

Review of Software Testing

Nikita Sharma

School of Computer and Electronics, IPS Academy, Indore, M.P., India
Sharmanikita738@gmail.com

Abstract: In general, testing is finding out how well something (software) works. In terms of human beings, testing tells what level of knowledge or skill has been acquired. In computer hardware and software development, testing is used at key checkpoints in the overall process to determine whether objectives are being met. When the design is complete, coding follows and the finished code is then tested. This paper will focus on the main points of testing like why we need testing, Software testing Life Cycle, types of software testing, **How testing is carried out in practical environment and finally at the last this paper will conclude about every point which will discuss in this paper.**



Figure1: Testing

Keyword: SDLC (Software Development Life Cycle), STLC (Software Testing Life Cycle), Project life cycle, testing, static testing, dynamic testing, **Black box testing (behavioral testing), White box testing (Structural or Glass box testing), Unit testing, Incremental integration testing, Integration testing, Functional testing, System testing, End-to-end testing, Sanity testing (confidence testing, smoke testing), Regression testing, Acceptance testing, Load testing, Performance testing, Stress testing, Usability testing, Install/uninstall testing, Recovery testing, Security testing, Compatibility testing, Comparison testing, Alpha testing, Beta testing, SRS (software requirement specification), bug, tool, Test Case.**

Why we need testing:

Testing is the process with the intention to find errors/bugs to analyze the actual results with expected results. The need of testing is as follows:

- * to meet the customer requirements
- * to check the efficiency & effectiveness of the product
- * to produce quality product. [1]

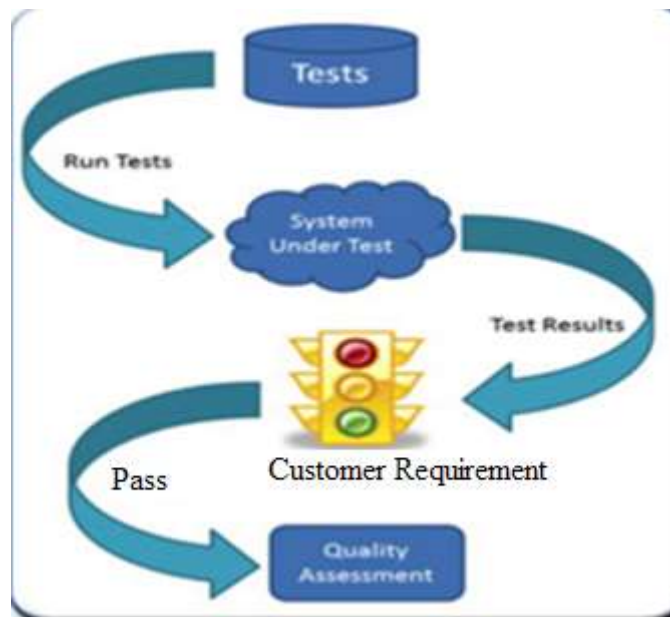


Figure2: We need testing to satisfy customer need.

SOFTWARE TESTING LIFE CYCLE (STLC):

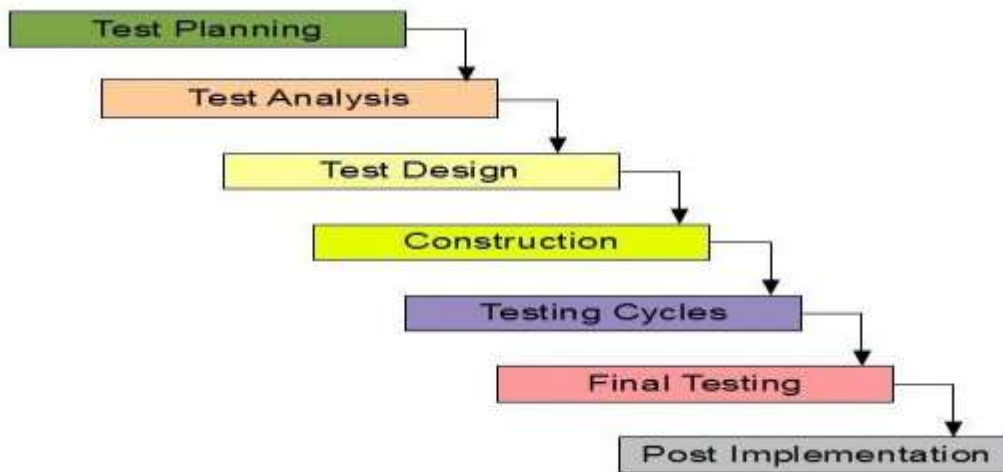


Figure 3: STLC (Software Testing Life Cycle)

- Requirements Analysis
- Test Planning,
- Test Analysis,
- Test Design,
- Construction and verification,
- Testing Cycles
- Final Testing and Implementation and

- Post Implementation.

Software testing has its own life cycle that intersects with every stage of the **SDLC**. The basic requirements in software testing life cycle are to deal with software testing methodologies like Manual, Automated and Performance testing. [2]

1. Requirements Analysis

In this phase Software testers analyze the customer requirements and work with developers during the design phase to see which requirements are testable and how they are going to test those requirements. [2]

2. Test Planning

In this phase all the planning about testing is done like what needs to be tested, how the testing will be done, test strategy to be followed, what will be the test environment, what test methodologies will be followed, hardware and software availability, resources, risks etc.[2]

3. Test Analysis

After test planning phase is over test analysis phase starts, in this phase we need to dig deeper into project and figure out what testing needs to be carried out in each SDLC phase. [2]

4. Test Design

In this phase various black-box and white-box test design techniques are used to design the test cases for testing, testers start writing test cases by following those design techniques, if automation testing needs to be done then automation scripts also needs to be written in this phase.[2]

5. Test Construction and Verification

In this phase testers prepare more test cases by keeping in mind the positive and negative scenarios, end user scenarios etc. All the test cases and automation scripts need to be completed in this phase and got reviewed by the stakeholders. The test plan document should also be finalized and verified by reviewers. [2]

6. Testing Cycles: Test Execution and Bug Reporting

Testing is an iterative process i.e. If defect is found and fixed, testing needs to be done after every defect fix. After tester assures that defects have been fixed and no more critical defects remain in software the build is given for final testing. [2]

7. Final Testing and Implementation

In this phase the final testing is done for the software, non functional testing like stress, load and performance testing are performed in this phase. The software is also verified in the production kind of environment. Final test execution reports and documents are prepared in this phase. [2]

8. Post Implementation

In this phase the test environment is cleaned up and restored to default state, the process review meetings are done and lessons learned are documented. A document is prepared to cope up similar problems in future releases. [2]

Software Testing Types:

Software testing is a method of assessing the functionality of a [software program](#). There are many different types of software testing but the two main categories are [dynamic testing](#) and testing. Dynamic testing is an assessment that is conducted while the program is [executed](#); static testing, on the other hand, is an examination of the program's [code](#) and associated documentation. Dynamic and static methods are often used together. [3]. Main types of testing are discussed below.

Black box testing – Internal system design is not considered in this type of testing. Tests are based on requirements and functionality. The term '**behavioral testing**' is also used for black box testing. Behavioral test design is slightly different from black-box test design because the use of internal knowledge isn't strictly forbidden, but it's still discouraged. [4]

White box testing – This testing is based on knowledge of the internal logic of an application's code. Also known as Glass box Testing. Internal software and code working should be known for this type of testing. Tests are based on coverage of code statements, branches, paths, conditions. White box testing (WBT) is also called **Structural or Glass box testing**. [4]

Unit testing – Testing of individual software components or modules. Typically done by the programmer and not by testers, as it requires detailed knowledge of the internal program design and code. may require developing test driver modules or test harnesses. [4]

Incremental integration testing – Bottom up approach for testing i.e. continuous testing of an application as new functionality is added; Application functionality and modules should be independent enough to test separately. Done by programmers or by testers. [4]

Integration testing – Testing of integrated modules to verify combined functionality after integration. Modules are typically code modules, individual applications, client and server applications on a network, etc. This type of testing is especially relevant to client/server and distributed systems. [4]

Functional testing – This type of testing ignores the internal parts and focus on the output is as per requirement or not. Black-box type testing geared to functional requirements of an application. [4]

System testing – Entire system is tested as per the requirements. Black-box type testing that is based on overall requirements specifications, covers all combined parts of a system. [4]

End-to-end testing – Similar to system testing, involves testing of a complete application environment in a situation that mimics real-world use, such as interacting with a database, using network communications, or interacting with other hardware, applications, or systems if appropriate. [4]

Sanity testing - Testing to determine if a new software version is performing well enough to accept it for a major testing effort. If application is crashing for initial use then system is not stable enough for further testing and build or application is assigned to fix. (Also **confidence tests, smoke testing**). [4]

Regression testing – Testing the application as a whole for the modification in any module or functionality. Difficult to cover all the system in regression testing so typically automation tools are used for these testing types. [4]

Acceptance testing - Normally this type of testing is done to verify if system meets the customer specified requirements. User or customer does this testing to determine whether to accept application. [4]

Load testing – It's a performance testing to check system behavior under load. Testing an application under heavy loads, such as testing of a web site under a range of loads to determine at what point the system's response time degrades or fails. [4]

Stress testing – System is stressed beyond its specifications to check how and when it fails. Performed under heavy load like putting large number beyond storage capacity, complex database queries, continuous input to system or database load. [4]

Performance testing – Term often used interchangeably with ‘stresses and ‘load’ testing. To check whether system meets performance requirements. Used different performance and load tools to do this. [4]

Usability testing – User-friendliness check. Application flow is tested, Can new user understand the application easily, Proper help documented whenever user stuck at any point. Basically system navigation is checked in this testing. [4]

Install/uninstall testing - Tested for full, partial, or upgrade install/uninstall processes on different operating systems under different hardware, software environment. [4]

Recovery testing – Testing how well a system recovers from crashes, hardware failures, or other catastrophic problems. [4]

Security testing – Can system be penetrated by any hacking way? Testing how well the system protects against unauthorized internal or external access. Checked if system, database is safe from external attacks. [4]

Compatibility testing – Testing how well software performs in a particular hardware/software/operating system/network environment and different combinations of above. [4]

Comparison testing – Comparison of product strengths and weaknesses with previous versions or other similar products. [4]

Alpha testing – In house virtual user environment can be created for this type of testing. Testing is done at the end of development. Still minor design changes may be made as a result of such testing. [4]

Beta testing – Testing typically done by end-users or others. Final testing before releasing application for commercial purpose. [4]

How testing is carried out in practical environment?

Whenever we get any new project there is initial project familiarity meeting. In this meeting we basically discuss on who is client? What is project duration and when is delivery? Who is involved in project i.e. manager, Tech leads, QA leads, developers, testers etc..?

First we should know about the SRS- “The SRS fully describes what the software will do and how it will be expected to perform”. [5]

From the SRS (software requirement specification) project plan is developed. The responsibility of testers is to create project plan from this SRS and. Developers start coding from the design. The project work is divided into different modules and these project modules are distributed among the developers. In meantime testers responsibility is to create test scenario and write [test cases](#) according to assigned modules. When developers finish individual modules, those modules are assigned to testers. Smoke testing is performed on these modules and if they fail this test, modules are reassigned to respective developers for fix. For passed modules manual testing is carried out from the written test cases. If any [bug](#) (A **software bug** is an error, flaw, [failure](#), or [fault](#) in a computer program or [system](#) that causes it to produce an incorrect or unexpected result .[6]) is found that get assigned to module developer and get logged in [bug tracking tool](#) (A **bug tracking system** or **defect tracking system** is a [software application](#) that keeps track of reported [software bugs](#) in software development projects. [7]). on bug fix tester do bug verification and regression testing of all related modules. If bug passes the verification it is marked as verified and marked as closed. Otherwise above mentioned bug cycle gets repeated. [8]

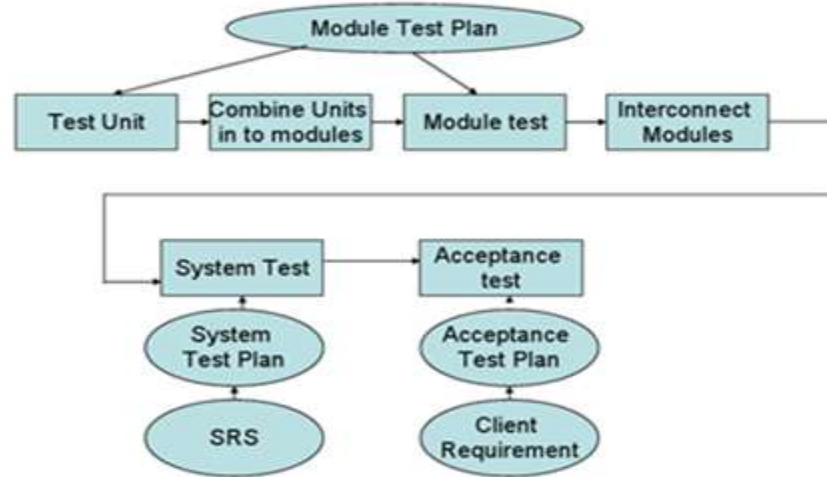


Figure 5: testing process

Different tests are performed on individual modules and integration testing on module integration. These tests include Compatibility testing i.e. testing application on different hardware, OS versions, software platform, different browsers etc. Load and stress testing is also carried out according to SRS. Finally system testing is performed by creating virtual client environment. On passing all the test cases test report is prepared and decision is taken to release the product! So this was a brief outline of process of project life cycle. [8]

Conclusion-

On the basis of overall discussion of software testing in this paper, we can conclude that testing is a method which finds how well a system (software) works, then we discussed about some points based on why we need testing, after this we have different steps of software testing life cycle and different types of software testing, and at last this paper revealed process of software testing in practical environment. Hence we can conclude that software testing is very important process while developing software, and for this we should have deep knowledge about the every point discussed in this paper.

REFERENCES:

- [1] <http://www.allinterview.com/showanswers/57274/basically-why-we-need-testing.html>
- [2] <HTTP://WWW.SOFTWARETESTINGTIPS.COM/SOFTWARE-TESTING-LIFE-CYCLE-STLC/>
- [3] <http://whatis.techtarget.com/definition/software-testing>
- [4] <http://www.softwaretestinghelp.com/types-of-software-testing/>
- [5] <http://searchsoftwarequality.techtarget.com/definition/software-requirements-specification>
- [6] http://en.wikipedia.org/wiki/Software_bug
- [7] http://en.wikipedia.org/wiki/Bug_tracking_system
- [8] <http://www.softwaretestinghelp.com/what-is-actual-testing-process-in-practical-or-company-environment/>