Prioritization: A Review*", International Journal of Engineering Research & Technology* Vol. 3 Issue 5, 2014.