International Journal of Advanced Science and Research
ISSN: 2455-4227
Impact Factor: RJIF 5.12
www.allsciencejournal.com
Volume 3; Issue 5; September 2018; Page No. 12-15

The approaches of regression testing based on cost effective model

Manoj Kumar Loganathan[1], K Saravanan[2]
[1] M. Phil. Research Scholar, Department of Computer Science, PRIST University, Thanjavur, Tamil Nadu, India
[2] Research Guide, Department of Computer Science, PRIST University, Thanjavur, Tamil Nadu, India

Abstract
In the sphere area of software engineering, several applications have been developed. An application requires changes because of changes in the customer requirements. Testing is a process of executing the program with the intent of finding the software bugs. Regression Testing is the systematic series of action for carrying out the set of test cases that passed on the previous release of the application under test in order to substantiate that the original characteristics and functions persist working as they were previously. Regression testing consumes maximum for maintenance in the terms of time and cost. The existing approaches for reduction of test cases are not sufficient to handle the problems. For that reason, an approach of deriving optimized test cases that provide the maximum test coverage has been proposed. Use cases are considered for identifying the flows and to reduce the number of test cases to provide satisfactory results.

Keywords: software maintenance, regression testing, test suite, test case prioritization

1. Introduction
Testing is the predominant activity of software engineering. It examines whether the software is running as expected or not and ensures the quality of the software. Testing is also an important element of software quality assurance (SQA). Many software organizations are spending up to 40% of their resources on the testing process [1]. There are three methodologies are in the software testing process. They are White box testing, Black box testing, and Gray Box testing. Software Quality Assurance team mostly they have used Black box testing process. Testing activities that are used by the testing team are requirement analysis, Test planning, test case generation, environment setup, smoke testing, system testing, regression testing, and test closure. Regression testing is a process of executing the test cases to confirm that the updated module in the application does not affect the existing behavior of the application [2]. This phase is the most expensive among the software testing life cycle. Hence, the testing team spends more costs in regression activity. In regression testing re-executing all the test cases is an expensive process. The selection and prioritization of significant test cases are based on some measures such that enhance the quality of testing. In the prioritization of test case process, all the test cases are well-organized. By this way, the most significant test cases are executed [3]. This work enhances a new technique for regression testing. The main aim of this technique is to reduce the time, increase the coverage and reduces cost expenses. The model-based method has been used to construct the working process of this technique [4]. Use cases are used for test case creations and effort calculations.
Along with the Model-based method, test case prioritization and history based method have been used for qualify the technique in terms of time and cost. This model-based method has been used to create test cases and execute the test cases manually to find the early discovery of bugs [5]. Early detection of bugs will make the developer easy to fix the bugs and also make comfortable in the time of the code deployment to production. Hence these models improve the regression testing and qualify the application without any bugs. Time and cost-effectiveness are important for the software testing process. This work completely makes the regression testing process in an effective way in the sense of cost and time. For the testing phase, the cost and resources will be playing the vital for the process [6]. Hence the minimization and prioritization is the method that provides the good outcome for the testing phase.

2. Cost efficacious model-based regression testing
The combination of the existing test cases and enhanced test cases are the regression test suits shown in figure 1. But if all the test cases are considered, then time and resource would be high [7]. Hence, new techniques are handled to improve the quality of regression testing and decrease the time and resource to minimize the cost and increase the profit to the organization.

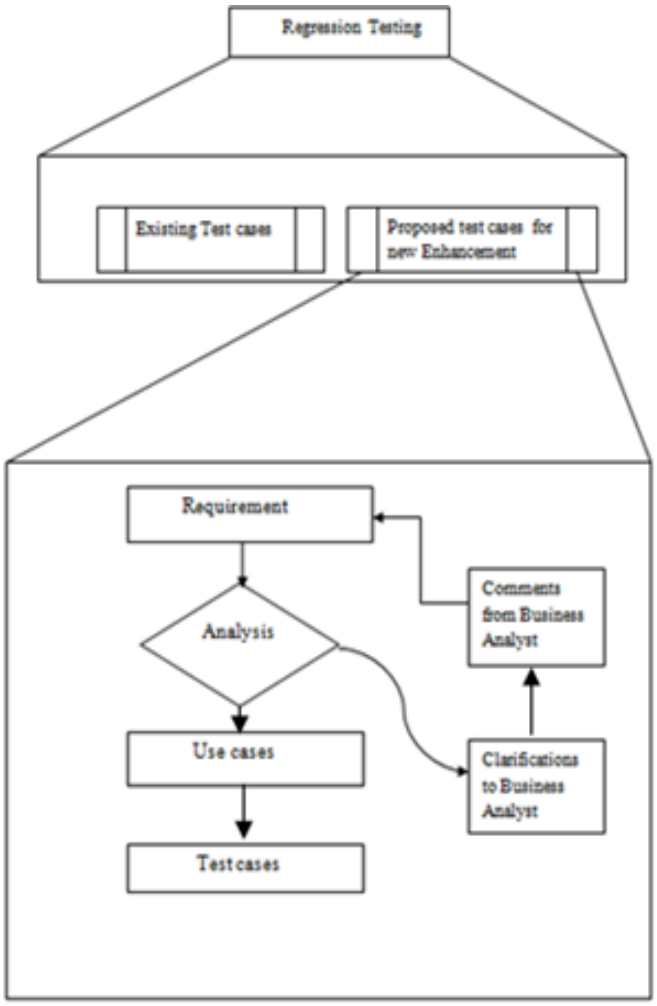

Fig 1: Cost Efficacious Model-Based Regression Testing

2.1 Test case generation

Use case diagram, class diagram, and state machines are commonly used for generating the test case. Generated test cases can be used in all testing activities like unit testing, integration testing, and regression testing [8]. Use case models are used to create the test case. Since this model is much closer to the system flows. The positive and negative flow should be added while creating the test case.

2.2 Common prioritization

Prioritization is the combination of two techniques. Prioritization through severity and history based. In severity based prioritization, the critical test cases are marked as high and rest of the test cases are marked as medium and low [9]. In history-based prioritization, test cases are related to the defects that often occurred in a particular place and that defects are marked as high.

2.3 Effort calculation

Unadjusted Use Case Weight calculation is based on the use case complexity. Factor value is 5, 10 and 15 then use case is classified as simple, average and complex respectively. Unadjusted Actor Weight is based on the actor complexity. It is simple if the external system interacts with the API (Factor=1), average if the external system interacts with TCP (Factor=2) and complex if the human actor is used (Factor=3). [10] Use case Point is computed by the summation of the adjusted use case point and environment complexity factor.

The Formula is used to calculate the efforts spent

Unadjusted Use Case Weight = (Number of Simple use case *5) + (Number of average use case * 10) + (Total Number of complex Use case * 15) (1)

Unadjusted Actor Weight = (Number of Simple actors) *1 + (Number of Average Use case * 2) + (Total Number of complex Use case * 3) (2)

Unadjusted Use Case Point (UUCP) = (Unadjusted Use Case Weight + Unadjusted Actor Weight) (3)

Adjusted Use Case Point = Unadjusted Use Case Point * [0.65+ (0.01*TEF)] (4)

Use case Points = Adjusted Use Case Point*0.5 (5)

2.4 Steps to be followed for reducing cost and resource

The Steps involved in reducing the cost and resource are as follows,

1. Generation of the test cases (Existing test cases + Enhanced test cases)
2. Effort calculation has to be done
3. Process the test cases using common prioritization or severity based prioritization
4. History-based prioritization has to be completed
5. Test cases related to dependency modules should be added with the regression test suit
6. Finally, the effort should be calculated and ultimate effort value will provide the feasible solution

3. Results and discussion

The aim is to reduce the cost and resources in the process of regression testing of any application. To elaborate final effort value, Library Management System has been taken for the case study.

3.1 Use case model

The test case is created with the help of the use case diagram. Figure 2 shows the use-case diagram for library management system.

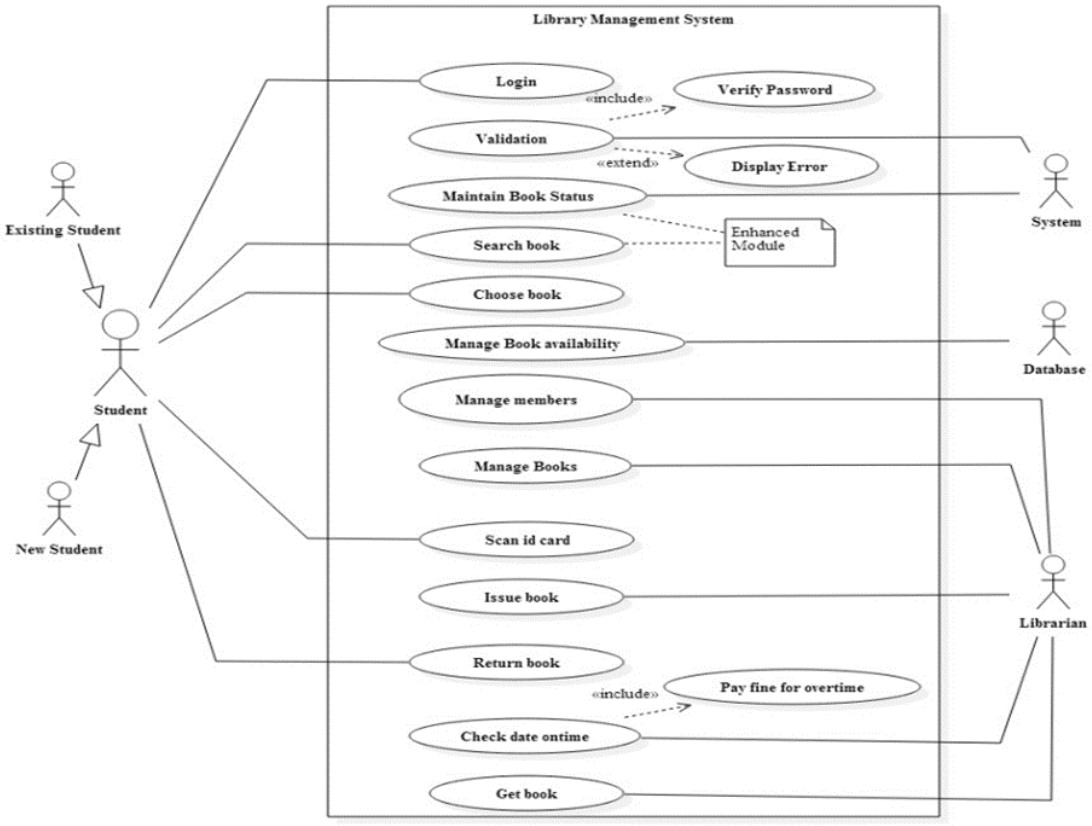

Fig 2: Sample use case diagram of library management system

3.2 Flow of library management system

The flow of positive and Negative test cases of Library Management system is tabulated in Table 1.

Table 1: Different flows of Use Case model of Library Management system

S. No.	Main flow	Positive Flows	Negative Flows	No. of Test flows
1	Login	1	2	3
2	Search Book	7	1	8
3	Manage Members	3	1	4
4	Manage Books	3	1	4
5	Scan ID card	1	1	2
6	Issue book	3	1	4
7	Return Book	2	1	3
8	Check the date on time	1	1	2
9	Choose Book	7	1	8
Total number of Positive flows				28
Total number of Negative Flows				10

In this system Login, Search Book, Manage Members, Manage Books, Scan ID card, Issue Book, Return Book, Check the date on time are considered as common flows. There are 28 positive flows and 10 negative flows. Totally 38 test cases are there to test this complete system.

3.3 Common prioritization

Prioritization can be done by the method of Severity based and History based techniques

Table 2: Prioritization of test cases involved in library management system

S. No.	Common Flow	Flows	Priority	History
1	Login	Login with Correct Credentials	High	1
		Login with Incorrect username	High	0
		Login with Incorrect password	High	0

The Factor has been calculated based on the number of transaction or test cases in unadjusted use case weight. Priorities are tabulated in table 2.

3.4 Applying Cost efficacious model-based regression testing technique

Consider, in Library management system if the options issue status is added in search book page. Then this particular Enhancement would be focused for system testing. The regression testing will be started after the completion of system testing. In this course of time, three things have to be considered for selecting the test cases. According to the proposed concept- module having enhancement, Dependency modules and the cases which have mentioned as high priority should be taken for regression testing.

For example in Library management system following modules are considered for regression. Module having enhancement is Search Book module and Dependent modules are Issue Book module and Login module.

3.5 Enhanced Modules

The two enhanced test cases that are added to the existing test cases are shown in table 3,

Table 3: New Enhanced test cases

S. No.	Main flow	Positive Flows	Negative Flows	No. of Test flows
1	Search Book	Validate the issue detail button is working as expected by giving the input as Book number	Validation of Search field with Incorrect name or ID	2
2	Issue Book	Validate the Book status is displayed as expected in Issue Book module	-	1
Total Positive flows				2
Total Negative Flows				1

The dependency module is the important module. Without this module, the regression testing cannot be done Issue Book module and Login module is identified as the Dependency modules and Manage Members is identified as the history based module.

Table 4: The effort estimation of the Library Management system after regression testing

Unadjusted Actor weights				Unadjusted Use Case Weights			
Actor Name	**Actor type**	**Factor**	**Weight**	**Use case Name**	**Use Case Type**	**Factor**	**Total Factor**
Student	Complex	3	1*3=3	Search Books	Simple	5	1*5=5
System	Simple	1	1*1=1	Issue Books	Average	10	1*10=20
Database	Average	2	1*2=2	Login, Scan ID Card	Complex	15	2*15=30
Librarian	Complex	3	1*3=3	Manage Members-History based Test case	Simple	1	1*5=5
Total Unadjusted Actor Weight			9	Total Unadjusted Use Case Weights			50
Unadjusted Use Case Point = UAW+UUCW							59
Adjusted Use Case Point = UUCP *[0.65+(0.01*TEF)]							41.89
Adjusted Use Case Point =294*[0.65+(0.01*6)]							
Total Effort through Use cases Point=AUCP*0.5							20.945
Total Regression Testing Effort =							20.945

Total numbers of Test flows are 18 before using the History-based and severity based techniques. The count reduced to 5 after the use of history based and severity based techniques. Total regression testing effort of Library Management system before test case enhancement is 101.175 and time consumption report after regression testing is 20.945. The same is shown in Table 4.

4. Conclusion

In order to reduce the time and human resource, the approach of deriving prioritized test cases was urbanized. The effort estimation of the library management system (after regression test) shows the good performance of cost efficacious model-based regression testing. The experimental result reveals that the testing effort after regression is high as compared with testing effort before regression.

5. References

1. Jaspreet Singh Rajal, Shivani Sharma**.** A Review on Various Techniques for Regression Testing and Test Case Prioritization, International Journal of Computer Applications (0975-8887), 2015, 116-16.
2. Jim Heumann. Generating test cases from Use Cases, Rational edge, Copyright Rational Software, 2001.
3. Prabhakar K, Ananda Rao A, Venu Gopala Rao K, Satyanarayana Reddy SS, Gopichand M. Cost-Effective Model-Based Regression Testing, Proceedings of the World Congress on EngineeringI WCE, London, U.K, 2017, 5-7.
4. Bharti Suri, Isha Mangal, Varun Srivastava. Regression Test Suite Reduction using a Hybrid Technique Based on BCO And Genetic Algorithm, University School of Information Technology, Guru Gobind Singh Indraprastha University, Dwarka, Delhi-110075, India.
5. Bahsoon R. The Metrics and techniques for selective regression testing in Computer Systems and its applications; IEEE International Conference, 2005.
6. Neeraj Kumar Saklani. Review of Prioritization Techniques in Regression Testing, International Journal of Computer Science and Mobile Computing. 2017; 6:4.
7. David Cohen, Siddhartha Dalal, Jesse Parelius, Gardner Patton. The Combinatorial Design Approach to Automatic Test Generation, This work will be presented at the International Symposium on Software Reliability Engineering, mite Plains, N.Y, 1996.
8. Aggarwal KK, Kaur A, Singh Y. Code coverage based technique for prioritizing test cases for regression testing, ACM SIGSOFT Software Engineering Notes, 2005, 49.
9. Susanne Rösch, Sebastian Ulewicz, Julien Provost Review of Model-Based Testing Approaches in Production Automation and Adjacent Domains, Journal of Software Engineering and Applications. 2015; 8:499-519.
10. Yanping. Regression Test Suit Reduction Using Extended Dependency Analysis, Pages 62-69 ACM New York, NY, USA ©2007 table of contents ISBN: 978-1-59593-724.