

A REVIEW ON TEST CASE PRIORITIZATION FOR REGRESSION TESTING

Yamini Pathania¹, Poonam Kumari², Nikita Bakshi³

Post Graduate Student, Dept. of Computer Science Engineering, Chandigarh University, Punjab, India

Abstract: Regression Testing is the process of retesting the modified parts of the software and checking that no new faults have been created into already existing code. When new features are added to an existing software system, then regression testing is necessary to test the new features as well as the existing features to ensure that their behaviors are not affected by the modifications. 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. Testers will prioritize the test cases to reduce the cost of regression testing. The main purpose of test case prioritization is to increase the rate of fault detection. There are different types of tools and techniques used to increase the rate of fault detection using test case prioritization for regression testing.

Keywords: Software testing, Regression testing, Test case prioritization, Tools.

I. INTRODUCTION

Software testing is the part of software development life cycle. Software testing may define as "A process of executing a program with the goal of finding errors". Testing means that one inspects behaviour of a program on a finite set of test cases for which valued inputs always exist. Software testing performed to find the defect in the existing system. It is the process of verifying and validating that computer program or application/product meet the requirements that guided its design and development, works as expected and also satisfies the needs of stakeholder. Software testing is also the process of evaluation a software item to detect differences between given input and expected output also to assess the feature of a software item. Software testing is an important component of software quality assurance, and there are almost many software organizations which spending up to 40% of their resources on testing. Regression testing concentrates on finding defects after a major code change has occurred. It exposes software regressions or old bugs that have reappeared after doing any change in the software.

A. Methods of Software Testing:

Software Testing consist different methods, which are white box testing, black box testing and gray box testing.

a) White box testing: is also called glass testing, open box testing, structural testing and transparent box testing. In white box testing, the tester needs to have knowledge of the internal working of the code. Optimization of code and extra lines of code can be removed which can bring in hidden defects. But

it is difficult to maintain white box testing as the use of specialized tools like code analyzers and debugging tools are required. Bugs can often be found before they cause trouble; so, this testing is extremely effective in detecting the problems, resolving problems.

b) Black box testing: is testing software based on output requirements and without any knowledge of the internal structure or coding in the program. Typically, when performing a black box testing, a tester will interact with the system's user interface by providing inputs and examining outputs without knowing how and where the inputs are worked upon. Advantages of black box testing are: Well suited and efficient for large code segments and code access not required. But There is limited Coverage since only a selected number of test scenarios are actually performed.

c) Gray box testing: is defined as testing software while already having some knowledge of its underlying code or logic. Grey box testers don't rely on the source code; instead they rely on interface definition and functional specifications. Testing is done from the point of view of the user and not the designer. But in this, testing every possible input stream is unrealistic because it would take lot of time, therefore many program paths will go untested.

B. Levels of testing:

Levels of testing include the different methodologies that can be used while conducting software testing. Software testing is divided into two levels, which are:

- Functional Testing.
- Non- functional Testing.

Functional testing: It is based on the specifications of the software that is to be tested. Functional Testing of the software is conducted on whole system (i.e. integrated or complete system) to evaluate the system's compliance with its specified requirements. It is a type of black box testing and also a quality assurance process. This testing having different types of testing: Unit testing, Acceptance testing, System testing, regression testing etc. There are six steps to perform functional testing:

- The determination of the functionality that the intended application is meant to perform.
- The creation of test data based on the specifications of the application.
- The output based on the test data and the specifications of the application.
- The writing of Test Scenarios and the execution of test cases.
- The comparison of actual and expected results based on the executed test cases.

- To check whether the application works as per the customer need.

Non Functional Testing: Non-functional testing of software involves testing the software from the requirements which are non-functional in nature related but important a well such as performance, security, user interface etc. The names of many non-functional tests are often used interchangeably because of the overlap in scope between various non-functional requirements. Non-functional testing consist different types of testing: performance testing, portability testing, usability testing, security testing etc.

II. REGRESSION TESTING

Regression testing is a type of software testing that intends to ensure that changes to the software have not affected it. It may include in the functional testing. Regression testing may define as “it is the testing process that is done to find the regression in the system after doing any changes in the product or it is used to check whether the new changes occurs the errors in the software or not”. For example, before applying any change on program it must be tested, then again retested the program in the selected areas after a change is applied, to detect whether the change created new bugs or issues, or it may achieve its intended purpose after the actual change is made. Thus, regression testing is essential for large organization. It can be performed during any level of testing. Research in regression testing is a wide range of topics. A test case is a single step, or occasionally a sequence of steps, to test the correct functionalities, features of an application. Earlier work in this area investigated different environments that can assist regression testing. Whenever the software is modified/changed, then regression testing is performed to check whether the new modification on the software can introduce any new errors or not, on the existing software. It is Impractical to re-execute every test case for program if change occurs. It takes lot of time, cost and resources to re-execute all test cases. There are various techniques are commonly used to reduce complexity of regression testing. Automation testing mainly used in regression testing and apart from regression testing, it is also used to test the application from load performance and stress point of view. Automation testing increases the test coverage; improve accuracy, saves time as well as money in comparison to manual testing. Retest all, Test case minimization, test case selection, test case prioritization are the important techniques to improve the effectiveness of regression testing.

A. Retest - All

In this method, the test cases that no longer apply to modified version of program are discarded and all the remaining set of test cases are used to test the modified program.

B. Test Case Minimization

It is a process that seeks to identify and then eliminate the redundant test cases from the test suite.

C. Test Case Selection

Test case selection deals with the problem of selecting a

subset of test cases that will used to test the changed parts of the software.

D. Test Case prioritization

Test case prioritization concerns the identification of the perfect ordering of test cases. We will further discuss test case prioritization in detail in our next section.

III. TEST CASE PRIORITIZATION

The prioritization of test cases depending on business impact, and frequently used functionalities. In this method, selection of test cases based on priority will greatly reduced the regression test suite. There are number of matrix available to calculate the fault detection rate using test case prioritization technique.

A. Goals

There are several aspects of the test case prioritization problem that are worth describing further. Following are possible goals of prioritization:

- Testers may wish to increase the rate of fault detection of a test suite.
- Testers may wish to increase the coverage of coverable code in the system under test at a faster rate, so proving a code coverage criterion to be met earlier in the test process.
- Testers may wish to increase their confidence in the reliability of the system under test at a faster rate.
- Testers may wish to increase the rate at which high risk faults are detected by a test suite, thus positioning such faults earlier in the testing process.
- Testers may wish to increase the likelihood of revealing faults related to specific code changes earlier in the regression testing process.

The main goal of the test case prioritization schedule test cases in order to increase their ability to meet some performance goal: Rate of fault detection, Rate of code coverage and rate of increase of confidence in reliability. The rate of fault detection which is a measure of how quickly the fault is detected so that during testing faster feedback can provide about system under testing and allow the software tester to correct the software at earlier phase as possible.

B. Techniques

There are several techniques to prioritize test cases for regression testing, including:

- Techniques that order test cases based on their total coverage of code components,
- Techniques that order test cases based on their coverage of code components not previously covered, and
- Techniques that order test cases based on their estimated ability to reveal faults in the code components that they cover.
- Coverage information
- Code complexity
- History data etc.

C. Advantages of test case prioritization

Test case prioritization is used to arrange and execute the test cases orderly to save cost and time. Test case prioritization is more efficient and widely used by the testers. Many researchers introduced more methods for test case prioritization in regression testing. Test case prioritization for early fault detection has many main advantages.

- First, if a bug is found early in the regression testing process, debugging can start earlier and hence bugs could be fixed faster.
- Minimization of testing costs.
- Identification of high risk defects earlier.
- Increase reliability.

D. Disadvantages of test case prioritization

Although many techniques have been proposed for test case prioritization, but there are some complexities exist in it. They are:

- To identify defects on poor coding software.
- Determine the importance of test case between same priority test cases.
- Determination of important test case when multiple test suites consist of multiple test cases.
- No common techniques to cover multi-coverage criteria.

E. Tools

Many tools are available which can be used to automate regression testing for test case prioritization. All tools consists different functionality, but MATLAB is the most populated tools used in these days. These tools are:

- Selenium: It is a portable software testing framework for web applications. It is open-source software, which can be downloaded and used without charge. Selenium deploys on Windows, Linux, and Macintosh platforms. The tests can then be run against most modern web browsers. Selenium also provides a test domain-specific language (Selenese) to write tests in a number of popular programming languages, including Java, Perl, PHP, Python, C#, Groovy, and Ruby and also provides a record/playback tool for authoring tests without learning a test scripting language (Selenium IDE).
- IBM Rational Functional Tester: It is an automated functional testing and regression testing tool. It is primarily used by Software Quality Assurance teams to perform automated regression testing. It is a software test automation tool used by quality assurance teams to perform automated regression testing. It allows users to create tests that mimic the actions and assessments of a human tester. Rational Function Tester supports a range of applications, such as web-based, Siebel, SAP, Adobe PDF documents, .Net, zSeries, iSeries, and pSeries, Java, etc.
- Testing Anywhere: - It is software produced by San Jose-based Automation Anywhere, it allows testers and developers to test applications, controls, Web sites,

objects and GUI front-ends. Testing Anywhere is used to schedule, record, debug, and run test cases for a wide variety of application types, such as Java, mainframe, C++, Silverlight, .Net, etc. In this software, test cases are created using one of five methods (object recording, Smart recording, web recording, image recognition, and editor). These test cases are recorded, saved, and can be edited and enhanced. Wizards make it possible for people with no programming skills to create and edit these test cases. Executable (EXE) files can be created so that testers can deploy (run) them on remote machines. A workflow designer creates high-level business and IT processes and enables the ability to manage them.

- WinRunner: - WinRunner, Mercury Interactive's enterprise functional testing tool. Winrunner helps you automate the testing process, from test development to execution. . It is used to quickly create and run sophisticated automated tests on your application. Prior to a software release, you can run these tests in a single overnight run- enabling you to detect and ensure superior software quality.
- HP Quick Test Professional: - This software, formerly known as HP Unified Functional Testing (UFT), which provides functional and regression test automation for software applications and environments. It can be used for endeavour quality assurance. HP Quick Test Professional supports keyword and scripting interfaces and features a graphical user interface.
- MATLAB: - It stands for MATrix LABoratory. This software is built up around vectors and matrices, which makes the software particularly useful for linear algebra but MATLAB is also a great tool for solving algebraic and differential equations and for numerical integration. It has powerful graphic tools and can produce nice pictures in both 2D and 3D. MATLAB is one of the easiest programming languages for writing mathematical programs. MATLAB also has some tool boxes useful for signal processing, image processing and optimization.
- Weka: - It stands for Waikato Environment for Knowledge Analysis. This tool is a popular suite of machine learning software written in Java. Weka is free software available under the GNU General Public License. Weka supports several standard data mining tasks, particularly supports data pre-processing, clustering, regression, classification, visualization, and feature selection. So, advantages of this software are:
 - It is portability, since it is fully implemented in the Java programming language and thus runs on almost any modern computing platform.
 - A comprehensive collection of data pre-processing and modeling techniques.
 - Ease of use due to its graphical user interfaces.

IV. 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. This technique is used in artificial intelligence as well as in different clustering algorithm to increase their effectiveness and fault detection. This paper describes different tools used in test case prioritization for regression testing. The main goals of test case prioritization are rate of fault detection, rate of code coverage and rate of increase of confidence in reliability.

REFERENCES

- [1] G Rothermel, R Untch, C Chu and M Harrold, "Test case prioritization: An empirical study", IEEE International conference on Software Maintenance (ICSM), IEEE, 1999.
- [2] Rothermel, Gregg, et al. "Prioritizing test cases for regression testing", IEEE Transactions on Software Engineering, 2001.
- [3] Elbaum, Sebastian, Alexey G. Malishevsky, and Gregg Rothermel. "Test case prioritization: A family of empirical studies", IEEE Transactions on Software Engineering, 2002.
- [4] Srivastava, Praveen Ranjan, "Test case prioritization", Journal of Theoretical and Applied Information Technology, 2008.
- [5] Liela Naslavsky, Hadar Ziv, Debra J. Richardson, "A Model based Regression Test Selection Technique", International Conference on Software Maintenance (ICSM), IEEE, 2009.
- [6] Carlson, Ryan, Hyunsook Do, and Anne Denton, "A clustering approach to improving test case prioritization: An industrial case study", 27th IEEE International Conference on Software Maintenance (ICSM), IEEE, 2011.
- [7] Jiang, Bo, et al. "How well do test case prioritization techniques support statistical fault localization", 33rd Annual IEEE International Conference on Computer Software and Applications Conference, 2009. COMPSAC'09, IEEE, 2009.
- [8] J. Wang, Y. Zhuang, C. Jianyun, "Test Case Prioritization Technique based on Genetic Algorithm", International Conference on internet Computing and information Services, Hong Kong , 2011.
- [9] Huang, Yu-Chi, Kuan-Li Peng, and Chin-Yu Huang, "A history-based cost-cognizant test case prioritization technique in regression testing.", Journal of Systems and Software, 2012.
- [10] Arafeen, Md Junaid and Hyunsook Do, "Test case prioritization using requirements-based clustering", IEEE Sixth International Conference on Software Testing, Verification and Validation (ICST), IEEE, 2013.
- [11] P Kandil, S Moussa, N Badr, "Regression testing approach for large scale", IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW), 2014.
- [12] Manika Tyagi and Sona Malhotra, "Test case prioritization using multi objective particle swarm optimizer", 2014 International Conference on Signal Propagation and Computer Technology (ICSPCT), IEEE, 2014.
- [13] Muthusamy, Thillaikarasi, "A New Effective Test Case Prioritization for Regression Testing based on Prioritization Algorithm", International Journal of Applied, 2014.
- [14] Tyagi, M.; Malhotra, S. "Test case prioritization using multi objective particle swarm optimizer", International Conference on Signal Propagation and Computer Technology (ICSPCT), 2014.
- [15] Hettiarachchi, Charitha, Hyunsook Do, and Byoungju Choi, "Effective Regression Testing Using Requirements and Risks", 2014 Eighth International Conference on Software Security and Reliability, IEEE, 2014.