

## **Role Of Software Testing Life Cycle(STLC) In Software Development Life Cycle (SDLC)**

*Mithelesh Parihar*

*Research Scholar*

*Sun Rise University Alwar (Rajasthan)*

**&**

*Dr. Anu Bharti*

*Associates Professor*

*Department of Computer Science & Engineering*

*Sun Rise University Alwar (Rajasthan)*

### **ABSTRACT**

This paper reveals the concept of STLC (Software testing life cycle) and its role in SDLC. SDLC is a structure imposed on the development of software product. Testing is an important part of software development. Testing should be started as early as possible to make it a part of a process of deciding requirements. In this paper I have explained various phases of software development and importance of testing in SDLC. Software Testing Life Cycle is the testing process which is executed in systematic and planned manner. In STLC process, different activities are carried out to improve the quality of the product. One of the phases that these two processes undergo is requirements gathering. Gathering of requirements is done by the business analyst in SDLC where the development team analyses the requirements right from the design architecture to the coding into perspective. STLC on the other hand uses requirements gathering for testing, review, and analysis of the requirements. The testing team finds the requirements needed such as the types of testing necessary, and a thorough review of the requirements ensures a logical functional relationship of features and modules. This ensures that any problem or gaps are caught at an early stage. SDLC is a systematic approach to develop a software and STLC is the process of testing a software in a systematic and planned way. The testing team finds the requirements needed such as the types of testing necessary, and a thorough review of the requirements ensures a logical functional relationship of features and modules.

**Keywords:** STLC, SDLC, Software Testing, Software Development Process, V-Model

### **I. INTRODUCTION**

Software testing is an integral part of the software development life cycle (SDLC). SDLC refers to software development life cycle, while STLC refers to software testing life cycle. Both of these contain six steps which present logical differences in between them: **Pressman (2001)**. These two processes are compared below in detail with a thorough review of both their differences and similarities. In the design phase, SDLC has a technical architect whose function is

to ensure that high level and low designs of the software are met. The business analyst also comes in here to create a user interface design for the application. STLC has the test architect acting as the lead manager in doing the test planning and the identification of the high level testing points. It is at this stage that there is detailing of requirements. Then comes the coding or development phase which the SDLC development team handles. In this stage the actual development which refers to coding occurs and this is based on the design architecture. The testing team in STLC, on the other hand, then gets to focus their energy on writing the detailed test cases. The fourth stage is the testing phase where in SDLC there is actual testing of the developed code. Within this phase, unit testing, integration testing and system testing is done. Any other tests that need to be done are handled here in SDLC. In STLC, there is test execution too in this stage in addition to reporting any errors that are found. Also this is a stage where manual reporting, automation, and testing is performed to ensure that the developed code functions as it should be done: **Bertolino (2004)**. Retesting and regression testing are also performed within this stage. The overall function of the testing phase in STLC is to get a review of test cases and scenarios of the test. Next up is the application deployment phase where SDLC deploys applications that have passed the last 4 stages. The deployment is done through the production environment to the ideal and real end users. In STLC, this is the final testing and implementation stage. Testing is done and a final report is prepared. The last phase is the maintenance phase which is a continuous one. In SDLC, it features post production and deployment support and follow up with enhancements as they continue to be developed. STLC on the other hand features updating and maintenance of test plans, and testing and support of test phases, as well as enhancements as part of maintenance. There are major importance of testing in the part of SDLC and it is better to introduce testing in the early stage of SDLC phases so it help to identify the defects in the early stage & try to avoid the bugs finding & get resolve in the last critical stage.

## **II. SOFTWARE TESTING**

Software testing is an investigation conducted to provide stakeholders with information about the quality of the product or service under test. Software testing can also provide an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation: **Mathur (2008)**. Test techniques include the process of executing a program or application with the intent of finding software bugs (errors or other defects). Software testing can provide objective, independent information about the quality of software and risk of its failure to users and/or sponsors. Software testing can be conducted as soon as executable software (even if partially complete) exists. According to **Myers (2004)** testing is the process of executing a program with the intent of finding errors. Testing is the process of executing a program with the intent of finding errors. Before explaining the role of Software testing, I would like to explain different phases of Software Development Life Cycle (SDLC) process and Software Testing Life Cycle (STLC).

### **a. The phases of SDLC are as follows:**

1. Requirement gathering/problem definition
2. Requirement Specification
3. Software Design

4. Coding
5. Software Testing and evaluation
6. Debugging
7. Software Maintenance

**1. Requirement Gathering:** In this phase the needs of the user are gathered and translated into a written set of requirements. These requirements do not include any technicalities according to the developer.

**2. Requirement Specification:** In this the user requirements are specified in the developer's terminology. It describes the requested behavior of a required system.

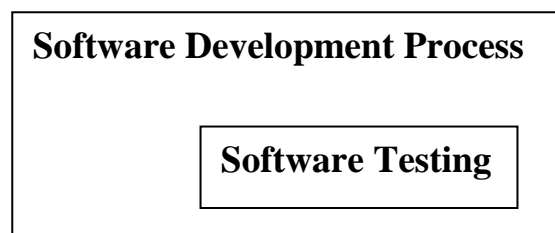
**3. Software Design:** It is the process of translating user requirements into a set of external interfaces. It includes both the problem solving and planning up solution for the software.

**4. Coding:** In this phase coding is done based on the design document for a module.

**5. Software Testing:** It includes software testing verification and validation of the system just build.

**6. Debugging:** It is not a part of testing domain. It is a separate process performed as a consequence of testing. The goal of this phase is the failure with the help of symptoms identified, locate the bugs and errors and finally correct it.

**7. Software maintenance:** In this software is being corrected or enhanced after delivery of the final product.



**Fig.1 Testing process emerged out of development process.**

## **b. Maintenance Activities**

**Four maintenance activities are as follows:**

- Adaptive Maintenance
- Perfective Maintenance
- Preventive Maintenance
- Corrective Maintenance

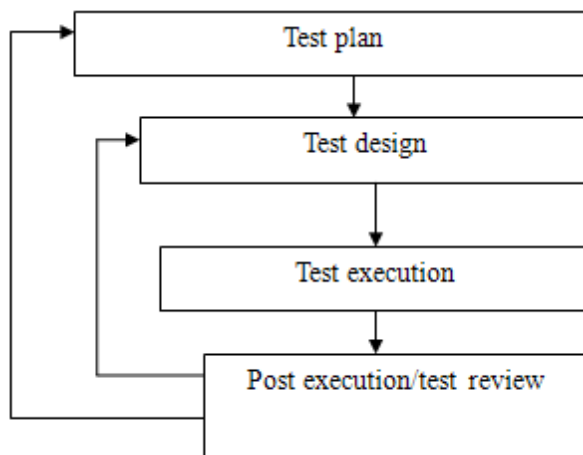
Software development is an engineering activity for a quality product, it consists of many processes, and Software testing has also emerged as a complete process in software engineering. Therefore our major concern in this paper to show that software testing is a process which runs parallel to SDLC as shown in Table.1. Software testing is a process that must be planned, specified, designed, implemented and quantified. Testing is a dual purpose process, as it is used to detect bugs as well as to establish confidence in the quality of software. The testing process divided into a well-defined sequence of steps is termed as STLC. STLC involves the testers at early stages of development. This has significant benefit in the project schedule and cost. STLC involves the testers at early stages of development. This has significant benefit in the project schedule and cost. The testing process for software development life cycle is a significant activity of software product as shown in above Fig. 1.

**Table1: Software testing is a process which runs parallel to SDLC**

Software Development Process	Software Testing
Requirement gathering	
Requirement Specification	Test Plan
Design	Test Case Design
Code	Test Execution
.....	.....
.....	.....

**c. STLC consists of the following phases are shown in Fig.2.**

1. Test Plan
2. Test Design
3. Test Execution
4. Post Execution/Test Review



**Fig.2 STLC Phases**

1. **Test Plan:** The main goal of this phase is to consider the important issues of testing strategy like resources, schedules, responsibilities, risks and priorities as a roadmap. The output of this phase is test plan document. Test plans are developed every phase of SDLC.
2. **Test Design:** In this phase Test Cases are designed. It includes the following activities.
  - Determine the test objectives.
  - Preparation of list of items to be tested.
  - Identification of test cases.
  - Selection of test case design techniques.
  - Creating test cases and test data.
  - Setting up the test environment and supporting tools.
  - Creating test procedure specification.
3. **Test execution:** In this phase, all the test cases are executed including verification and validation. Verification test cases are started at the end of each phase of SDLC. Validation test cases are started after the completion of a module. Test results are documented in the test incident reports, test logs, testing status and test summary reports.
4. **Post Execution/ test review:** After the successful test execution, bugs will be reported to the concerned developers. This phase is to analyze bugs related issues and get feedback so that maximum number of bugs can be removed. After fixing the bug, the developer reports to the testing team and modified portion of the software is tested once again. The final bug report is reviewed and analyzed for overall testing process.

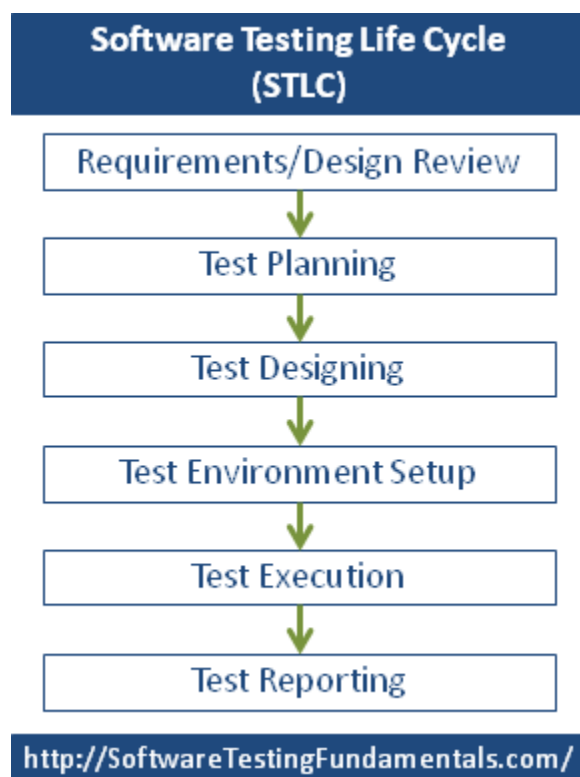
**d. Software Testing Life Cycle (STLC):** Software testing life cycle defines the steps/ stages/ phases in testing of software: **Beizer (2002)**. STLC is a very important phase of SDLC and the final product or the software cannot be released without passing through the STLC process. Each phase has different goals and deliverables. STLC is limited to the testing phase where quality of software or product ensures. SDLC has vast and vital role in complete development of a software product.

### **III. OVERVIEW OF SOFTWARE TESTING LIFE CYCLE (STLC)**

- STLC is an integral part of Software Development Life Cycle (SDLC). But, STLC deals only with the testing phases.
- STLC starts as soon as requirements are defined or SRD (Software Requirement Document) is shared by stakeholders.
- STLC provides a step by step process to ensure quality software.

- In the early stage of STLC, while the software or the product is developing, the tester can analyze and define the scope of testing, entry and exit criteria and also the Test Cases. It helps to reduce the test cycle time along with better quality.
- As soon as the development phase is over, the testers are ready with test cases and start with execution. This helps to find bugs in the initial phase.

Nevertheless, Software Testing Life Cycle, in general, comprises of the following phases, shown in Fig.3 and Table 2.



**Fig.3 Software Testing Life Cycle**

**Table2: Software Testing Life Cycle Phases**

Phase	Activity	Deliverables	Necessity
Requirements/ Design Review	We review the software requirements/ design (Well, if they exist.)	<ul style="list-style-type: none"> <li>Review Defect Reports</li> </ul>	Curiosity

Test Planning	Once you have gathered a general idea of what needs to be tested, you 'plan' for the tests.	<ul style="list-style-type: none"> <li>• Test Plan</li> <li>• Test Estimation</li> <li>• Test Schedule</li> </ul>	Farsightedness
Test Designing	We design/ detail your tests on the basis of detailed requirements/design of the software (sometimes, on the basis of your imagination).	<ul style="list-style-type: none"> <li>• Test Cases/Test Scripts /Test Data</li> <li>• Requirements Traceability Matrix</li> </ul>	Creativity
Test Environment Setup	We setup the test environment (server/ client/ network, etc.) with the goal of replicating the end-users environment.	<ul style="list-style-type: none"> <li>• Test Environment</li> </ul>	Rich company
Test Execution	We execute your Test Cases/ Scripts in the Test Environment to see whether they pass.	<ul style="list-style-type: none"> <li>• Test Results (Incremental)</li> <li>• Defect Reports</li> </ul>	Patience
Test Reporting	We prepare various reports for various stakeholders.	<ul style="list-style-type: none"> <li>• Test Results (Final)</li> <li>• Test/ Defect Metrics</li> <li>• Test Closure Report</li> <li>• Who Worked Late &amp; on Weekends (WWLW) Report [Depending on how fussy your Management is]</li> </ul>	Diplomacy

The rule for development of a test strategy is that testing begins from the smallest unit and progresses to enlarge. This means testing strategy should start at the component level and finish at the integration of the entire system. As the project under consideration starts, testing too start from the first level of SDLC, therefore the test strategy should be such that the testing process

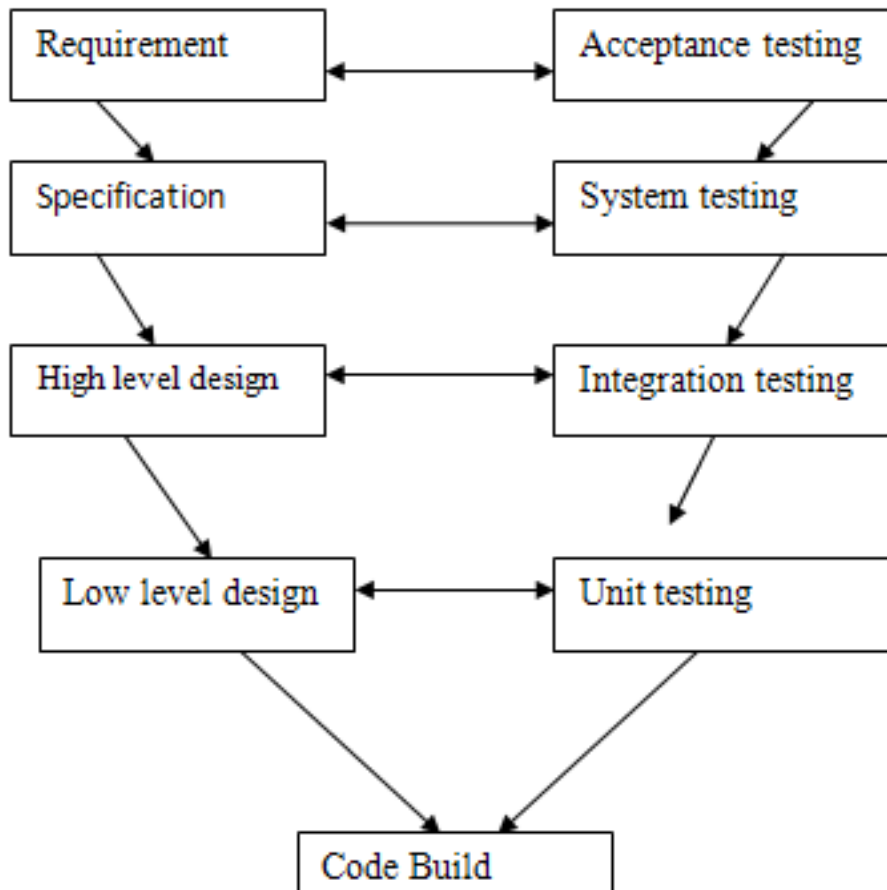
continues till the implementation of project as shown in the Fig.3. To show the coordination between development process and testing various models exist but here we consider V model: Copeland (2004).

#### IV. V-MODEL

The V-model is an SDLC model where execution of processes happens in sequential manner in a V-shape. It is also known as verification and validation model. The V-Model demonstrates the relationships between each phase of the development life cycle and its associated phase of testing: Pressman (2001). Although variants of the V-model exist, a common type of V model uses four test levels, corresponding to the four development levels as shown in Fig.4.

The four levels frequently used are as follows:

- Component(Unit)Testing
- Integration Testing
- System Testing
- Acceptance Testing





**Fig.4 V-Model uses four test levels**

- **Unit Testing:** In this phase all the unit test case, created in the Low level design phase are executed. Unit testing is a white box testing technique, where a piece of code is written which invokes a method (or any other piece of code) to test whether the code snippet is giving the expected output or not. This testing is basically performed by the development team. In case of any anomaly, defects are logged and tracked: **Jorgensen (2002)**.
- **Integration Testing:** In this phase the integration test cases are executed which were created in the Architectural design phase. In case of any anomalies, defects are logged and tracked. Integration testing is a technique where the untested modules are integrated and tested whether the integrated modules are rendering the expected results. In simpler words, it validates whether the components of the application work together as expected: **Jorgensen (2002)**.
- **Systems testing:** In this phase all the system test cases, functional test cases and nonfunctional test cases are executed. In other words, the actual and full fledged testing of the application takes place here. Defects are logged and tracked for its closure. Progress reporting is also a major part in this phase. The traceability metrics are updated to check the coverage and risk mitigated: **Jorgensen (2002)**.
- **User acceptance Testing:** Acceptance testing is basically related to the business requirements testing. Here testing is done to validate that the business requirements are met in the user environment. Compatibility testing and sometimes nonfunctional testing (Load, stress and volume) testing is also done in this phase: **Jorgensen (2002)**.

## **V. STLC PHASES IN SOFTWARE TESTING**

STLC has the following different phases but it is not mandatory to follow all phases: **Pressman (2001)**. Phases are dependent on the nature of the software or the product, time and resources allocated for the testing and the model of SDLC that are as follows:

**Requirement Analysis:** When the System Requirements Document (SRD) is ready and shared with the stakeholders, the testing team starts high level analysis concerning the Application Under Test (AUT).

- **Test Planning:** Test Team plans the strategy and approach.
- **Test Case Designing/Development:** Develop the test cases based on scope and criteria's.
- **Test Environment Setup:** When integrated environment is ready to validate the product.

- **Test Execution:** Real-time validation of product and finding bugs.
- **Test Closure:** Once testing is completed, matrix, reports, results are documented.

The following lists the factors of comparison study between SDLC and STLC based on their phases is shown in Table 3.

**Table 3:Comparative Study of SDLC and STLC**

Phase	SDLC	STLC
Requirement Gathering	<ul style="list-style-type: none"> <li>• Business Analyst gathers requirements.</li> <li>• Development team analyzes the requirements.</li> <li>• After high level, the development team starts analyzing from the architecture and the design perspective.</li> </ul>	<ul style="list-style-type: none"> <li>• Testing team reviews and analyzes the SRD document.</li> <li>• Identifies the testing requirements - Scope, Verification and Validation key points.</li> <li>• Reviews the requirements for logical and functional relationship among various modules.</li> <li>• This helps in the identification of gaps an early stage.</li> </ul>
Design	<ul style="list-style-type: none"> <li>• The architecture of SDLC helps us develop a high-level and low-level design of the software based on the requirements.</li> <li>• Business Analyst works on the mocker of UI design.</li> <li>• Once the design is completed, it is signed off by the stakeholders.</li> </ul>	<ul style="list-style-type: none"> <li>• In STLC, either the Test Architect or a Test Lead usually plans the test strategy.</li> <li>• Identifies the testing points.</li> <li>• Resource allocation and timelines are finalized here.</li> </ul>
	<ul style="list-style-type: none"> <li>• Development team starts developing the software.</li> </ul>	<ul style="list-style-type: none"> <li>• Testing team writes the test scenarios to validate the quality of</li> </ul>

Development	<ul style="list-style-type: none"> <li>• Integrate with different systems.</li> <li>• Once all integration is done, a ready to test software or product is provided.</li> </ul>	<p>the product.</p> <ul style="list-style-type: none"> <li>• Detailed test cases are written for all modules along with expected behavior.</li> <li>• The prerequisites and the entry and exit criteria of a test module are identified here.</li> </ul>
Environment Setup	<ul style="list-style-type: none"> <li>• Development team sets up a test environment with developed product to validate.</li> </ul>	<ul style="list-style-type: none"> <li>• The Test team confirms the environment set up based on the prerequisites.</li> <li>• Performs smoke testing to make sure the environment is stable for the product to be tested.</li> </ul>
Testing	<ul style="list-style-type: none"> <li>• The actual testing is carried out in this phase. It includes unit testing, integration testing, system testing, defect retesting, regression testing, etc.</li> <li>• The Development team fixes the bug reported, if any and sends it back to the tester for retesting.</li> <li>• UAT testing performs here after getting sign off from SIT testing.</li> </ul>	<ul style="list-style-type: none"> <li>• System Integration testing starts based on the test cases.</li> <li>• Defects reported, if any, get retested and fixed.</li> <li>• Regression testing is performed here and the product is signed off once it meets the exit criteria.</li> </ul>
Deployment/ Product Release	<ul style="list-style-type: none"> <li>• Once sign-off is received from various testing team, application is deployed in prod environment for real end users.</li> </ul>	<ul style="list-style-type: none"> <li>• Smoke and sanity testing in production environment is completed here as soon as product is deployed.</li> <li>• Test reports and matrix preparation are done</li> </ul>

		by testing team to analyze the product.
Maintenance	<ul style="list-style-type: none"><li>• It covers the post deployment supports, enhancement and updates, if any.</li></ul>	<ul style="list-style-type: none"><li>• In this phase, the maintaining of test cases, regression suits and automation scripts take place based on the enhancement and updates.</li></ul>

## VI. IMPORTANCE OF TESTING

Both system development process and testing are parallel to each other. STLC is a part of SDLC. It is like a set and subset, we cannot have STLC running individually on its own. STLC is the most important part of SDLC one cannot release the final product without running it through STLC process: **Pressman (2001)**. Advantages of software testing in the Software Development Life Cycle:

- Testing should be introduced in the early stage of the SDLC, The cost of fixing the bug is larger if testing is not done in early stage & bugs found in later stages.
- In the today's competitive market only the quality product stays long time firmly, so to make sure the product the good quality product the testing of application is key factor in SDLC.
- As it not possible makes it software application is defect free but testing will be necessary.
- Most important thing of testing is the development environment is different than the Testing environment and the testing done on testing environment is similar to the production environment. Testing play a vital role in the development process.
- Testing has a significant part in SDLC although the testing also upgrades the standards of the software and programmed by recognizing errors prior in the system. It also upgrades the standard of organization.

## VII. CONCLUSIONS

In this chapter, I learned that complete, exhaustive Software Development Life Cycle and Software Testing Life Cycle. Software testing is an important step in a product's life cycle, as it will determine whether a product works correctly and efficiently according to the requirements of customers. Software testing is a process used to identify the correctness, completeness and quality developed computer software. SDLC and STLC look at both the development and testing phases of software. Differences occur in six main areas which outline the entire production environment, namely requirements gathering, coding, design, testing deployment, and

maintenance. Software Development Life Cycle (SDLC) is a procedural process in the development of a software product. This process is carried out in a series of steps, which explains the whole idea behind the development of a software product for the testing fundamental concepts. It is important to note that STLC is contained in SDLC, as testing is mainly encompassed in the umbrella of software development.

In the today's fast growing world of the Internet customer wants to get products to be implemented and updated faster than their competitors. SDLC and STLC makes vital role in testing. Software testing strategies and techniques is an important for improvement and measurement of a software and software development life cycle. It represents the review of specification, design and coding. Software testing also provides an objective, independent view of software to allow the business to appreciate and understand the risk at implementation. Testing is under SDLC, it should be understood that testing is an independent function in the software development process and software testing.

Lastly, we have explained various phases of software development and importance of testing in SDLC. It represents the review of specification, design and coding. Software testing also provides an objective, independent view of software to allow the business to appreciate and understand the risk at implementation. Various types of software strategies used for conventional and object oriented software development life cycle process.

## REFERENCES

1. Bezier, B. (1990), "Software Testing Techniques," 2nd Edition, International Thomson Computer Press, New York, Vol.18, No.3, pp.82-88.
2. Bertolino, A. (2004), "Software Testing In: *Guide to the Software Engineering Body of Knowledge (SWEBOK)*", 2<sup>nd</sup> Edition, IEEE Computer Society, Chapter 5, Prentice Hall, Inc., New Jersey, Vol.9, No. 2, pp. 5-16.
3. Copeland, L. (2004), "A Practitioner's Guide to Software Test Design", Boston: Artech House Publishers, 2<sup>nd</sup> Edition, Vol.2, No.6, pp.280-283.
4. Jorgensen, P. (2002), "Software testing: A Craftsman's Approach", 2<sup>nd</sup> Edition, Auerbach Publications Boston, MA, USA, CRC Press, Volume 32, No.1, p. 167-171.
5. Mathur, A.P. (2008), "Foundation of Software Testing", 1<sup>st</sup> Edition, Published by John Wiley & Sons, Inc., Hoboken, New Jersey, *Pearson Education Inc.*
6. Myers, G. J. (2004), "The Art of Software Testing", Second Edition" Published by John Wiley & Sons, Inc., Hoboken, New Jersey.
7. Pressman, Roger S. (2001), "Software Engineering: A Practitioner's Approach", 5<sup>th</sup> Edition, Boston: McGraw-Hill.