Volume 5, Issue 4, 2015 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Hybrid Technique for the Enhancement in Regression Testing

Neha Dwivedi[1*], Rakesh Chandra Gangwar[2], Dalwinder Singh[1]
[1]Department of Computer Science and Engineering, Lovely Professional University, Phagwara, Punjab, India
[2]Department of Computer Science and Engineering, Beant College of Engg. & Tech., Gurdaspur Punjab, India

Abstract*— Regression Testing is the important and irreplaceable part of the test life cycle. Regression testing is executed whenever there is any modification taking place in the system. In order to validate the functionality of the system and to deliver quality product regression testing plays an enriched role. Regression testing requires a lot of time and resource, hence in order to increase the efficiency of regression testing, test case prioritization technique is widely used. Taken into consideration the large and complex project, we have used the approach of the functional dependencies and the model based technique to prioritize the test cases as they provide the advantage of detection of the fault in the early phases of the development life cycle. In this hybrid technique the model based approach provides an easy way to detect modification in the project, whereas the dependency approach is used to identify the dependent functions. Functional value of each function is calculated on the basis of its importance and the number of function it affects. Finally test cases are prioritized according to the descending order of the total functional value. We show that our hybrid technique provides the higher rate of fault detection at the early stages of the software development. The results have also been proved by the APFD metric.

***Keywords*— regression testing, prioritization, model based testing, dependency, functional values.**

I. INTRODUCTION

Software testing is the process of validating and verifying the program or application. Checking the proper functionality of the system that confirms that the project meets the technical and business requirement as expected and can be implemented with the same characteristics. Testing procedure includes comparison of the expected outcome with actual outcome which helps in detecting the errors which may occur during the development of the project [1].

Any application has to undergo through a process of testing, so that it must not result in severe failures. In order to ensure that, defects should be identified at the starting stage of the software testing because the main aim of testing is to detect the faults as early as possible. Otherwise it can be very expensive in the future or in the later stages of the development. Hence testing ensures the quality of the product and this quality product when delivered to the customers helps in gaining their confidence. [2]

Validation and Verification are dynamic and static processes respectively. Validation is executed during, as well as end of the development to ensure the building of the right product. Whereas verification tests those conditions which are given at the starting of the development phase, that has been satisfied or not. [2]

To test the models, there are two types of testing:

The testing models are white box testing and black box testing. White box testing model is the type of model which relies on analysis of the code and the internal logic of the software. It requires good programming skills and logic ability of programmer to test with this model. The Black box testing is a type of model, where no internal working of the system is needed to be known. Even there is no need of programming skills. Input is given to the system and output is manipulated as per understanding wrong or right. Only the external behaviour is observed, without any knowledge of internal functioning. [2]

II. REGRESSION TESTING

Regression testing is a part of the test life cycle. In regression testing the programs are tested in order to find out that, the changes or the modifications made in the program, does not affect the functions which are not meant to be affected. [3] In this method of testing, those test cases which are tested earlier are supposed to be tested again. Testing of the test cases over and over again is very time consuming and involve a lot of budget. Hence many companies have to pay lot of expenses on the testing of the code. Moreover if the quality of testing is not good it hampers the quality of product. [4]

There are many techniques for regression testing [5] which includes the three main ones. Retest all technique is the one where all the test cases are executed again. It is very time consuming and costly. Second one is the test case prioritization technique, where the test cases are prioritized according to some priority. The test is executed so that rate of fault detection is higher. Finally the third one is the test case reduction technique. This technique reduces the test cases such that it will remain effective and consume fairly less time.

Test case prioritization is one of the efficient techniques of regression testing. It aims at scheduling the execution process of test cases in such a way that improves the efficiency of regression testing. [6] Whenever there is any modification in

the software, the regression testing has to be executed in order to validate the functionality of the software which is a costly affair in terms of both money and time. The burden of regression testing has to be reduced and this reduction can be brought up by testing prioritization of test cases. [1] To prioritize the test cases different researchers came forward and brought up various techniques. Some of the techniques and literature survey is represented in the next section of the related work. [6]

III. RELATED WORK

Amitranjan Gantait [7] has used model transformation approach in order to provide test cases commencing activities. The author used a technique which generates test cases from extended activity models. This technique will first create a halfway model and then its test cases get generated from it. After the generation of test cases the author also invented a technique to prioritize those test cases. This technique is based on the possibility of twigs (branches) at the node which plays a role in decision making in the model of activities. ATM was used as an example and has been put forward for checking the effectiveness of the proposed approach.

Xiaobo Han, Hongwei Zeng and Honghao Gao [8], based on heuristic models implemented a new technique of prioritization. This technique obtains two types of information from the models. The heuristic models help to detect the defects in earlier stages. This earlier detection of defects serves by reducing the error rate and because of this the average percentage of the fault detection gets increases. This technique has proved the better early rate of the detection of faults. This will lead to the quality product.

Quart-ul-an-farooq,Mohammad Zohaib Z. Iqbal, Zafar I Malik and Matthias Riebish [9] studied an approach of state based regression testing (START). START, an eclipse based tool was used for this approach. Dependencies exists is various states gets treated by START. START makes it possible to care for the dependencies, during the modifications are taking place. UML 2.1 class diagrams and the state machines are taken into account by START. The efficiency of this technique was proved by a case study on the "student enrolment system". This shows the test suites reduction. START works fine even when it is integrated with other testing tools. Parsor, comparator and test suite analyser are the main parts of START. XMI v2.1 format is used as an input in START.

Swarnendu Biswas and Rajib Mall [10], have described test suite management. The authors suggested that in order to provide more efficient functionality, during the modifications being made in the original test suites, there is a requirement of test suits to be managed. It can be done by putting some new test cases in, while some of the test cases are needed to get out of the suite. This in and out process require management. The process of again using the test cases which have been used earlier, before the modifications were made is known as test case reuse. The reuse of the test cases reduces the effort and money.

Gregg Rothermel, Roland H. Untch and Mary Jean Harrold [1], have proposed the technique for prioritizing the test cases with the help of execution information. Firstly test case prioritization is done on the basis of the code which is covered. Secondly, prioritization is done on the basis of the code coverage which has not been covered earlier. Thirdly, prioritization is done on the basis of the capability of discovering the defects. Practical implications of these techniques have been done on various test suites. Increased average percentage rate of fault detection (APFD) has been found which reflects the efficiency of these techniques. The main drawback of these techniques is that they are very costly. Hence further research is going on to reduce the cost of the factor also.

Prateeva Mahali and Arup Abhinna Acharya [11], have studied the model based optimization and prioritization for the test suites in regression testing. Their methodology suggests using the genetic algorithm, the optimization of test cases is first performed and then prioritization is performed. This can result in producing the better results and the successful regression testing. The authors also found that there is reduction in the cost and time with these methods. Case study of a shopping mall was taken into account in order to provide results which are better in comparison. They also have represented the system under test by an activity diagram in UML 2.0.

Shifa-e-Zehra Haidry and Tim Miller [12] used the dependencies and have conducted a study on prioritization. Their study suggested that, existence of large coupling makes the prioritization technique more efficient. Test-case prioritization is done based on dependencies, as the dependencies can be observed. This leads to the ordering of test cases. The test cases which have more dependents will be executed first. Resources can also be used efficiently with the help of prioritization. Otherwise, the test cases get no resources even which are important to execute. They also suggested that, dependency can be open or closed. Different algorithms can be used to detect both kinds of dependencies, basically the test case.

J Ryser and M Glinz, [13] has described a kind of dependency diagram and the notation which are the advancement in the UML diagram. These are used to depict the functionalities of the project as in the form of the scenario. Each scenario has some relation with another scenario and they are interlinked. The interaction between the scenarios was shown in the form of the diagram with dependencies in them. The authors also explained that how these dependencies chart can be useful in the testing of the product. This will be beneficial to reduce the cost of the regression testing.

IV. MODEL BASED TESTING

Model based testing is an automated type of black box testing approach, which provides an innovative and advance method for test case generation and test case evaluation. The test cases are built at the early stages of development, based on the model. The model provides an actual overview about the requirement, specification and behaviour of the system. Specific industry standard are there which governs the generation of the model, which can be further enhanced by experience standard [14].

Model based testing is very flexible in nature. Changes in the requirement can be simply implemented by updating the model and then test suits can be quickly generated which is unlike other conventional testing process. In this process editing of test suites is very tedious and error prone. Moreover model based testing reduces both cost and time. This can further increase the effectiveness of the test life cycle.

V. DEPENDENCY APPROACH

A large software product consists of large number of subsystem or components, which further contains programs to a specific task. In software engineering these subsystems or components are called as scenarios. This scenario depicts the behaviour or sequence of action which is performed by the system. These ordering of the scenario show the dependency and the interaction among the scenario. Action of one scenario is dependent on another scenario and that cannot be executed before the completion of the action of the dependent scenario. [12]

On the basis of scenarios, test cases are generated. There are mainly two types of test cases:

1. Independent test cases: these are the type of test case which is executed independently without the concern of the other test cases.

2. Dependent test cases: are those types of test cases whose execution cannot occur freely without concerning the dependent test cases. The execution of test case is only possible when it had finished the execution of the all the dependent test cases.

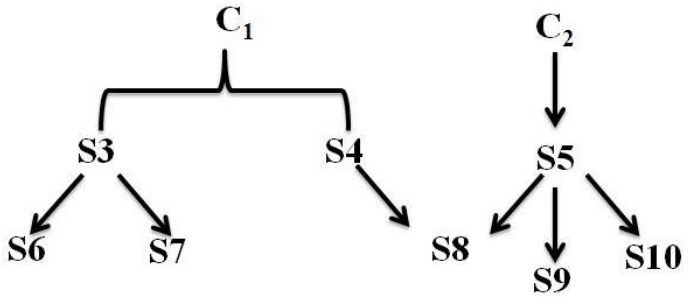

Fig.1 Dependency graph: This diagram shows that C_1 and C_2 are two independent components. S3 to S10 are dependent subcomponent. S3 is dependent on C_1 directly and S6 is indirectly dependent on C_1.

Prioritization of the Test Cases using Dependency: Complexity of the large system is greater due to the large amount of coupling and the interaction between the parts of the system. For the early detection of the fault it is recommended to find out those parts of system, which have higher interaction. [4] Interaction among the system indicates the dependencies of the function in the system. Therefore prioritization of the test cases in the order of the weights of the dependencies will result into the early detection of the faults in the system. [2] In order to find the priority of the functional dependencies, a technique called as dependency structure prioritization (DSP) is used.

VI. METHODOLOGY

In this study, there were mainly two objectives, 1) to propose a hybrid technique for test case prioritization and 2) to implement this hybrid technique for test case prioritization and compare it with the previous technique. Taking these objectives, in this paper, we have used the approach of model based testing to generate the model of the system which is to be tested. Recognition of the function which is modified becomes easy with the help of the model. After the successful identification of the modified function, the dependency approach will help us to find out the affected function of those functions. Hence both of these approaches can work in the hybrid form to prioritize the test cases efficiently and it also detects defects at early stages. After the prioritization of the test cases the result is compared from the old genetic algorithm with the APFD metric.

Proposed technique:

Steps of proposed technique are as follows:

- **a.** Select the project and maintain the database (i.e., information related to the project, such as number of function it has, number of time each function has occurred in the project, number of function affected by each function).
- **b.** Prepare the model of the project.
- **c.** Identify the changes which had occurred due to modification in the project. This is done by comparing the modified project database with the old database.
- **d.** Identify the functions which are affected because of the modification.
- **e.** Evaluate Function Values (FV) of each affected function, according to the formula of Function value.

$$\textit{Function value} = \frac{\textit{Number of time function encounter}}{\textit{Total number of factor being affected}}$$

- **f.** Traverse activity diagram with the help of Depth First Search (DFS) to find out the dependent functions.
- **g.** Calculate the Total Functional value (FTV) by adding functional values of each dependent function.
- **h.** Test cases are prioritized according to the decreasing order of the FTV value.
- **i.** Calculate the rate of fault detection by the APFD metric.

VII. CASE STUDY OF ONLINE SHOPPING WEBSITE

In our work, a case study of the online shopping website has been taken. In this, using the concept of model based testing we created the model of the online shopping website. This model displays the various function of the model.

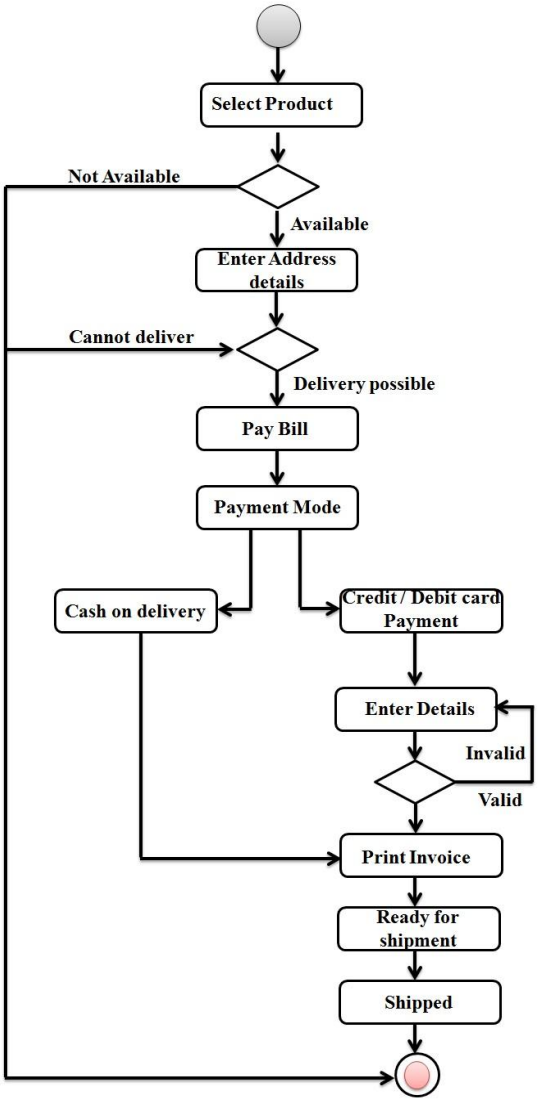

Fig. 2 Online shopping website: displays the different functions used.

There are seven functions which are used in the activity diagram. These functions and their descriptions are given in the Table 1:

Table 1: Function id and their functionality

FUNCTION ID	FUNCTIONALITY
F1	START
F2	SELECT A PRODUCT
F3	CHECK AVAILABLE
F4	BILLING
F5	PAYMENT MODE
F6	ONLINE PAYMENT
F7	CASH ON DELIVERY

VIII. RESULT AND DISCUSSION

In the previous studies it has been reported that, different techniques are dependent upon the number of fault detected in the past history by the test cases. These test cases were prioritized on the basis of the greedy algorithm. The higher the number of fault detected in the past history, the greater will be its priority. This algorithm has certain problems associated with it. Firstly it is dependent on the past and secondly it has not considered the importance of the function. However there can be such test cases which have higher importance value because of the criticality of the function. This was just because it has discovered fewer faults in past, it have provided less priority.

Another algorithm which is used for test case prioritization is the genetic algorithm. Genetic algorithm is based upon the fitness value, the chromosome no. and mutation. This prioritizes the test cases based upon the estimated fitness value. In the proposed technique the test cases were prioritized based on the calculated Total Functional Value (FTV). This leads to much better result of the rate of the fault detection. The FTV value is based upon three important aspects. 1) The importance of the function. 2) The number of the function associated with its function, that provides greater functional coverage and 3) it does not need the information of the fault detected in past history, which means it is not dependent on past history. The model which is generated is itself sufficient to provide all information. Calculated FTV value is more reliable than the estimated value of the genetic algorithm. It provides the higher rate of the fault detection. The higher rate of fault detection has been shown in this study. The calculation and results are proved by APFD metric. The model which is generated will help in the identification of the modified function. After the successful identification of the modified function, the functional value of each function is calculated.

All the implementation of this work was carried out in MATLAB. Table 2 shows the calculated function value, which gives the importance of the function. Functional value and the importance of the function are directly proportional to each other. Hence more is the functional value; more will be the importance of function. The function value is calculated according to the given formula.

$$\textit{Function Value:} \quad \frac{\textit{Number of times Function run}}{\textit{Total number of factor being affected}}$$

Table 2: Calculated function value

FUNCTION	NO. OF TIMES THE FUNCTION ENCOUNTERS	NO. OF FUNCTIONS AFFECTED	FV
F1	4	2	2
F2	2	6	0.3333
F3	2	3	0.6667
F4	1	7	0.14286
F5	3	2	1.5
F6	4	6	0.6667
F7	2	4	0.5

After the calculation of function value, each function of test cases is traverse and function test values (FTV) are calculated. CHANGE_1, CHANGE_2, CHANGE_3, CHANGE_4 are the changes which are required in the project and following functions which are shown in the Table 3 are affected due to these changes.

Table 3: Affected functions and FTV values

NO. OF CHANGES	AFFECTED FUNCTIONALITY	FTV
CHANGE_1	F1,F2, F3,F4	3.1429
CHANGE_2	F3,F5	2.1667
CHANGE_3	F6,F7,F2	1.5
CHANGE_4	F1,F2,F3,F4,F5,F7	5.8095

At the end, test cases are prioritized according to the decreasing order of FTV. The test cases are executed on the descending order of the FTV. The test sequence is: **T4>T1>T3>T2**

Table 4: Prioritized order of test cases

TEST CASES	FTV
T4	5.8095
T1	3.1429
T3	2.1667
T2	1.5

The APFD metrics is used to detect the rate of the fault detection. The higher the value of APFD metric, the better will be the rate of the fault detection. The APFD can be calculated by the following formula:

$$APFD = 1 - \frac{(TF1 + TF2 + \ldots\ldots\ldots TF\,m)}{mn} + \frac{1}{2n}$$

Where, T is the test suite which is under the evaluation.
m = number of the fault which is contained in the program p.
n = number of the test cases to be tested.
TFi = it is the first test which has discovered the fault i

According to the APFD metric the rate of fault detection of new algorithm is 44.3805% and the old algorithm is 10.2481%. This means that, the rate of fault detection is 4 times higher in the new algorithm compared to the old algorithm.

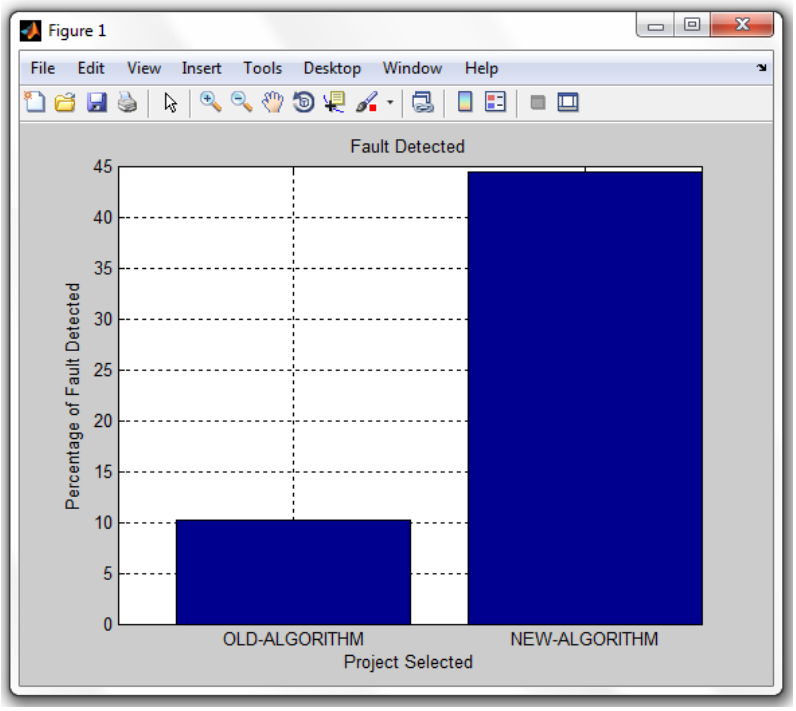

Fig. 3 Bar graph representation of fault detection with old and new algorithm.

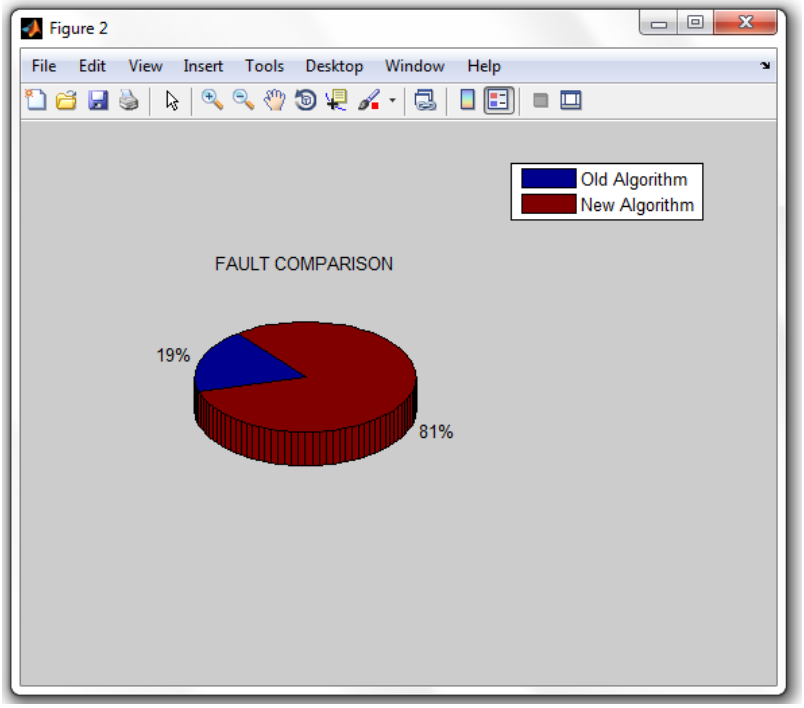

Fig. 4 Pie chart representation shows that new algorithm can detect 4 times higher faults than old algorithm.

The Fig. 3 and 4 shows the comparison of both the existing and the new technique. The graph shows that the new technique has higher rate of fault detection as compared to the existing algorithm. The comparison between the new algorithm and existing algorithm has been done by the metric, called APFD metric. In Fig 4, the graph justifies that, our prioritization order detects the higher number of fault in the early stages.

IX. CONCLUSION

Over the past few years, many of the prioritization techniques have been introduced and they have their own advantages and some limitations too. Our newly proposed technique will enhance the efficiency of regression testing. Our technique is a hybrid of the two techniques- one is Functional Dependency Technique and another is Model Based Testing Technique. Dependency approach helps us to depict the dependent functions and Model based technique helps in generating all those test cases, which are important to test the functionality of complex system. Both of them together give a stronger technique to prioritize test cases. This technique will detect fault in the early stages of development. Moreover it also provides higher rate of fault detection and will reduce the time of testing.

X. FUTURE SCOPE

The scalability of this algorithm is however be tested with larger number of project. The efficiency of the test case prioritization can also be enhanced by the use of automated tool for computing. This approach can also be extended by using the non functional features of the project.

ACKNOWLEDGEMENTS

We thank Mr. Prateek Sharma for his support and the infrastructure provided at the Lovely Professional University, Phagwara, Punjab, India.

REFERENCES

[1] G Rothermel, R.H. Untch and M.J. Harrold "*Prioritizing Test Cases For Regression Testing*" IEEE Transaction of Software Engineering Vol. 27, No.10 October 2001.

[2] Glenford J.Myers, Badgett, Todd M. Thomas, and Crey Sandler. *The Art of Software Testing*, Second Edition. John Wiley and Sons, Inc., 2004.

[3] M. J. Harold, J. A. Jone, T. Li, and D. Liang, "*Regression test selection for java software*," in Proc. of the ACM Conference on OO Programming, Systems, Languages, and Applications, 2001.

[4] X Lin "*Regression Testing in Research and Practice"* IEEE International Conference on. IEEE, *2006.*

[5] Hema Srikanth1, Laurie Williams and Jason Osborne "*System Test Case Prioritization of New and Regression Test Cases*" IEEE 2005 International Symposium.

[6] S. G. Elbaum, A. G. Malishevaky, & G Rothermel, (2002), *Test Case Prioritization: A Family of Empirical Studies*. IEEE Transaction on Software Engineering.

[7] A Gantait IBM India Pvt. Ltd Kolkata, India "*Test case Generation and Prioritization from UML Models"IEEE 2011.*

[8] X Han, H Zeng and H Gao "*A Heuristic Model-Based Test Prioritization Method for Regression Testing*" IEEE 2012 International Symposium on Computer, Consumer and Control.

[9] Qurat-ul-ann Farooq, M Zohaib Z. Iqbal Zafar I. Malik and M Riebisch "A *Model-Based Regression Testing Approach for Evolving Software Systems with Flexible Tool Support*" IEEE International Conference on the 2010, pp. 41-49.

[10] S Biswas and R Mall "*Regression Test Selection Techniques: A Survey*" Informatica 35 (2011) 289–321.

[11] P Mahali and A.A. Acharya "*Model Based Test Case Prioritization Using UML Activity Diagram and Evolutionary Algorithm*" International Journal of Computer Science and Informatics 2013.

[12] Shifa-e-Zehra Haidry and T Miller, "*Using Dependency Structures for Prioritization of Functional Test Suites,*" IEEE Transaction on software Engineering, vol. 39,no. 2,Feb 2013.

[13] J Ryser and M Glinz, "*Using Dependency Charts to Improve Scenario-Based Testing*," Proc. 17th Int'l Conf. Testing Computer Software, 2000.

[14] M Utting and B Legeard, *Practical Model-Based Testing: A Tools Approach***.** Morgan Kaufmann Publishers Inc., 2006.

BIOGRAPHY OF THE AUTHORS

[1] **Er. Neha Dwivedi** is a final semester student of the M.Tech Computer Science Engineering degree program of Lovely Professional University, Phagwara, Punjab, India. She received her B.E. Information Technology degree at University of Pune, Maharashtra, India in 2012. She will be soon joining Cognizant Technology Services. She is interested in Software Engineering, Databases, Cloud Computing and Data Mining. She can be contacted at nedwivedi@gmail.com

[2] **Dr. Rakesh Chandra Gangwar** is an Associate Professor and Head at the department of Computer Science Engineering at Beant College of Engg. & Tech., Gurdaspur, Punjab, India. He received his Ph.D. in Network Security from IIT Roorkee, M.S from BITS Pilani, and B.Tech Computer Science Engineering, CoT, GBPUAT Pantnagar, India. His research interests are mainly in the area of Cryptography and Network Security. He can be contacted at rakeshgangwar@gmail.com

[3] **Er. Dalwinder Singh** is an Assistant Professor and Head at the department of Computer Science Engineering at Lovely Professional University, Phagwara, Punjab. He received his B.E. degree in Computer Science and Engineering from SSJCOE, Jalgaon, Maharashtra, India in 2003 and M.Tech degree in Computer Science and Engineering from Guru Nanak Dev Engineering College, Ludhiana Punjab, India in 2008. His main research interests include Databases and Software Engineering. He can be contacted at dalwinder.singh@lpu.co.in

