

Software Test case Classification based on ‘Defect-Rate’ of Test Suites

Ms. P. Ranjani,

AP/IT Department, Faculty of Engineering,
Avinashilingam Institute for Home Science and Higher Education for Women,
Coimbatore, Tamil Nadu, INDIA

Abstract

Software Testing plays an important role in software engineering to measure the quality system as well as artifacts produced during the software development life cycle. So it consumes 40% of cost from the total development effort. To reduce consumption of cost to test the system the used test cases in a test suite can be organized in a particular manner in the reuse repository. This may help the project plan team to frame the decision for where they can adjust the resources for effective software development. The Classification Tree Method can help to organize the test cases for further consideration. In this we proposed to take ‘Bugs-Rate’ analysis also to group the test cases to check their performance when they are played in different platform.

Keywords – Software Test Case Classification, Bugs-Rate Analysis, Reusability, Data Mining

I. INTRODUCTION

In software development the testing phase is playing a vital role to find and declare the quality of the software products by assessing them through executing a different set of test procedures. It is a filtering phase for finding faults in the product as well as in applied development procedures. Software failures may be in any of the following form such as ‘Errors’ or ‘Faults’ or ‘Bugs’. Another term is also available that is ‘Defects’. It represents that mistakes happened in the development procedures. If a requirement is misunderstood by software developer then it is a ‘Defect’. It is happened in the initial stage of software development then rest of the development activities is not correct. Hence defects removal it is also one of the task required to apply in software constructing activity. But here we have taken other system failures except ‘Defects’ of the system.

A. Data Mining on Test case Classification

Data mining is consisting different procedures to process the available data for having an expected result. The different procedures are classified as ‘Static Data Mining’ and ‘Dynamic Data Mining’ techniques. In static data mining the attributes are already selected for further classification process. But in dynamic technique the attributes may be selected at run time or real time to have the different groups of entities. The advantages of Data Mining Techniques are very huge. It plays vital role in various business fields and different engineering areas for taking decisions and evaluating the development procedures.

Data Mining is used in Software Engineering for different purposes from top to bottom of development activity. In software engineering many data are produced and used while developing a software system. Hence it will record all the information in a formatted manner for future scope. Those data can be processed for taking decisions for current development. In that same manner the ‘Software Test cases’ will be recorded as per their data attributes and their performance also recorded for performing testing activity.

The test cases for new proposed system will be designed or identified from exiting test cases of previous products. Then reusing the test cases is also possible when it is suitable to the proposed system functionality. This may help to design the test case for performing Regression Testing [1] & [6].

$$\text{Test Case } T3 = R1 \cap R2 \quad (1)$$

Where

R1 = Existing System Requirement

R2 = New System Requirement

The new test case is formed when new and old requirements are matched. Hence the new test case for a proposed system will be created by mapping the requirements with existing requirements and their test cases used for testing the system. So time, effort and cost for designing the test cases can be reduced.

II. TEST CASE CLASSIFICATION

The software development is in 3 different engineering concepts. The first one is 'Green Engineering'; it means that the software will be developed from the scratch only. Hence the software development will consume much or maximum effort to build the software systems in terms of satisfying users' expectations in a successful manner. The second one is 'Reengineering'; it represents that the software can be developed from the existing software systems or from their parts. In this type of software development the suitable parts are identified from already developed software systems in order to reduce the development effort up to certain stage. The third one is purely about developing User Interfaces effectively. So it is known as 'Interface Engineering'. In this effective user interface design will be designed for improving the 'Usability' feature of a system or existing system [2] & [3].

Among the above 3 engineering development fields, the 'Reusability' based engineering the system such as 'Reengineering' may focus on finding the perfect components or software packages from the existing software system to design the solution for the proposed one. By applying the identified elements to perform the proposed requirements may consume less development effort to construct the software system in an effective manner. Suppose the packages and components are racked in a perfect manner by using their different characteristics then it is also reducing the time of searching suitable elements to be used for developing software system.

A. Test suite for Testing

The software testing is focused on filtering the software bugs or faults by testing the software system with different testing procedures. In software testing area different types of testing procedures are available to find expected and unexpected bugs or faults. If a test case or test suite finds unexpected bugs then it should be updated in the test case dataset. Then only in coming testing moments that testing procedure may come with updated test cases to find the expected faults present in the system. In this

manner updated testing tools or test cases may replace the old ones. Therefore placing such test suites is very important one to reduce the total effort and cost required for testing the software system in perfect condition status.

III. CLASSIFICATION OF TEST CASES

The software testing is taking the different testing tools, procedures and along with test cases to test the software system in a required status. Here taking testing procedures is a complex one because system to system the same testing type will be conducted in a different form in order to check the deviation of systems from their expected system behavior. In this paper we discuss about different classification techniques for grouping the test cases for identifying them in a simple manner. The different classification procedures are

- Test case priority
- Role of test case
- Resource Consumption based
- Defects Based Testing

Classification Tree Method (CTM) is a technique used to classify the test cases as per their selected attributes [4] & [5]. The test case attributes should be framed in advance to find the similarity among the other tests cases of testing procedures to make their family.

A. Test case priority

In this method the test case level in the proposed testing procedure will be identified as per its function in finding results. The initial level test cases may be in highest priority, but which test case will inspect the unexpected bugs will get highest priority. It represents that the test case can identify expected and unexpected bugs induced in the developed system.

B. Role of Test case

The designed test case is in the aspect of finding particular expected faults only. But while running test cases against the system execution it may find or generate new faults or bugs present in the system. This represents that the function of the test case is for playing an important role in order to find the bugs and technical faults in the proposed system. The test cases can be organized as per their working roles in the testing the software to fulfill the customer expectations.

C. Resource Consumption Based

The test case is designed for finding faults in developed source code of the system by executing the test case against the system execution. For executing the system in order to find the bugs the certain execution conditions that means execution atmosphere should be designed to execute the system. Hence the time and cost may be acquired to create such a platform for running the test cases or testsuite. From this above discussion about the cost for creating the platform is important one to put in to account to fix the product price. Therefore arranging same role of test cases in the aspect of their resource consumption may give a clear idea about the investment required for testing software system.

D. Bugs-Rate Based Testing

In this classifier method, after applying each test case the amount of expected bugs and faults will be the result. There may be an expected bugs rate while design the test case for specific functionality of software system. When the expected bugs rate is increased then the designed test case act as smart to enable the hidden bugs exist in the functionality of the system. So the same test case may give different bugs rate when it works under different atmosphere. This feature of the test case may give very good support when choosing the same test case for testing the new system in a new platform. Here we propose this classifier method to classify the test cases as per the found Bugs-Rate while testing the software system against software functionality.

Algorithm: Test Case Bugs-Rate (TCBR)

Input : Test cases and their Bugs-Rate

Output: Different Groups of Test cases

Procedure

Read Test Cases and their Role of Test cases

Read Test Cases' platform requirements

Read Bugs-rate of each Test case

Take Threshold value of Bugs-Rate

Make them in each group

Output : Low Bugs-Rate Test case, High Bugs-Rate Test Case, Medium Bugs-Rate Test case

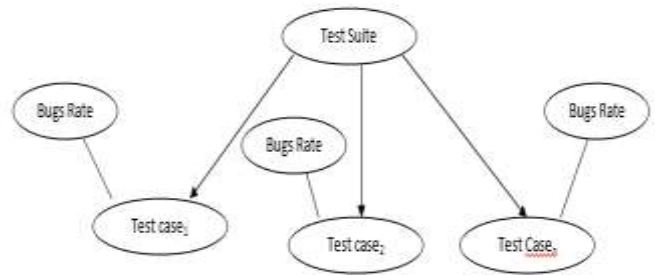


Fig. 1. Test case Classification based on Bugs-Rate

The above figure 1 shows that the test cases of a particular test suite can be organized as per their 'Bugs-rate' also. It may help to find the constraints of test cases to be applied on same functionality but for different applications. Through this a clear idea will be derived for testing the system in a perfect manner with low cost.

IV. MERITS AND DEMERITS

The proposed system is fully based on the 'Bugs-Rate' of the test cases when they are executed under different execution environments. Thus represents when test cases are played in different platform then their 'Bugs-Rate' also will get difference. Then only the Test cases can be organized in that manner. Until then the test cases may be in single group or class only.

By classifying the test cases as per their test case performance then it may be perfect one when it is identified for testing new products.

V. CONCLUSION

In this paper we discussed about the different ways are available for classifying test cases and we proposed 'Bugs-Rate' analysis method to classify the test cases for further level of test case usage. It may be perfect for applying test cases in suitable environment and it will be used to check whether the same test case bring different unexpected result while testing the similar functionality on different products and different environments.

REFERENCES

[1] Sujata , G.N.Purohit, "Tool Support For Test Case Selection In Regression Testing", IJSEA, Vol.2, No.4, October 2011,pp 26-28.
 [2] S.S.Sreeja Mole et. al "Unsupervised Hybrid Classification for Texture Analysis Using Fixed and Optimal Window Size". IJCSE International Journal on Computer Science and Engineering Vol. 02, No. 09, 2010 ,pp 2910-2914.

- [3] B. Ramadoss et al “Combined Classification Tree Method for Test Suite Reduction”, IJCA, 2011, pp 27-30
- [4] Matthias Grochtmann, Klaus Grimm, Joachim Wegener, “Tool-Supported Test Case Design for Black-Box Testing by Means of the Classification-Tree Editor”, EuroSTAR’ Vol. 93,1993, pp 25-28.
- [5] Wong, et al., “A study of effective regression testing in practice”, In Proceedings of 8th International Symposium on Software Reliability Engineering, 1998, pp 264-274.
- [6] Liela Naslavsky, Hadar Ziv, Debra J. Richardson, “A Model based Regression Test Selection Technique”, Proceeding of IEEE International Conference on Software Maintenance(ICSM).2009, pp 515-518.