INTERNATIONAL JOURNAL FOR TRENDS IN ENGINEERING & TECHNOLOGY
VOLUME 4 ISSUE 1 – APRIL 2015 - ISSN: 2349 - 9303

Prioritizing Test Cases for Regression Testing A Model Based Approach

Vinothkumar.N[1]
[1]Sona College of Technology,
Department of CSE.
vinothkumaaraa@gmail.com

Galeebathullah.B[2]
[2]Sona College of Technology,
Department of CSE.
kalifathullah@gmail.com

Abstract— Testing is an important phase of quality control of Software Development Life Cycle (SDLC). There are various types of testing methodologies involved to test the application. Regression Testing is a type of testing, which is done to ensure whether the modified features or bug fix had an impact over the existing functionality. Defects are identified by executing the set of test cases. Regression Test case selection is not at all possible to conclude how much retesting is required to identify the deviation when the test suites are larger in size. Prioritization of test cases is done to change the order of test case execution based on the severity. In the proposed a model based approach prioritization of test cases are generated based on UML diagrams (Sequence and State Chart). The modified features have the reflection in the model generation and the number of states and transitions covered. Prioritized test cases are then clustered based upon the severities using dendragram approach. It leads to decrease in the time and cost of regression testing.

Index Terms— Model Based Testing, Test cases, Regression Testing, Software Engineering, Test Case Prioritization.

1 INTRODUCTION

SOFTWARE Testing is one of the important activities in software development process. Depending on the testing method employed, software testing can be implemented at any stage in the development process. Traditionally, most of the test effort occurs once the requirements had been collected and the coding process has completed. Regression testing is used to verify the correctness of the software under development and in maintenance process. It is very inefficient and time consuming process to re-execute every test case in regression testing to find the small changes over the application. Hence model based test case prioritization is used to schedule the test cases in a specific order that maximizes the testing efficiency.

In model based prioritization the regression test cases are prioritized from the models and test cases are clustered based on its severity factor. The test effort is on-going and test cases are prioritized from the models before the coding process begins. This technique involves constructing a graph model for the source code to represent those controls, data dependence as well as object oriented relations such as inheritance, polymorphism, data abstraction and message passing. Model based approach for test case prioritization involves model augmentation with the information such as message paths and object states which are available from the UML design models. Changes in the model will be reflected in UML diagrams where states can be compared to identify the absence of any functionality. Forward slice of the constructed model is used to identify all the possible states in the program that may be affected during the modification or bug fix. Test information available from the intersection of the forward slice of the model and the changed state is then used to prioritize the test cases.

- *Vinothkumar. N is currently pursuing a masters degree program in software engineering in Sona College of Technology, India, PH-9865123410. E-mail: vinothkumaaraa@gmail.com*
- *Galeebathullah.B is currently working as Assistant Professor in computer science and engineering in Sona College of Technology, India, PH-9790722912. E-mail: kalifathullah@gmail.com*

Test case prioritization using a model based approach has opened up new opportunities for software testing, which provides prioritizing a large number of test cases by clustering factor. It reduces the execution time of retesting of large applications and lead to cost-effective solutions. However, it also explores challenges such as test automation and a lack of testing deadlines to be met.

2 TEST CASE

Test Case is defined as a set of conditions or variables with step by step procedure in a sequential flow to test the application. It contains the set of inputs, expected output and actual output. The strategy which is used to measure whether a software program or system has passed or failed in a test is stored in the test report and termed as a test oracle. In some conditions, an oracle could be a requirement or the use case for defining the system. Various test cases are required to determine the overall functionality of the system by incorporating performance, load and stress test cases. Test cases are often referred as test scripts, particularly when written are stored in a test repository to share among the test server. A group of test cases is named as Test suites, where all the possible test cases are stored in the execution.

Each test case should contain minimum of two test cases for each requirement where it should consist of positive and negative test cases. In case of any sub-requirements, each sub-requirement must also have at least two test cases. Requirements can be mapped to improve the efficiency of testing by mapping with the Requirement document using Requirement Traceability Matrix. Test cases should include a description about the performance and functionality of the application to be tested, and the master test plan is required to ensure that the test could be conducted formally. A formal test-case must contain a valid input and the expected output. Based on the types of testing going to be done, test case format will differ, where the precondition is the input and the expected output is the post condition to compare the actual and expected results.

3 REGRESSION TESTING

Regression testing is the process of re-executing the old test cases in the new build to ensure that enhancements or defect fixes made to the software works properly and does not affect the existing functionality. It is done every time where any changes are made in the application or any new modules are integrated in the application. It is very essential in the system testing to test the end functionalities by selecting the appropriate minimum set of test efforts required to cover a particular change. Regression testing types are classified as,

i) Unit Regression-Retest
ii) Linear Regression
iii) Full Regression

4 TEST CASE PRIORITIZATION

Test Case Prioritization is the process of scheduling the test cases which are to be executed in a specified order by assigning the values. Test cases with the highest priority are executed first in the test sequence. Test cases are arranged in a logical manner based on the required conditions and run them according to their logical order. It offers test cases having highest priority to be run earlier than the other test cases in the test suite. Based on the criticality of the function in an application and deadlines the test cases are prioritized. Test cases having a highest priority need to be run earlier to prevent blocking state and test cases having low priority can be run later. Prioritization decreases the time consumption in test planning and executing where it will lead to early detection of fault during the integration.

5 RELATED WORK

Santosh Kumar Swain and Subhendu Kumar Pani [1] et al, demonstrate the various important software models used in Model Based Testing. A model contains a UML Based Testing, Finite State Machines, Markov Chains and Grammars [1]. It describes a typical model based testing process to generate test cases and automatic execution of test cases. The Behavior of a system is difficult to extract from the code, but it is easily obtained from design models. State diagram represents the various states involved in the model.

Gurudiksha and Janpreet Singh [2] stated the approaches used for regression test case prioritization and its importance in Software Testing Life Cycle. Defines the prioritization approaches using Dependency Structure, Model Based Prioritization, Requirement Based Prioritization and Prioritization based on Clustering. Test case generation processes are compared and its efficiencies are calculated with respect to the time and cost. It compares the traditional method which is entirely code based and used for a post implementation phase of software development.

A.Hartman and K.Nagin [3] describes about the MBT tool AGEDIS and its interfaces [3]. The approach includes an environment for model test case generations, execution and model checking. It is used for automation and widely accepted tool by industries where it supports model based testing. AGEDIS tool is suitable for small projects where it does not support large scale projects.

Hassan Reza et al.[4] discussed a software testing method for dynamic websites which utilizes the model behavior of the SUT from State chart models originally devised by Harel (1987,1988). State chart models can be used in modeling and generating of test cases to test the applications. However, the major focus was on the flow of design in the module and testing those interactions. Their approach was a systematic to test the front-end functionality of web applications. Usability of links, forms and graphical arrangements are mostly concerned based on the requirement specification documents and also they describe the method of modeling the web applications with the State chart in their work. Generation of test cases from the State chart diagram depends on five coverage criteria: (1) all-blobs (2) all- transitions (3) all- transitions -pairs (4) all-conditions (5) all-paths.

Samaila Musa [5] present an evolutionary prioritized approach that will select the best test cases from the existing test suite T, which is used to test the original program P by using Dependence Graph as an intermediate to identify the changes in P, at statements level. Identification of the changes using this kind of graph will lead to the precise detection of changes. The changed statements will be used to identify the affected statements and test cases that execute the affected statements are selected for regression testing. The selected test cases will be prioritized by using genetic algorithm in order to have a superior rate of fault detection. This approach will reduce the cost of regression testing by increasing the rate of fault detection and reducing the number of test cases to be used in testing the modified program which changes the model. Methodologies available for regression testing involve retest all, regression test selection, test suite reduction, test case prioritization.

Sanjukta Mohany, Arup Abhinna Acharya [6], et al defined that in code-based test case prioritization, source code of the system is used to prioritize the test cases. Most of the test case prioritization methods are code based. In model-based test case prioritization a system's model is used to prioritize the test cases. System models are used to capture some aspects of the system behavior. Fault can be detected earlier in a model based testing, when compared to code-based test case prioritization method. Model-based test prioritization may be an efficient alternative to the existing prioritization methods. However, model-based test case prioritization may be sensitive to the correct/incorrect information provided by the testers/developers.

G. Rothermel, R. Untch, M. Harrol [7] described Regression testing is the process of testing a modified system using the old test suites. Developers need to make sure that modifications are correct and do not adversely affect the unchanged portion of the system. During regression testing the modified parts of the system are first tested. Then the whole system needs to be retested using the old test suite to have confidence that the modifications did not introduce new faults into the system. Because of the large size of a test suite, system retesting tends to consume a large amount of time and computing resources; where it may last for hours, or even days. So one of the issues developer's facing during retesting of the system is ordering the test cases for execution. Test case prioritization tries to address this issue. Test case prioritization orders tests for execution so that the test cases with the highest priority, based on some criterion, are executed before lower priority test cases. Several test prioritization criteria are compared. For example, tests can be ordered to achieve selected code coverage at the fastest rate.

Siripong and Jirapun [8], made researches on test case prioritization techniques where testing consumes 40-70% of the time and cost of software development process. Many techniques are proposed to reduce the testing time, including test case prioritization techniques. It introduces and organizes a new ‘‘4C’ classification of existing prioritization algorithm such as Customer requirement-based techniques, Coverage-based techniques, Cost effective-based techniques and Chronographic history-based techniques.

Salam AL-EMARI and Izzat Mahmoud ALSMADI [9] evaluated the usage of Spec Explorer from Microsoft Research for model based test suite generations. It clearly defines the application functionalities in the form of activity states where the possible states for the modules can be re-explored to identify the modifications done on the application. The two necessary states which are used to explore the functionalities are the initial state and end state. The initial state is to initiate the flow of the application and based on the GUI and final state is the end of the FSM. The finite state machine plays an important role in software testing to plan the flow of the test. State graphs generated for the selected modules can be compared with the other states to explore the input and output conditions for testing. It is also known as finite automation to explore the available finite number of states.

Shin Yoo, Mark Harman [10], et al performs analysis on criteria for test case prioritization where single criterion prioritization will not dominate other methods. They proposed a potential way to enhance a prioritization criterion to utilize the domain expert judgement by asking the human tester to compare the importance of different test cases. Test cases are clustered based on Analytic Hierarchy Process (AHP) algorithm. Using dendragram approach the test cases are selected at defined level.

6 PROPOSED SYSTEM

The design model from the requirement defines the state variables and rules written in the application and the states are identified by state machine which is written in cord scripts for exploring the models. The states are compared and prioritized based on severity. Test case generator generates and clustering the test cases using Agglomerative Hierarchical Process which is cost effective method using dendragram. The following architecture describes the schematic representation of the proposed approach. The architectural design of the proposed system is shown in Figure 1.

The process starts with exploring the models with an application using spec explorer. States and transitions are absorbed for the model and state graph is generated. The model is then modified and a model program for the model is created by configuration cords where the test suites are stored. The machine is created for the test suite and the model programs to explore the possible states are created.

The machine is validated for the available states and transitions which covers code coverage results. A test code and test suite for the model is visualized in test suite graph. The model program illustrates the flow of events in the state based graph. Test case generator generates the test case using test code and test is executed. Test results and states executed can be displayed in the test view editor. Test suites for modified model can be obtained in the test view by refreshing the tests.

The proposed system has the following modules,

(i) Design Model
(ii) States Identification and Comparison
(iii) Flows Prioritization
(iv) Test Case Generator
(v) Prioritized Test Cases

6.1 Design Model

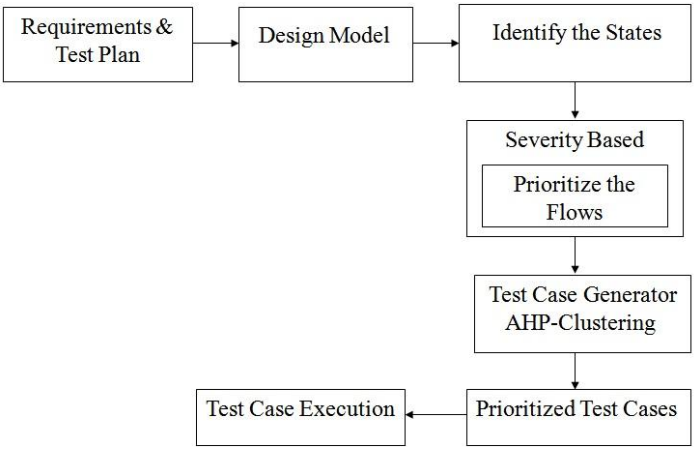

Fig 1. Architectural Design of the Proposed System

The design model describes the behavior of the specified functionality in the state based diagram. Models are used to create the design and possible to find weakness in the design before implementation. Design models are created using spec explorer with UML plug-in. The state based graph is generated based on the requirements and modifications done. Whenever the changes are made the model program has to be changed for exploring the new FSM Model. Dependency graph can be generated to understand the model clearly.

6.2 States Identification and Comparison

The states are explored for a given model using spec explorer. The machine explores the number of transitions and states available for the specified model. Test suites are generated from the defined models. States are compared with previous versions to ensure the modifications had done. It shows the total number of transitions and states covered such as (S0, S1, S2….S10) with start and end states in a graph.

6.3 Flows Prioritization

Flows of a module or event are important to achieve and understand the functionality of the application. In this module flows are prioritized to generate the specified test cases in order to achieve regression test cases .Flows are prioritized to improve testing effort where functionalities are very important in the application under test. Based on the start and end states of the model design flow of an event is achieved.

6.4 Test Case Generation

In this module the test suites are generated and clustered using the hierarchical method. Test cases are prioritized and group together based on severity factor such as Low, Medium and High. The priority value of the test cases depends on the severity factor where high severity test cases are prioritized first and generated according

to the criteria given. Generations of test cases are based on the prioritization criteria and clustering is done to group similar test cases having High, Medium and Low priorities.

6.5 Prioritized Test Cases

The test cases that are generated based on the prioritization criteria for regression testing is stored in test case repositories, with a set of test data. Test cases are grouped based on the testing to be performed such as retest or regression test. Regression test cases are executed on the modified application to ensure the correctness. These test cases are compared with old test cases in the previous build to verify the changes.

TABLE 1
SEVERITY TABLE

SEVERITY	PRIORITY
S1- Low	P1- Low
S2- Medium	P2- Medium
S3- High	P3- High
S4- Severe	P4- Very High

CONCLUSION

Model Based Testing is an effective approach to understand the behavior of any applications. Test case generation and prioritization play an important role in software testing. Test cases are generated earlier in a model based approach when compared to other approaches. It offers high reliability where changes in the models have a direct impact on the application. Existing techniques for prioritization of test cases are a post implementation testing process.

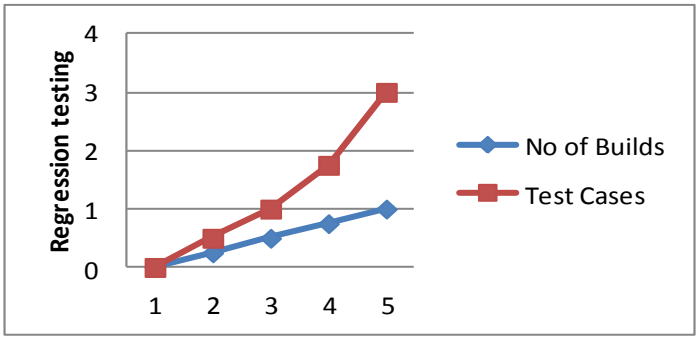

Fig 2.Testing Efforts

The model based approach proposed here is a pre-implementation testing process starts at the design phase. Changes in the requirements have a quicker effect on the models rather than changes in coding phases. Agglomerative Clustering approach is used to group similar test cases based on severity factor provide an efficient method to group similar test cases. Prioritization of test cases for multiple criteria can be improved in the future work.

REFERENCES

[1] Kumar Swain, Subhendu Kumar Pani, Durga Prasad Mohapatra,‘‘Model Based Object-Oriented Software Testing” . *In Proceedings of the Journal of Theoretical and Applied Information Technology*, 2010.

[2] Gurdiksha, Janpreet Singh,‘‘Approaches Used for Prioritization of Test Suites ”. *In proceedings of the International journal of Scientific Engineering and Research*, 2014.

[3] A.Hartman and K.Nagin,‘‘The AGEDIS Tools for Model Based Testing*”, IBM Haifa Research Laboratory,ISSTA’04, ACM,* 2004.

[4] Hazzan Reza, K. Ogaard and A. Malge," A model based testing technique to test web applications using statecharts", *Proceedings of 5th International Conference on Information Technology: New Generations*, 2008.

[5] Samaila Musa,‘‘A Regression Test Case Selection and Prioritization for object-oriented programs using Dependency graph and Genetic Algorithm*”. In Proceedings of the International Journal of Engineering and Science*, 2014.

[6] Sanjukta Mohanty, Arup Abhinna Acharya, Durga Prasad Mohapatra,‘‘A Survey On Model Based Test Case Prioritization”. *In the proceedings of International Journal of Computer Science and Information Technologies*, 2011.

[7] G. Rothermel, R. Untch, M. Harrol, “Prioritizing Test Cases For Regression Testing,” *In the proceedings of IEEE Transactions on Software Engineering*, 2001.

[8] Siripong and Jirapun, ‘‘Test Case Prioritization Techniques”. *In the proceedings of Journal of Theoretical and Applied Information Technology*, 2005-2010.

[9] Salam AL-EMARI, Izzat Mahmoud ALSMADI ,‘‘Using Spec Explorer for Automatic Checking of Constraints in Software controlled Systems”. *In the proceedings of Informatica Economica*, 2011.

[10] S. Yoo, M. Harman, P. Tonella, and A. Susi,‘‘Clustering Test Cases to Achieve Effective and Scalable prioritization Incorporating expert knowledge”. *In the proceedings of International symposium Software Testing and Analysis*, 2009.

[11] R.V.Binder, ‘‘Testing Object-Oriented Systems: Models, Patterns, and Tools”. *In the proceedings of Addison-Wesley*, 1999.

[12] Logan Yu, Robert B. France, Indrakshi Ray, ‘‘Scenario-based Static Analysis of UML Class Models”. *Springer-VerlagBerlin Heideberg*, 2008.

[13] Lin Chen, Ziyuan Wang, ‘‘Test Case Prioritization for Web Service Regression Testing”. *In the proceedings of Fifth IEEE International Symposium on Service Oriented System Engineering*, 2010.

[14] Quart-ul-an-Farooq, M.Z, ‘‘A Model Based Regression Testing Approach for Evolving Software Systems with Flexible Tool Support”. *In the proceedings of IEEE*, 2010.

[15] Ryan Carlson, ‘‘A Clustering Approach to Improving Test Case Prioritization’’.An Industrial Case Study. *In the proceedings of IEEE*, 2011.