**International Journal of Advanced Science and Research**
**ISSN: 2455-4227, Impact Factor: RJIF 5.12**
**www.allsciencejournal.com**
**Volume 1; Issue 11; November 2016; Page No. 31-35**

# Importance of testing in quality assurance and system development life cycle

**Er Suruchi**

Assistant Professor, Department of Computer Science and Engineering, Gian Jyoti Group of institutions, Chandigarh, Punjab India

**Abstract**
In every organization, testing is an important and valuable phase in the software development life cycle. However, the way it is carried out differs from one organization to another. Software testing has become the part of development and it is better to start testing from the initial stages, in order to avoid difficulty by correcting the bug at the last stage. The major role of testing involves checking, that there should be no discrepancy in the software development process anywhere throughout. In my paper I have explained various phases and importance of testing in software development life-cycle. The importance of testing in software development life cycle is to improve reliability, performance and other important factors, which may define under SRS (software requirement specification). Customer can wait more for software release, but they don't like to work with defected software.

**Keywords:** software development life cycle, software development process, software testing

## 1. Introduction

Software testing plays a vital role in the software development life-cycle to recognize the difficulties in the process very well. [1] SDLC stands for Software Development Lifecycle and it is the process of developing information system with proper analysis, design, implementation and maintenance. During the software development lifecycle errors occur and defects are inevitably introduced. Most organizations are aware of the importance of testing within the software development life-cycle in order to detect and remove these defects. Research has shown that the test process frequently accounts for 40% of the cost of software development. With the growing requirement for high quality and efficiency, it is becoming increasingly important for organizations to improve their soft-ware testing.

Waterfall is a sequential and non-iterative SDLC model which describes flowing of phases downwards one by one. The process does not start a phase unless the previous phase is completed once and for all completely. The waterfall model consists of the following phases:

1. Requirements gathering 2. Design 3. Implementation
4. Testing 5. Maintenance

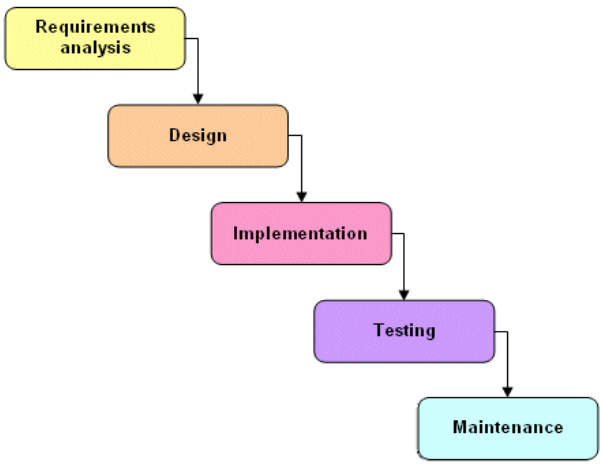



**Fig 1:** Waterfall Model

Before explaining the importance of testing in SDLC, I would like to explain different phases of software development process.

## 2. Phases in Software Development Lifecycle

Testing phase has much importance in SDLC due to a major role in debugging and error correction. The phases of SDLC is being followed in both testing and development cycle of any software application. Here are the phases of SDLC that is being followed:

### 2.1 Feasibility Study/Problem Definition

An investigation about requirement is conducted to state the problem to be solved. Main work in this phase is to determine the condition to meet for a new product or altered product [1].

### 2.2 Requirements Gathering and Analysis

Under this phase, proper requirements of project are gathered. All close functions are brought in to focus. All kinds of re-quirements and analysis of user requirement are done in this phase.

### 2.3 System Design

This is the next phase in SDLC where a rough system design is made. With all data and information being gathered, a system design is made [2].

Some of the main design concepts are:

1) Abstraction
2) Modularity
3) Data structure
4) Information hiding, and
5) Structural partitioning

### 2.4 Coding

**This** is the next phase after system design when development of project is made. According to design, proper coding is done to gain that design. Programming language might be selected according to the project.

### 2.5 System Testing
Just after development phase, testing is carried out to know the outcome of application. Testing is made to know the actual result and the expected result [3].

### 2.6 Operations and Maintenance
This is the final stage of SDLC, where the software that is being developed is being distributed to end users who are responsible for maintaining and using it for proper operations. The software that is being developed must be open to any changes being made in coding.

Four main maintenance activities are:-

1) Adaptive Maintenance
2) Preventive Maintenance
3) Perfective Maintenance
4) Corrective Maintenance

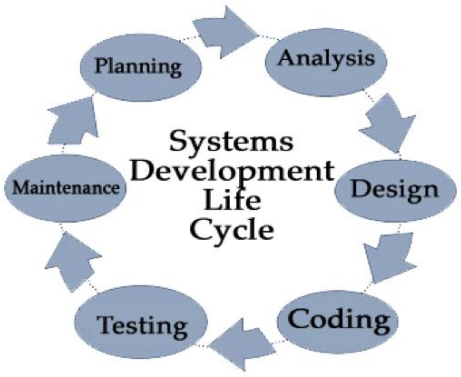



**Fig 2:** System Development life Cycle

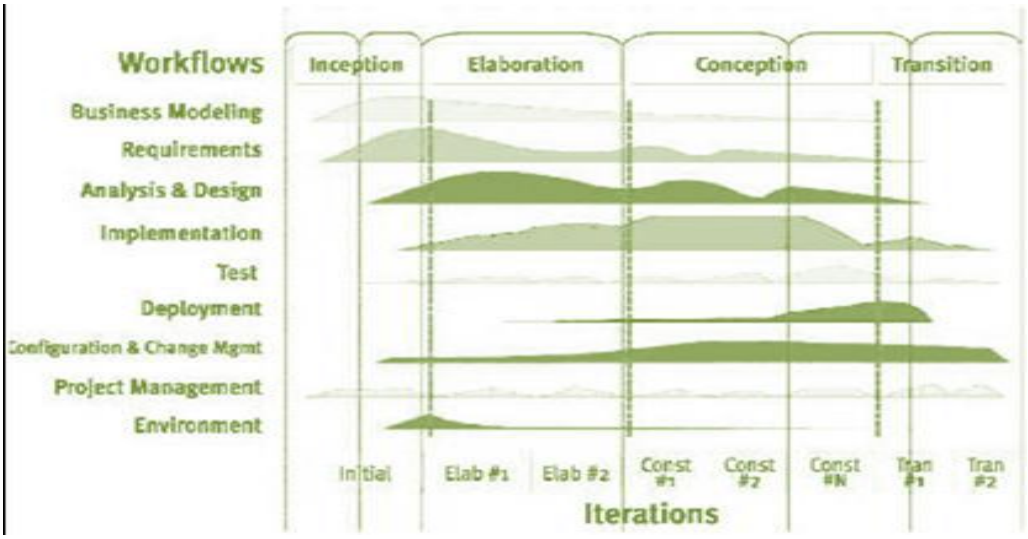
## 3. Scope of Software Testing

### 3.1 Inception Phase
A test engineer understands the need of project in this phase.

### 3.2 Elaboration Phase
In this phase, any tester tries to understand how the project is being developed. Requirements are easily made.

### 3.3 Construction Phase
Developers play an important role and they help developing design of the software. Tester has to know that all requirements are being traced through test cases. System testing and integration testing is necessary in this phase.

### 3.4 Transition Phase
In this phase if any defects or bugs are found then they are re tested and it goes under regression testing phase. With regression testing, reliable products come out. With help of testing in SDLC, any basic product is transformed in to a strong and reliable product [4].




## 4. Types of Testing In SDLC
Each software development life cycle has passed through aset of common phases one or more times. So starting activities early means we can catch small problems before they become big problems later on. Starting testing activities early also provides the chance to review requirements for important quality attributes, to ask questions and to resolve issues. There are three different testing phases in SDLC are [12]:

i) Test Analysis: tester tries to understand about the project.
ii) Test Design: tester design the test cases based on user requirement.
iii) Test Execution: tester execute the test cases and raise defects, if any.

Various types of testing involves throughout SDLC are [1, 12]:

1. **Acceptance Testing:** Formal testing with respect to user needs, requirements, and business processes conducted to determine the acceptability of the system.
2. **Ad-Hoc Testing:** It is performed without planning or

documentation and its main work is to find errors that are not uncovered by other types of testing.

3. **Alpha and Beta Testing:** Alpha testing is the testing done by test teams at development site after the acceptance testing. Beta testing carried out by real users in real environment.
4. **Black Box Testing:** Black box testing is the testing technique whereby the internal workings of the item being tested are not known by the tester.
5. **White Box Testing:** White box testing is the testing of a software solution's internal coding and infrastructure.
6. **Automated Testing:** Using automation tools to write and execute test cases is known as automation testing.
7. **Grey Box Testing:** Grey box testing is a software testing technique that uses a combination of black box testing and white box testing**.**
8. **Integration Testing**: In integration testing the individual tested units are grouped as one and the interface between them is tested.
9. **Regression Testing:** Regression testing means rerunning test cases from existing test suites to build confidence that software changes have no unintended side-effects.
10. **Stress Testing:** Stress testing is a software testing activity that determines the robustness of software by testing beyond the limits of normal operation.
11. **UAT (User Acceptance testing):** It is performed by the end users of the software**.** This testing happens in the final phase of testing.
12. **Security Testing:** Security testing tests the ability of the software to prevent unauthorized access to the resources and data.
13. **Performance Testing:** The goal of application performance testing is to appraise any user experience in realistic scenarios on our target application. In software engineering, performance testing is in general testing performed to determine how a system performs in terms of responsiveness and stability under every condition.

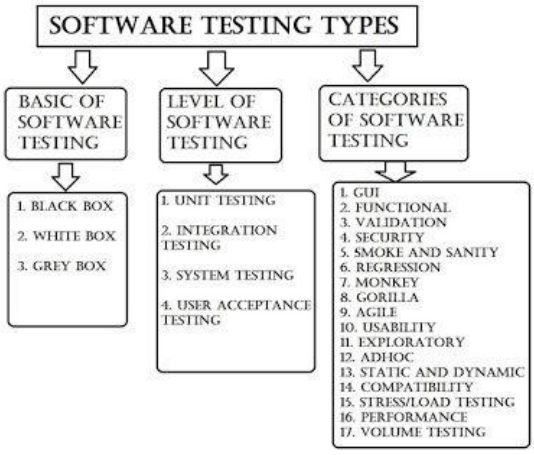



**Fig 3:** Types of Software Testing

## 5. Importance of Testing in SDLC

These are necessary major things in SDLC phase.

### 5.1 Identification of Bugs and Defects

Testing phase is one phase that determines the bugs and errors in the application. These bugs can be at unit level or system level. With so many testing phases, you get to keep away from any kind of bugs that may be affecting your application.

### 5.2 Information to stakeholders and reputation of company

With the help of testing phase, it helps you to know the state of product and service quality. The stakeholders get good information through testing phase about service quality too.

### 5.3 Improvement in Product Quality

As testing is a phase that helps in knowing the actual outcome and the expected outcome. Surely, it can help you to improve your product quality. With proper testing done, you come out of errors and develop perfect software application for the users [5].

### 5.4 Technical Importance

In any SDLC lifecycle, testing phase is important for technical aspect, as it has to come out with best technically sound application.

### 5.5 To Get ahead of any Competitive Developers

Perfect testing tools and methods help you and your application to grow up in market and keep away your competitors. With getting away in all phases of testing, you develop more sound, safe and secure software application [6].

### 5.6 Keep Away any Hazards

When you develop any software, testing plays an important part. When you give your software without any testing then it may be hazardous to the users who are using it. To keep everyone away from any hazard, it is necessary that you go under all testing phases.

### 5.7 Verification and validation

One of the main aims of testing phase in SDLC is for verification and validation. Testing is a phase that can serve as a metrics as it is used heavily in V&V method. Based on the result, you could come out with comparison between qualities amongst different products.

### 5.8 Reliability Estimation

This is yet another important factor that is determined by the testing phase. If your software application has gone through all small level like unit testing and major testing like regression testing and all, then sure it is a reliable application. Hence testing determines reliability of your software application. Testing may serve as the best statistical method that defines operational testing on application resulting in a reliable soft-ware product.

### 5.9 Prove Usability and Operability

One very important aim of software testing is to prove the software is both usable and operable. Usability testing is where the software is released to a select group of users and their working with the product is observed. All aspects of a user's interaction with the software, like ease of use and where users are facing problems, are recoded and analyzed.

### 5.10 Prevent Defect Migration

The majority of errors are usually introduced in the software requirements gathering phase. If the errors are detected early, they can be prevented from migrating to the subsequent

development phase. Early detection and debugging of errors leads to huge savings in software development costs [7].

### 5.11 Economic importance

A well-tested software application will have good economic impact. This is because any one would love to go with reliable and trusted application in market.

## 6. Impact on Quality of Software

Proper tested software gives you more confidence of coming up with great software. Secondly, it improves quality of your software application as continuous and all kinds of testing modes have made a secure and safe application that can be used by the end users. Good testing provides measures for all relevant factors. The importance of any particular factor varies from application to application. Any system where human lives are at stake must place extreme emphasis on reliability and integrity [8, 13].

A testable design is a design that can be easily validated, falsified and maintained. Because testing is a rigorous effort and requires significant time and cost, design for testability is also an important design rule for software development [10].

**Table 1:** Typical Software Quality Factors

| Functionality (exterior quality) | Engineering (interior quality) | Adaptability (future quality) |
|---|---|---|
| Correctness | Efficiency | Flexibility |
| Reliability | Testability | Reusability |
| Usability | Documentation | Maintainability |
| Integrity | Structure | None |

## 7. STLC and Its Phases

STLC is software testing life cycle. The different phases of STLC are as follows:

- System Study
- Test planning
- Writing Test Case or scripts
- Review the test case
- Executing test case
- Bug tracking
- Report the defect


**Fig 4:** Testing Life Cycle

### Entry and Exit Criteria in Software Testing

The Entry Criteria is the process that must be present when a system begins, like

- SRS - Software Requirement Specifications
- FRS - Functional Requirement specifications
- Use case
- Test Case
- Test Plan

The Exit Criteria ensures whether testing is completed and the application is ready for release, like

- Test Summary Report
- Metrics
- Defect Analysis Report.

## 8. Conclusion

An effective and efficient testing will provide timely visibility into the quality and readiness of the system and its input artifacts that is necessary to enable effective decision making throughout the SDLC. In my paper, I have explained various phases of SDLC and importance of testing in each and every phase.

Testing is a very large undertaking. It is very important to have an effective process in place prior to embarking on any major software development effort. This is essential to the success of the overall effort, and delivering quality software to end-users.

## 9. References

1. McConnell Steve. 7: Lifecycle Planning, *Rapid Development*. Red-mond, Washington: Microsoft Press. p. 140.
2. Quick Study: System Development Life Cycle, By Russell Kay, 2002.
3. Selecting a Development Approach. Retrieved, 2008.

4. Introduction to software testing available at http://www.onestopsoftwaretesting.com/introductionand-importance-of-software-testing-in-sdlc/
5. Requirement analysis available at http://en.wikipedia.org/wiki/Requirements analysis
6. Functional specification available at http://en.wikipedia.org/wiki/Functional_specification
7. Software architecture available at http://en.wikipedia.org/wiki/Software_architecture
8. Software construction available at http://en.wikipedia.org/wiki/Software_construction
9. Software design available at http://en.wikipedia.org/wiki/Software_design
10. Software testing available at http://en.wikipedia.org/wiki/Software_testing
11. Debugging available at http://en.wikipedia.org/wiki/Debugging
12. Software deployment available at http://en.wikipedia.org/wiki/Software_deployment
13. Software maintenance available at http://en.wikipedia.org/wiki/Software_maintenance
14. Software development process available at http://en.wikipedia.org/wiki/Software_development_process
15. Involve testing throughout the SDLC available at http://www.silverpath.com