Minimization of Test Suite through Regress Analysis of Requirement Coverage

Dr. Rajat Sheel Jain, Dr. Amit Gupta

Abstract – Requirements are very important features for the designing of any software application. Test suite creation is a big and hectic task for the Software Quality Analyst. We propose to develop a technique for analysis of Covered Requirement and its impact on the designed Test suite. The requirement associated with the test cases accepts specification like execution time, costing for minimization of test suite. The specification analyser compares the information about the techniques like Precision, Efficiency, Inclusiveness and Generality. By reducing the test suite size, we can reduce the execution cost and time, validation and management of the test cases from the suite for future releases of the software and able to maintain the fault detection capability by reusing the refined test cases. The requirement coverage will increase time-effectiveness in sorting the features of the application and reduces the duplicacy. An improved rate of testing activity will provide faster feedback of the system under test.

Keywords: Requirement Coverage, Retesting & Regression Testing, Data Flow Technique and Suite Refinement*.*

Dr. Rajat Sheel Jain
Department of Information Technology, Institute of Management Studies, Noida, India
jainrajatsheel@gmail.com

Dr. Amit Gupta
Maharaja Agrasen Institute of Management Studies, New Delhi, India
amitgupta21@gmail.com

I. Introduction

Software Quality Testing is a part of verification and validation. The software quality assurance is essential for organisations. The main objective is to reduce the cost of guarantying quality throughout the software development process.

Software Testing is the activity that individual does with the intention to find out the errors in software applications. Regression Testing is the process of validating modified software to detect whether the new errors have been introduced into the previously tested codes and provide confidence that the modifications are correct.

Since the regression testing is an expensive process, researches have proposed regression test selection techniques as a way to reduce some of this experience. These techniques attempt to reduce the costs by selecting and running subsets of the test cases in the program's existing test suites .however it is difficult to compare and evaluate these techniques because they can be used to solve the different problem goals.

All code-based regression test selection techniques attempt to select a subset T' of T that will be helpful in establishing confidence that P' was modified correctly and that P's functionality has been preserved where required. In this sense, all code-based test selection techniques are concerned, among other things, with locating tests in T that expose faults in P'. Thus, it is appropriate to evaluate the relative abilities of the techniques to choose tests from T that detect faults.

II. VARIOUS SELECTION TECHNIQUES FOR REGRESSION TESTING

There are so many regression test selection techniques: Path Analysis Technique, Data flow Techniques, Requirement Coverage.

A. Requirement Coverage

The Software Application designed on the requirement gathering. The requirements gathering technique operate as minimization techniques, they return small test suites and thus reduce the time required to run the selected tests. The test suite can be reduced until and unless it can get associated with the requirements. However, due to the calculations required to solve systems, it can be found that the same requirements can be associated with more than one test case. As soon as the requirement can be associated with the test cases, the status of the requirement will get updated from “NOT COVERED” to “NO RUN”. Despite this possible worst-case behaviour is that the same requirement will get associated with number of existing test cases in the test suite that can obtain solutions, in practice, in times that may be acceptable.

A selective requirement coverage retest technique that uses systems under test to select test suites that yield segment coverage of modified code. Requirement coverage techniques use systems to express relationships between tests and program segments. The technique obtain requirements from matrices that track program segments reached by test cases, segments reachable from other segments, and (optionally) definition-use information about the segments.

B. Data Flow Technique

Several selective retest techniques are based on dataflow analysis and testing techniques. Dataflow test selection techniques identify definition-use pairs that are new in, or modified for, P’, and select tests that exercise these pairs. Some techniques also identify and select tests for definition use pairs that have been deleted from P. Two overall approaches have been suggested. Incremental techniques process a single change, select tests for that change, incrementally update dataflow information and test trace information, and then repeat the process for the next change. Non-incremental techniques process a multiply-changed program considering all modifications simultaneously.

III. SYSTEM ARCHITECTURE

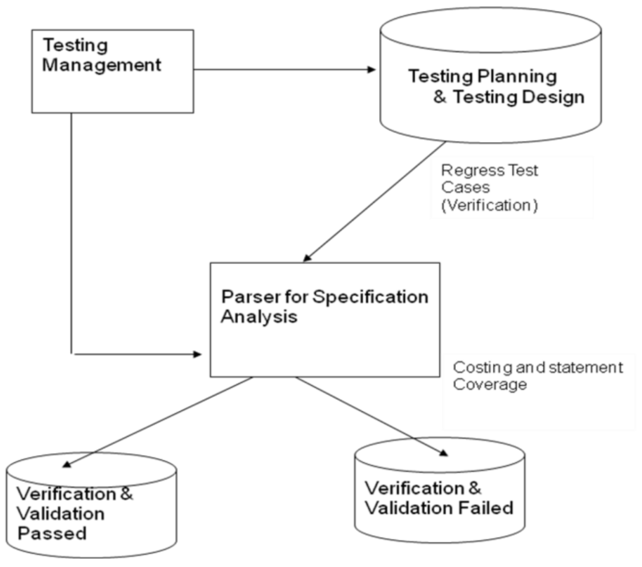

Testing Management
Testing Planning & Testing Design
Regress Test Cases (Verification)
Parser for Specification Analysis
Costing and statement Coverage
Verification & Validation Passed
Verification & Validation Failed

Fig. 1 System Architecture

The above framework will be used for the finding of the fault detection capability of the test suite. There will be the source program through which we can take the input and give it to the test suite.
The test suite that contains the numbers of the test cases specifies that which test cases will cover more number of the program criteria.
The sizes of the test suite minimizes without reducing their fault detection capability of the test cases in the suite.

The framework analyses the test cases on the basis of the four properties i.e. Inclusiveness, Precision, Generality and Efficiency. Inclusiveness measures the extent to which a technique chooses tests that will cause the modified program to produce different output than the original program, and thereby expose faults caused by modifications. Precision measures the ability of a technique to avoid choosing tests that will not cause the modified program to produce different output than the original program. Efficiency measures the computational cost, and thus, practicality, of a technique. Generality measures the ability of a technique to handle realistic and diverse language constructs, arbitrarily complex code modifications, and realistic testing applications. These categories form a framework for evaluation of the test cases that should be analysed through our specification analyser and compare them.

IV. PROPOSED REQIREMENT COVERAGE PARSER

In this paper we proposed to implement the Coverage algorithm which selects the test cases from T whose outputs may be affected by the modifications made to the programs.The algorithm exploits the following observations:

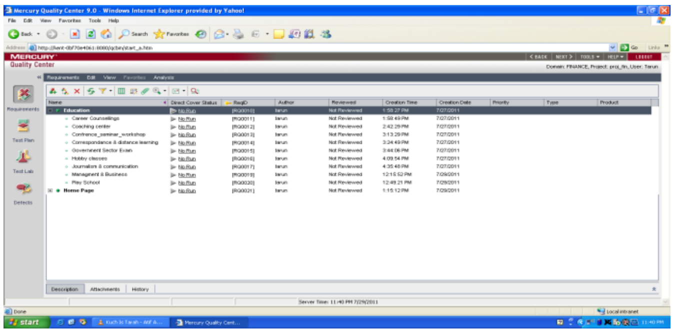
1. Not all statements in the program are executed under all test cases.
2. If a statement is not executed under a test case, it cannot affect the program output for that test case.
3. If a statement is executed under a test case, it does not necessarily affect the program output for that test case
4. Every statement does not affect every part of the program output.

The requirement coverage analyser will work as a parser and to track test cases from the test suite pass to the analyser where these cases will be observed in terms of the statement coverage and costs-benefits.

Using this algorithm to parse which decomposes the program and selects test cases to ensure that there is no linkage between the modified and unmodified code.

V. RESULTS AND DISCUSSIONS

By using above discussed strategies and models we found some good results for requirement coverage and their association with the designed test suite. Some results of our work are displayed here:



Mercury Quality Center 9.0 - Windows Internet Explorer provided by Yahoo!

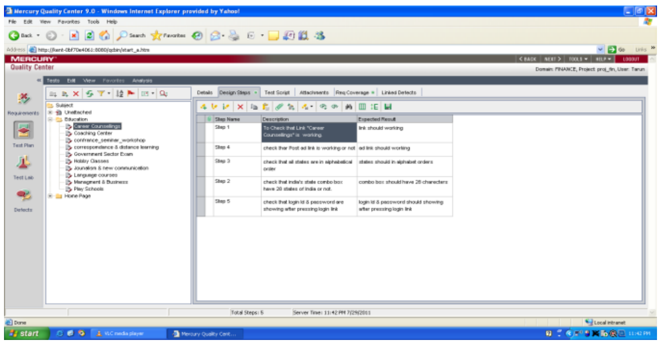
http://lkent-0bf70e4061:8080/qcbin/start_a.htm

Domain: FINANCE, Project: proj_fin, User: Tarun

Name	Direct Cover Status	ReqID	Author	Reviewed	Creation Time	Creation Date	Priority	Type	Product
Education	No Run	[RQ0010]	tarun	Not Reviewed	1:58:27 PM	7/27/2011			
Career Counsellings	No Run	[RQ0011]	tarun	Not Reviewed	1:58:49 PM	7/27/2011			
Coaching center	No Run	[RQ0012]	tarun	Not Reviewed	2:42:29 PM	7/27/2011			
Confrence_seminar_workshop	No Run	[RQ0013]	tarun	Not Reviewed	3:13:29 PM	7/27/2011			
Correspondance & distance learning	No Run	[RQ0014]	tarun	Not Reviewed	3:24:49 PM	7/27/2011			
Government Sector Exam	No Run	[RQ0015]	tarun	Not Reviewed	3:44:06 PM	7/27/2011			
Hobby classes	No Run	[RQ0016]	tarun	Not Reviewed	4:09:54 PM	7/27/2011			
Journalism & communication	No Run	[RQ0017]	tarun	Not Reviewed	4:35:48 PM	7/27/2011			
Management & Business	No Run	[RQ0019]	tarun	Not Reviewed	12:15:52 PM	7/29/2011			
Play School	No Run	[RQ0020]	tarun	Not Reviewed	12:49:21 PM	7/29/2011			
Home Page	No Run	[RQ0021]	tarun	Not Reviewed	1:15:12 PM	7/29/2011			

Server Time: 11:40 PM 7/29/2011

Fig 2: Designing of Requirements



Mercury Quality Center 9.0 - Windows Internet Explorer provided by Yahoo!

http://lkent-0bf70e4061:8080/qcbin/start_a.htm

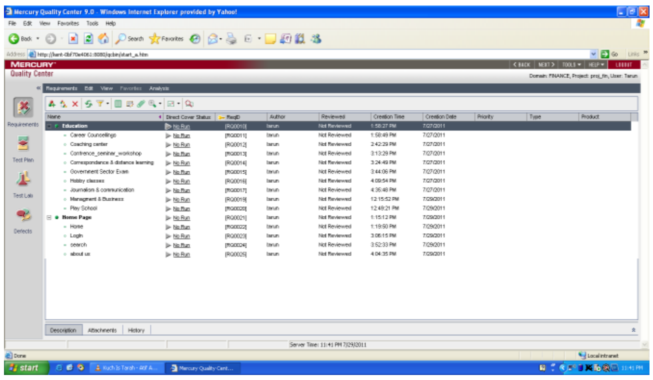
Subject
Unattached
Education
Career Counsellings
Coaching Center
confrence_seminar_workshop
correspondence & distance learning
Government Sector Exam
Hobby Classes
Journalism & new communication
Language courses
Management & Business
Play Schools
Home Page

Step Name	Description	Expected Result
Step 1	To Check that Link "Career Counsellings" is working.	link should working
Step 4	check thar Post ad link is working or not	ad link should working
Step 3	check that all states are in alphabetical order	states should in alphabet orders
Step 2	check that india's state combo box have 28 states of india or not.	combo box should have 28 characters
Step 5	check that login id & password are showing after pressing login link	login id & password should showing after pressing login link

Total Steps: 5

Server Time: 11:42 PM 7/29/2011

Fig 3: Designing of Test Cases



Mercury Quality Center 9.0 - Windows Internet Explorer provided by Yahoo!

http://lkent-0bf70e4061:8080/qcbin/start_a.htm

Domain: FINANCE, Project: proj_fin, User: Tarun

Name	Direct Cover Status	ReqID	Author	Reviewed	Creation Time	Creation Date	Priority	Type	Product
Education	No Run	[RQ0010]	tarun	Not Reviewed	1:58:27 PM	7/27/2011			
Career Counsellings	No Run	[RQ0011]	tarun	Not Reviewed	1:58:49 PM	7/27/2011			
Coaching center	No Run	[RQ0012]	tarun	Not Reviewed	2:42:29 PM	7/27/2011			
Confrence_seminar_workshop	No Run	[RQ0013]	tarun	Not Reviewed	3:13:29 PM	7/27/2011			
Correspondance & distance learning	No Run	[RQ0014]	tarun	Not Reviewed	3:24:49 PM	7/27/2011			
Government Sector Exam	No Run	[RQ0015]	tarun	Not Reviewed	3:44:06 PM	7/27/2011			
Hobby classes	No Run	[RQ0016]	tarun	Not Reviewed	4:09:54 PM	7/27/2011			
Journalism & communication	No Run	[RQ0017]	tarun	Not Reviewed	4:35:48 PM	7/27/2011			
Management & Business	No Run	[RQ0019]	tarun	Not Reviewed	12:15:52 PM	7/29/2011			
Play School	No Run	[RQ0020]	tarun	Not Reviewed	12:49:21 PM	7/29/2011			
Home Page	No Run	[RQ0021]	tarun	Not Reviewed	1:15:12 PM	7/29/2011			
Home	No Run	[RQ0022]	tarun	Not Reviewed	1:19:50 PM	7/29/2011			
Login	No Run	[RQ0023]	tarun	Not Reviewed	3:06:15 PM	7/29/2011			
search	No Run	[RQ0024]	tarun	Not Reviewed	3:52:33 PM	7/29/2011			
about us	No Run	[RQ0025]	tarun	Not Reviewed	4:04:35 PM	7/29/2011			

Server Time: 11:41 PM 7/29/2011

Fig 4: Requirement Coverage with Test Cases

The requirement coverage may refine those test cases that are associated with the designed requirements and perform maximum statement coverage with the minimum cost. The cost could be found on the basis of the fact that, suppose that there are hundreds of test cases in a test pool and out of which only 30% test cases are found that satisfy the criteria for the statement coverage and their fault detection capability or efficiency cannot be effected due to the any modifications made to the program.

The main benefit of requirement coverage is to evaluate that every test case from the test suite will get associated with the designed requirement and compare them with the manual test cases that can be directly given to the test pool. Now we will be able to evaluate that which test cases are more efficient, either manual test cases or those that would be analysed by the requirement coverage analyser satisfying different properties.
This will help in reducing test suite sizes by refining them but the minimization cannot compromise the fault detection parameters in effectiveness of the coverage requirements.

VI. CONCLUSION AND FUTURE WORK

This paper find out that the framework is used to identify the strengths and the weaknesses of those test cases that cannot help for the minimization of the test pool and may affect their fault detection capability. With this approach, we have to analytically evaluate the requirement coverage associated with test suite and efficiency in terms of cost and time for that test suite.
Our evaluation indicates that requirement that will be associated with the test cases will be much more effective for finding the faults which can be compared and understood if our framework is used for the industrial purposes. Once the framework is to be applied it is used to demonstrate that for a given test pool how many test cases are efficient for the finding of the statement coverage and how much cost is to be obtained for these test cases.

ACKNOWLEDGEMENT

The author express his sincere gratefulness to Mr. Rajeev Kumar Gupta, The President, Institute of Management Studies, Noida, India for his encouragement and support throughout the work of this wish and also for facilitating technical and literature facilities, required in the development of this work. One of the authors (Rajat Sheel Jain) expresses his thanks to Mr. Alok Agrawal, Advisor, Institute of Management Studies, Noida, India for his valuable suggestions and encouragement.

REFERENCES

[1] G. Rothermel and M. Harold. Analyzing regression test selection techniques. *IEEE Trans. On Softw. Eng.*, 22(8):537-561, Aug. 2006.

[2] 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 *Proc. of the 21st Annual Int'l. Comp. Softw. & Appl. Conf.*, pages 522-528, Aug. 1997.
[3]. Agrawal, H., Horgan, J.R., Krauser, E.W., and London, S.A. Incremental regression testing. In *Proceedings of the IEEE Software Maintenance Conference* (1993), pp. 348–357.

[4]. Rothermel, G. and Harrold, M.J. *A Comparison of Regression Test Selection Techniques.* Tech. Rep., Department of Computer Science, Clemson University, Clemson, SC, Oct. 1994.

[5] J.-M. Kim, A. Porter, and G. Rothermel. An empirical study of regression test application frequency. In *Proc. of the 22nd Int'l. Conf. on Softw. Eng.*, June 2000.

[6] D. Rosenblum and G. Rothermel. A comparative study of regression test selection techniques. In *Proc. of the 2nd Int'l. Workshop on Empir. Studies of Softw. Maint.*, Oct. 1997.

[7]. Rothermel, G. and Harrold, M.J. A safe, efficient algorithm for regression test selection. In *Proceedings of the IEEE Software Maintenance Conference* (1993), pp. 358–367.

[8] W.E. Wong, J.R.Horgan, S.London, and A.P.Mathur ,"Effect of the Test Set Minimization on Fault Detection Effectiveness,"Proc.17th Int'l Conf.Software Eng., pp.41-50, Apr.1995.