

# Regression testing and quality improvements: a brief analysis

<sup>1</sup> Mubah Ishtiaq, <sup>2</sup>Hira Iqbal

<sup>1,2</sup>International Islamic University, Islamabad, Pakistan

Email: <sup>1</sup>mubahkhan32@gmail.com, <sup>2</sup>hirra.awan@yahoo.com

## ABSTRACT

Software testing facilitates with different methodologies. Among them regression testing has specific and valuable worth. Regression testing is merely a repetitive testing process, ensuring that the existing activities of the software are not affected by the changes being made in it. This testing basically quantifies the quality control aiming at approval of meeting the particular requirements after the code has been modified and making it sure that the presented code is not been pretentious. This paper is about analysis of regression testing and how it enhances and defines quality improvement in the particular software.

**Keywords:** software testing; regression testing; software quality; quality improvement; software regression analysis;

## 1. INTRODUCTION

The exclusive use of software testing is to progress in consistency of a software as well as the identification of faults. Software quality and maintenance aspects include and focus on regression testing [1]. Therefore, testing is a significant feature in the maturity of a software project as small bugs in the system can be catastrophic and expensive to correct after the software has already being delivered [2]. The foremost task during the development of software is to bring about the level of quality which is appropriate. There are several reasons for regression testing, one of the key reasons for regression testing is to find out whether an alteration in one part of the software affects other parts of the software [3]. In June 1972, a paper was presented in North Carolina at the opening software testing conference by Hetzel that deals with the significance of software testing methods [4]. Regression testing basically deals with the modifications, to endow with assurance the code after change performs as proposed and that the changes are not unfavorable for the unmodified part of the software [5, 6].

Regression testing is different from other types of testing that are used in development stage of the software, as it has a surpassing requirement about the reused test cases which means it recapitulate the functions which have been effectively tested in the past [7]. Due to this aspect, the automation of provoked test cases can notably improve the quality and competence of large-scale software regression testing [7]. Engstrom (2010) describes, that regression testing verifies that the formerly running software is properly operational after the modifications [8, 9]. It also enhances the code quality which indicates that it is also used to identify errors in changes done in software [10, 11].

## 2. RESEARCH METHOD

The method followed for writing this analysis was divided into several steps as shown in Figure 1.

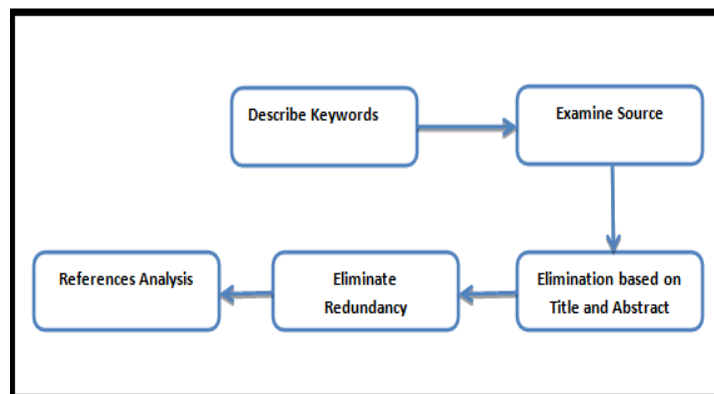


Figure. 1 Steps for analysis

Initially we focused on the keywords that were relevant with the topic we have selected for analysis. Further we checked out the sources that provided us the appropriate and significant information. Next to this we followed a criterion to include and exclude the papers that we have searched. After making the repository of the selected papers we focused on redundancy factors, if the papers were repeated or describing the same scenario we have excluded them or wrote them once.

### 2.1 Sources of information

The sources that we have explored in order to find out the research papers are listed below:

- IEEE Explore
- Science Direct
- Google Scholar
- Springer

### 2.2 Search keyword

We have used following keywords in order to search out the relevant papers.

- Software testing
- Regression testing
- Software regression testing
- Quality improvement
- Software quality
- Software regression analysis

### 2.2 Study selection criteria

Table 1 describes the selection criteria for inclusion and exclusion of a research study.

**Table. 1** Selection criteria

Applicability	Measure
1. By surfing	<ul style="list-style-type: none"> <li>✓ Contain search keywords</li> <li>✓ Text in English</li> <li>✓ Year of publication after 2010</li> <li>✓ Type of paper (Journal/Conference)</li> </ul>
2. Abstract	✓ Related to research interest and keyword
3. Citation	✓ Number of citation
4. Full Text	✓ Focus on software regression testing
5. Title	<ul style="list-style-type: none"> <li>✓ Observations</li> <li>✓ Summaries of lecture</li> <li>✓ Workshops</li> <li>✓ poster sessions.</li> </ul>

### 2.3 Data extraction

Tabular representation of some of the papers which we have studied is shown in Table 2. Other papers are presented in the reference list which helped to analyze regression testing and its impact on quality improvement.

**Table. 2** Data extraction

Sr#	Paper Title	Author	Reference type	Year
1	Regression Testing in Research and Practice	Xuan Lin	Journal	2007

2	Analysis of Different Regression Testing Approaches	Chandana Bharati & Shradha Verma	Journal	2013
3	A Maintainability Spreadsheet-Driven Regression Test Automation Framework.	Liu, Z., Chen, Q.& Jiang, X	Conference	2013
4	A Framework to Support Research in and Encourage Industrial Adoption of Regression Testing Techniques.	Kauffman, J.M.& Kapfhammer, G.M	Conference	2012
5	30 Years of Regression Testing: Past, Present and Future	Jean Hartmann	Article	2012
6	Reduce, Reuse, Recycle, Recover: Techniques for Improved Regression Testing	Mary Jean Harrold	Conference	2009
7	Testing: A Roadmap	Mary Jean Harrold	Conference	2000
8	A systematic review on regression test selection techniques	Engström, E., Runeson, P., & Skoglund, M	Journal	2010
9	A Framework for Evaluating Regression Test Selection Techniques in Industry	Alex Augustsson	Article	2012
10	A framework for the execution of automated regression testing	Arleny Rebeca Lopez Triana	Thesis	2015
11	A Literature Survey on Evaluation and Comparison of Regression Testing Techniques for Web Applications	Muhammad Hassnain, Anjum Abbas	Journal	2014
12	Parallel Computing for Continuous Regression Automation Framework	Omkar P & IIRampur Srinath	Journal	2016
13	Techniques for Improving Regression Testing in Continuous Integration Development Environments	Sebastian Elbaum , Gregg Rothermel , John Penix	Journal	2014
14	Regression Test Selection Techniques: A Survey	Swarnendu Biswas & Rajib Mall	Survey. Informatica	2010
15	A qualitative survey of regression testing practices	E. Engström and P. Runeson	Conference	2010
16	Study and Analysis of Regression Test Case Selection Techniques	Sunidhi Puri Abhishek Singhal Abhay Bansal	Journal	2014

### 3. BASIC CONCEPTS

A tabular description of few basic concepts in context to regression testing has been discussed in Table 3. For notational ease, further this analysis indicates actual given program by P with its modified version by P'. A test set that was used to previously test P is indicated by T with t indicating test case in T.

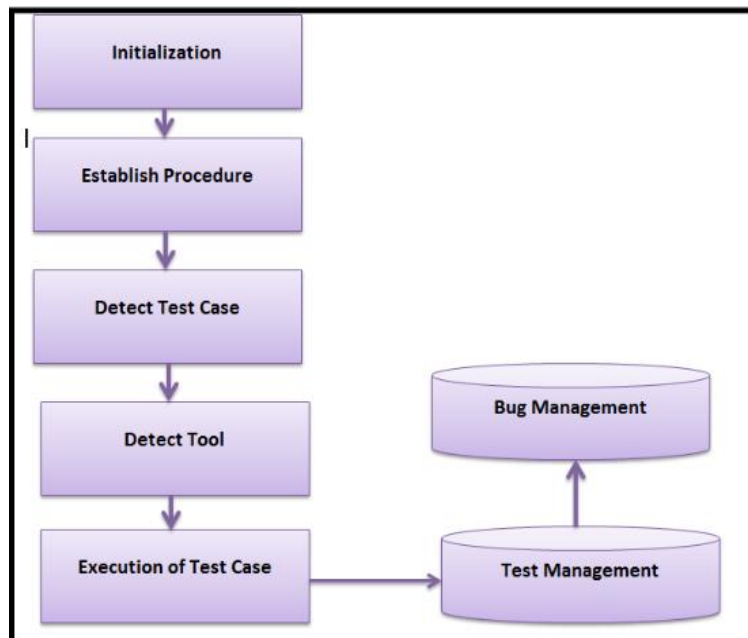
**Table. 3** Basic concepts

<b>Concepts</b>	<b>Description</b>
<b><i>Obsolete, Retestable and Redundant Test Cases</i></b>	<b>Obsolete</b> test cases are not applicable for adapted changes [12, 13]. <b>Retestable</b> test cases rerun during regression testing as it carry out the tailored and the affected parts of the program [12, 13]. <b>Redundant</b> test cases execute only the impassive parts of the program [13].
<b><i>Execution Trace of a Test Case</i></b>	It indicates the set of statements implemented when a program is validated by a test case. It is denoted by ET (P (t)), t indicates to any test case and P indicates to a program [13, 14].
<b><i>Fault-revealing Test Cases</i></b>	It points towards the failure of a program as these test cases produce erroneous outputs [15].

<i>Modification revealing Test Cases</i>	It indicates such tests cases which result in distinctive turnout for novel version and previous version of program [15].
<i>Modification-traversing Test Cases</i>	It results with beheading indications for novel version and previous version of programs [15]. It deals with changed version of the program.
<i>Inclusive, and Safe Regression Test Cases</i>	<b>Inclusiveness</b> indicates towards the extent of selection of a modification-revealing case from the original set of test cases [14, 15]. A 100% inclusive technique is called as <b>safe</b> .

#### 4. REGRESSION TESTING PROCESS

Figure 2 below describes the process or steps being involved in regression testing. The steps of regression testing are initialization, establishing a procedure, detecting test cases, detecting a tool, execution of test cases, test management and bug management.



**Figure. 2** Regression testing process

##### 4.1 Regression testing

Regression testing have different steps and each step indicates a specific problem related to regression testing. A typical regression procedure is described as [16, 17]:

**Step 1:** Assortment problem in Regression Test: test  $P'$  select a subset  $T'$  of  $T$ .

- Select  $T' \subseteq T$ , a set of test cases to execute on  $P'$ .

**Step 2 & 4:** Test suite execution problem: to check test results for exactness and competently execute the test suite.

- Test  $P'$  with  $T'$ , establishing  $P'$ 's suitability w.r.t  $T$
- Test  $P'$  with  $T''$ , establishing  $P'$ 's suitability w.r.t  $T'$

**Step 3:** Coverage identification problem: require new test cases if  $P'$  has new functionality.

- If required, form  $T''$ , for new operative set of test cases for  $P'$

**Step 5:** Test suite maintenance problem: to bring up to date and accumulate test information.

- Create T'', a new test suite and test execution profile for P', from t, T', and T''.

#### **4.2 Need of regression testing**

Regression testing is required for [18]:

- modification in needs as per the new suggestions
- addition of novel attribute
- defect amendment
- mending attainment concerns
- quality improvement

#### **4.3 Types of regression testing**

Classification of regression testing is as [19]:

- **Local** - modification brings in new error.
- **Unmasked** - newly written code causes error by now in code to reveal itself.
- **Remote** - modification in one part of the software breaks functionality in another unit or component.
- **Corrective** It is applied when conditions are unchanged and test cases can be reused. It is provoked by corrections made to the previous version.
- **Progressive** It is applied when conditions are modified and new test cases must be intended. It is provoked by new aspects added to the old version.

#### **4.4 Regression testing tools**

There are plenty of testing tools in present times [20-22]. Some of them are discussed here as tools that support regression testing in a rigorous way. Some important tools for regression testing are discussed below:

##### **TestComplete**

It is designed for utilization on Windows. It guides for testing the web pages [23]. TestComplete also manages its own submission testing and many forms of tests that can be within or apart from the system [19]. This tool affords support for daily regression testing.

##### **QEngineForms**

It is independent test automation tool which is used for web operations and performance [24]. It supports testing of applications which are matured using HTML, PHP, .NET, XML. This tool is developed using java [24].

##### **Junit**

It is a famous open source regression testing framework. This is used to ease the automatic execution and management of test cases [25].

##### **Selenium**

A unique tool designed for browser constructed regression testing. This open source tool is used for systematizing web applications.

##### **Rational functional tester**

A java tool used for automated test cases of software application [26]. This tool is used for systematizing regression test cases, it assimilates with rational test manager [26].

#### **4.5 Regression testing technologies**

The Figure 3 below describes the technologies of regression testing which covers the steps beings discussed above.

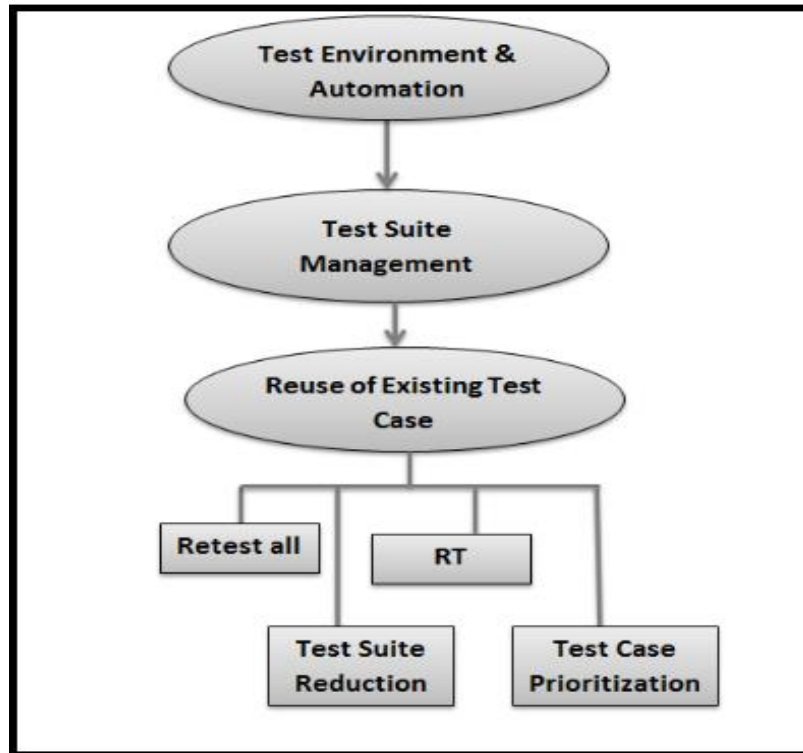


Figure. 3 Regression testing technologies

#### 4.5.1 Test environments and automation

Test environment and automation is being discussed by several authors [27-30]. They discussed that regression testing process is a problem for test environment. Through regression testing achievement of improved system quality and maintenance costs is desirable.

#### 4.5.2 Test suite management

Test suite management is also focused by different authors [30-33]. When an adaptation occurs some test cases are selected from ancient test cases, test cases that are out dated needed to be deleted, novel test cases should be adopted for new functionality. The change that occur during deletion and innovation of test cases should be accomplished [16].

#### 4.5.3 Reuse of existing test cases

The following options are taken into account.

- **Retest-all**

When the set of instructions is improved, tester has two main procedures to test that improved instructions [34]. In order to decrease cost, some of test cases that are yet present in test suites should be omitted [35]. To rerun again all test cases is placed in retest all [34, 35]. Retest all methodology established test suite performed on modified program P' [34, 35].

- **Regression test selection**

The selection of regression tests is carried out to reduce cost. Hence, all the test cases present in the test suite are not executed rather some of the test cases denoted as T' are gathered from set T and only selected test cases T' will be executed on improved program P' [16]. In selection of regression test cases, the selected process can't equalize the execution of extra test cases [18].

- **Test case prioritization**

In recycling of regression testing some of the test cases are used to test the new program with respect to the order of priority, so that important test cases should be executed first [36, 37]. The penalty area for prioritization of test cases is to upsurge the error gratitude [21, 36, 37].

- **Test suite reduction**

One of another technology used in regression testing is test suite reduction. In this technique, some test cases are eliminated in order to reduce the cost as well to reduce size of test suite [38].

## **5. REGRESSION TESTING AND QUALITY IMPROVEMENT**

Regression testing is representation of overall test effort on each application release. Regression testing detects problems that are not detected yet [2]. Sometimes regression testing can't test hidden bugs and these bugs are found after deployment which has major effect on cost, resources as well as time [2]. The costliest action that happened during the software development and maintenance is testing [39]. Much of the cost is incurred upon testing and retesting, which is often considered as regression testing [39]. According to some reports regression testing spend 80% of its economical amount and can put away 50% of amount of software maintenance [39].

In regression testing, time can be reduced by creation of effective test suits that tests the alternative component, by identifying test cases in the regression test suites. These test cases should not rerun by detecting and eliminating complete test cases [2, 39]. For improving software quality, test suites can be recycled by effective ordering that are created for one version of software [39]. The quality costs are those that would not be rejected even if all the work has been done well in their jobs [40]. The quality cost can be divided into two kinds: conformance and non-conformance cost that is conformance is related to improvement of existing product and process whereas cost of non-conformance is spent on internal failure before product is shipped to customer [40].

Regression testing has variety of topics including test environment as well as test automation [28]. In regression testing, selective retest technique plays an important role in quality improvement as this technique reduces cost by reprocessing current tests and it can identify a portion that is modified or conditions that should be tested [28, 33]. Selective testing technique is unique from all other techniques, which runs all test cases in current test suits [33]. Selective retest technique in regression testing is cost-effective rather than other testing techniques [28, 33], only when cost of selecting subsets that will run less than cost of running all test cases. Thus, selective regression testing technique should not be omitted [28, 33, 39].

## **6. CONCLUSION**

This paper was intended to analyze regression testing and its association with quality improvement. We studied research studies which concluded that for improvement of a software the regression testing is very essential and it provides high rate increase in quality by minimizing the cost and increasing the maintainability. This paper presents a short analysis but it summarized that when and how regression testing works, what are its parameters and how it helps in enhancing quality. Some specific tools are briefly described as tools play a significant role in regression testing.

## **ACKNOWLEDGEMENT**

Authors are thankful to International Islamic University for providing resources to complete this research.

## **REFERENCES**

1. Panda, M. and D.P. Mohapatra. *Automated Graphical User Interface Regression Testing*. in *Proceedings of International Conference on Internet Computing and Information Communications*. 2014: Springer.
2. Torniainen, S., *IMPROVING EFFECTIVENESS OF REGRESSION TESTING OF TELECOMMUNICATIONS SYSTEM SOFTWARE*. 2008, HELSINKI UNIVERSITY OF TECHNOLOGY.
3. Hartmann, J., *30 Years of Regression Testing: Past, Present and Future*. 2012.
4. Hetzel, W.C. *Program test methods*. in *Computer Program Test Methods Symposium (1972: University of North Carolina)*. 1973: Prentice-Hall.

5. Rothermel, G. and M.J. Harrold. *Selecting tests and identifying test coverage requirements for modified software*. in *Proceedings of the 1994 ACM SIGSOFT international symposium on Software testing and analysis*. 1994: ACM.
6. Bharati, C. and S. Verma, *Analysis of different regression testing approaches*. International Journal of Advanced Research in Computer and Communication Engineering, 2013. **2**(5).
7. Xiaowen, L., *Research on Regression Testing Methods for Industry Applications*. International Journal of Smart Home, 2013. **7**(6): p. 111-122.
8. Engström, E., *Exploring regression testing and software product line testing-research and state of practice*. 2010, Lund University.
9. Engström, E., P. Runeson, and M. Skoglund, *A systematic review on regression test selection techniques*. Information and Software Technology, 2010. **52**(1): p. 14-30.
10. Oezbek, C. *Introducing Automated Unit Testing into Open Source Projects*. in *IFIP International Conference on Open Source Systems*. 2010: Springer.
11. Rebeca, A. and L. Triana, *A framework for the execution of automated regression testing*. 2015.
12. Iqbal, M.Z.Z., Z.I. Malik, and A. Nadeem. *An approach for selective state machine based regression testing*. in *Proceedings of the 3rd international workshop on Advances in model-based testing*. 2007: ACM.
13. Puri, S., A. Singhal, and A. Bansal, *Study and Analysis of Regression Test Case Selection Techniques*. International Journal of Computer Applications, 2014. **101**(3).
14. Biswas, S., et al., *Regression test selection techniques: A survey*. Informatica (Slovenia), 2011. **35**(3): p. 289-321.
15. Panigrahi, C.R. and R. Mall, *A hybrid regression test selection technique for object-oriented programs*. Int. J. Soft. Eng. Applic, 2012. **6**(4): p. 17-34.
16. Rothermel, G. and M.J. Harrold, *Analyzing regression test selection techniques*. IEEE Transactions on Software Engineering, 1996. **22**(8): p. 529-551.
17. Lin, X., *Regression testing in research and practice*. Computer Science and Engineering Department University of Nebraska, Lincoln, 2007: p. 1-402.
18. Rajal, J.S. and S. Sharma, *A Review on Various Techniques for Regression Testing and Test Case Prioritization*. International Journal of Computer Applications, 2015. **116**(16).
19. Testing, O.S. *Regression Testing*. 2017 [cited 2017 2017]; Available from: <http://www.onestoptesting.com/regressiontesting/types.asp>.
20. Finisar. *QLogic Troubleshoots*. [cited 2017; Available from: <http://www.finisar.com/nt/documents/QLogicTroubleshootsiSCSISANs.pdf>
21. Elbaum, S., A.G. Malishevsky, and G. Rothermel, *Test case prioritization: A family of empirical studies*. IEEE Transactions on Software Engineering, 2002. **28**(2): p. 159-182.
22. Devdirect. *Testing PCAT 1986*. [cited 2017; Available from: [http://www.devdirect.com/ALL/Testing\\_PCAT\\_1986.aspx](http://www.devdirect.com/ALL/Testing_PCAT_1986.aspx).
23. Automatedqa. *TestComplete*. 2017 [cited 2017; Available from: <http://www.automatedqa.com/products/testcomplete/>.
24. Adventnet. *QEngineforms*. 2017 [cited 2017; Available from: <http://www.adventnet.com/products/qengine/index.html>.
25. Junit. *Junit*. 2017 [cited 2017; Available from: <http://www.junit.org/index.htm>
26. IBM. *Rational functional tester*. 2017 [cited 2017; Available from: <http://www-306.ibm.com/software/awdtools/tester/functional/features/index.html>.
27. Hoffman, D. and C. Brealey, *Module test case generation*. Vol. 14. 1989: ACM.
28. Brown, P.A. and D. Hoffman, *The application of module regression testing at TRIUMF*. Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, 1990. **293**(1): p. 377-381.
29. Ziegler, J., J.M. Grasso, and L.G. Burgermeister. *An Ada based real-time closed-loop integration and regression test tool*. in *Software Maintenance, 1989., Proceedings., Conference on*. 1989: IEEE.
30. Harrold, M.J., R. Gupta, and M.L. Soffa, *A methodology for controlling the size of a test suite*. ACM Transactions on Software Engineering and Methodology (TOSEM), 1993. **2**(3): p. 270-285.
31. Lewis, R., D. Beck, and J. Hartmann, *Assay-a tool to support regression testing*. ESEC'89, 1989: p. 487-496.



32. Hartmann, J. and D. Robson. *Revalidation during the software maintenance phase*. in *Software Maintenance, 1989., Proceedings., Conference on*. 1989: IEEE.
33. Taha, A.-B., S.M. Thebaut, and S.-S. Liu. *An approach to software fault localization and revalidation based on incremental data flow analysis*. in *Computer Software and Applications Conference, 1989. COMPSAC 89., Proceedings of the 13th Annual International*. 1989: IEEE.
34. Onoma, A.K., et al., *Regression testing in an industrial environment*. Communications of the ACM, 1998. **41**(5): p. 81-86.
35. White, L., *Insights Into RegressionTesting*. Proceedings of the Conference on Software Maintenance-1989, pages 60-69. IEEE Computer Society Press, October 1989.
36. Rothermel, G., et al., *Prioritizing test cases for regression testing*. IEEE Transactions on Software Engineering, 2001. **27**(10): p. 929-948.
37. Wong, W.E., et al. *A study of effective regression testing in practice*. in *Software Reliability Engineering, 1997. Proceedings., The Eighth International Symposium on*. 1997: IEEE.
38. Chen, T.Y. and M.F. Lau, *Dividing strategies for the optimization of a test suite*. Information Processing Letters, 1996. **60**(3): p. 135-141.
39. Harrold, M.J. *Reduce, reuse, recycle, recover: Techniques for improved regression testing*. in *Software Maintenance, 2009. ICSM 2009. IEEE International Conference on*. 2009: IEEE.
40. Campanella, J. *Principles of quality costs: Principles, implementation, and use*. in *ASQ World Conference on Quality and Improvement Proceedings*. 1999: American Society for Quality.

#### **AUTHORS PROFILE**

**Mubah Ishtiaq** is an endeavouring researcher in the field of image processing, completed her BSCS from MUST Mirpur, Pakistan. At present, she is student of MS Software Engineering at International Islamic University Islamabad, Pakistan. Her affection in research is strongly associated to, model based testing, software system architecture and quality assurance.

**Hira Iqbal** is a fresh researcher and she received her Bachelor Degree in Computer Science from UAJK, Pakistan. Now she is perusing her Ms Degree in Software Engineering from International Islamic University Islamabad, Pakistan. Her interests in research include software quality assurance, global software development.