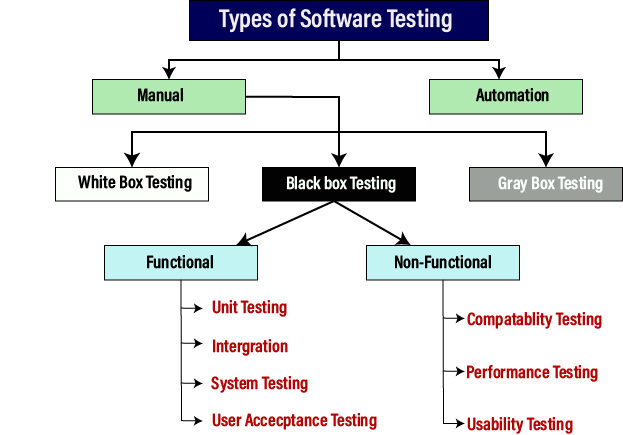
## What is Testing

* Testing is a group of techniques to determine the correctness of the application under the predefined script but, testing cannot find all the defect of application.
* The main intent of testing is to detect failures of the application so that failures can be discovered and corrected. It does not demonstrate that a product functions properly under all conditions but only that it is not working in some specific conditions.
* Testing includes an examination of code and also the execution of code in various environments, conditions as well as all the examining aspects of the code.
* In the current scenario of software development, a testing team may be separate from the development team so that Information derived from testing can be used to correct the process of software development.



Traceability Matrix

Traceability matrix is a table type document that is used in the development of software application to trace requirements. It can be used for both forward (from Requirements to Design or Coding) and backward (from Coding to Requirements) tracing. It is also known as **Requirement Traceability Matrix (RTM) or Cross Reference Matrix (CRM).**

It is prepared before the test execution process to make sure that every requirement is covered in the form of a Test case so that we don't miss out any testing. In the RTM document, we map all the requirements and corresponding test cases to ensure that we have written all the test cases for each condition.

**The test engineer** will prepare RTM for their respective assign modules, and then it will be sent to the Test Lead. The Test Lead will go repository to check whether the Test Case is there or not and finally Test Lead consolidate and prepare one necessary RTM document.

### Manual testing

The process of checking the functionality of an application as per the customer needs without taking any help of automation tools is known as manual testing. While performing the manual testing on any application, we do not need any specific knowledge of any testing tool, rather than have a proper understanding of the product so we can easily prepare the test document.

Manual testing can be further divided into three types of testing, which are as follows:

* **White box testing**
* **Black box testing**
* **Gray box testing**

### Automation testing

Automation testing is a process of converting any manual test cases into the test scripts with the help of automation tools, or any programming language is known as automation testing. With the help of automation testing, we can enhance the speed of our test execution because here, we do not require any human efforts. We need to write a test script and execute those scripts.

Software Development Life Cycle (SDLC)

SDLC is a process that creates a structure of development of software. There are different phases within SDLC, and each phase has its various activities. It makes the development team able to design, create, and deliver a high-quality product.

SDLC describes various phases of software development and the order of execution of phases. Each phase requires deliverable from the previous phase in a life cycle of software development. Requirements are translated into design, design into development and development into testing; after testing, it is given to the client.

Software Development Life Cycle

## Software Development Life Cycle (SDLC) Models

The software development models are those several process or approaches which are being selected for the development of project based on the project's objectives. To accomplish various purposes, we have many development life cycle models. And these models identify the multiple phases of the process. Picking up the correct model for developing the software application is very important because it will explain the what, where, and when of our planned testing.

* **Waterfall model**
* **Spiral model**
* **Verification and validation model**
* **Prototype model**
* **Hybrid model**

### Waterfall Model

It is the first sequential-linear model because the output of the one stage is the input of the next stage. It is simple and easy to understand, which is used for a small project. The various phases of the waterfall model are as follows:

* **Requirement analysis**
* **Feasibility study**
* **Design**
* **Coding**
* **Testing**
* **Installation**
* **Maintenance**

# Software Testing Life Cycle (STLC)

The procedure of software testing is also known as STLC (Software Testing Life Cycle) which includes phases of the testing process.The testing process is executed in a well-planned and systematic manner. All activities are done to improve the quality of the software product.

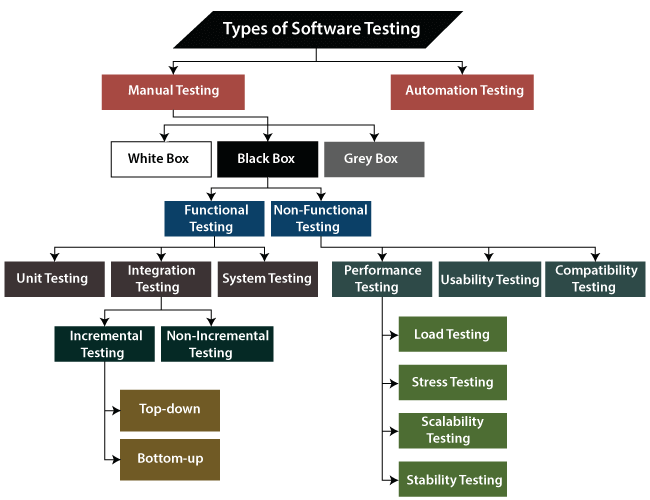
Let's see, the different steps of STLC.

**Software testing life cycle contains the following steps:**

1. [Requirement Analysis](https://www.javatpoint.com/software-testing-life-cycle#requirement-analysis)
2. [Test Plan Creation](https://www.javatpoint.com/software-testing-life-cycle#test-plan-creation)
3. [Environment setup](https://www.javatpoint.com/software-testing-life-cycle#environment-setup)
4. [Test case Execution](https://www.javatpoint.com/software-testing-life-cycle#test-case-execution)
5. [Defect Logging](https://www.javatpoint.com/software-testing-life-cycle#defect-logging)
6. [Test Cycle Closure](https://www.javatpoint.com/software-testing-life-cycle#test-cycle-closure)

## Requirement Analysis:

The first step of the manual testing procedure is requirement analysis. In this phase, tester analyses requirement document of SDLC (Software Development Life Cycle) to examine requirements stated by the client. After examining the requirements, the tester makes a test plan to check whether the software is meeting the requirements or not.



The software testing mainly divided into two parts, which are as follows:

* **Manual Testing**
* **Automation Testing**

## What is Manual Testing?

Testing any software or an application according to the client's needs without using any automation tool is known as **manual testing**.

In other words, we can say that it is a procedure of **verification and validation**. Manual testing is used to verify the behavior of an application or software in contradiction of requirements specification.

We do not require any precise knowledge of any testing tool to execute the manual test cases. We can easily prepare the test document while performing manual testing on any application.

## Classification of Manual Testing

In software testing, manual testing can be further classified into **three different types of testing**, which are as follows:

* **White Box Testing**
* **Black Box Testing**
* **Grey Box Testing**

White box testing is also known as **open box testing, glass box testing, structural testing, clear box testing, and transparent box testing**

Black box testing is a technique of software testing which examines the functionality of software without peering into its internal structure or coding. The primary source of black box testing is a specification of requirements that is stated by the customer.

In other words, we can say that black box testing is a process of checking the functionality of an application as per the customer requirement. The source code is not visible in this testing; that's why it is known as **black-box testing**.

Black box testing further categorizes into two parts, which are as discussed below:

* **Functional Testing**
* **Non-function Testing**

## Automation Testing

The most significant part of Software testing is Automation testing. It uses specific tools to automate manual design test cases without any human interference.

Automation testing is the best way to enhance the efficiency, productivity, and coverage of Software testing.

It is used to re-run the test scenarios, which were executed manually, quickly, and repeatedly.

In other words, we can say that whenever we are testing an application by using some tools is known as **automation testing**.

In software testing, we also have some other types of testing that are not part of any above discussed testing, but those testing are required while testing any software or an application.

* **Smoke Testing**
* **Sanity Testing**
* **Regression Testing**
* **User Acceptance Testing**
* **Exploratory Testing**
* **Adhoc Testing**
* **Security Testing**
* **Globalization Testing**

### Different Levels of Testing

The levels of software testing involve the different methodologies, which can be used while we are performing the software testing.

In [software testing](https://www.javatpoint.com/software-testing-tutorial), we have four different levels of testing, which are as discussed below:

1. **Unit Testing**
2. **Integration Testing**
3. **System Testing**
4. **Acceptance Testing**

### Level1: Unit Testing

**Unit testing** is the first level of software testing, which is used to test if software modules are satisfying the given requirement or not.

The first level of testing involves **analyzing each unit or an individual component** of the software application.

Unit testing is also the first level of [**functional testing**](https://www.javatpoint.com/functional-testing). The primary purpose of executing unit testing is to validate unit components with their performance.

A unit component is an individual function or regulation of the application, or we can say that it is the smallest testable part of the software. The reason of performing the unit testing is to test the correctness of inaccessible code

### Level2: Integration Testing

The second level of software testing is the **integration testing.** The integration testing process comes after **unit testing**.

It is mainly used to test the **data flow from one module or component to other modules.**

In integration testing, the **test engineer** tests the units or separate components or modules of the software in a group.

The primary purpose of executing the integration testing is to identify the defects at the interaction between integrated components or units.

When each component or module works separately, we need to check the data flow between the dependent modules, and this process is known as **integration testing**.

### Level3: System Testing

The third level of software testing is **system testing**, which is used to test the software's functional and non-functional requirements.

It is **end-to-end testing** where the testing environment is parallel to the production environment. In the third level of software testing, **we will test the application as a whole system.**

To check the end-to-end flow of an application or the software as a user is known as **System testing**.

In system testing, we will go through all the necessary modules of an application and test if the end features or the end business works fine, and test the product as a complete system.

### Level4: Acceptance Testing

The **last and fourth level** of software testing is **acceptance testing**, which is used to evaluate whether a specification or the requirements are met as per its delivery.

The software has passed through three testing levels (**Unit Testing, Integration Testing, System Testing**). Some minor errors can still be identified when the end-user uses the system in the actual scenario.

In simple words, we can say that Acceptance testing is the **squeezing of all the testing processes that are previously done.**

The acceptance testing is also known as **User acceptance testing (UAT)** and is done by the customer before accepting the final product.

hecks whether the application is working according to given business scenarios and real-time scenarios.

## What is Test Management?

Test management, process of managing the tests. A test management is also performed using tools to manage both types of tests, automated and manual, that have been previously specified by a test procedure.

Test management tools allow automatic generation of the requirement test matrix (RTM), which is an indication of functional coverage of the application under test (SUT).

**Test Management Process** is a procedure of managing the software testing activities from start to the end. The test management process provides planning, controlling, tracking and monitoring facilities throughout the whole project cycle. The process involves several activities like test planning, designing and test execution. It gives an initial plan and discipline to the software testing process.

There are two main Parts of Test Management Process: –

* Planning
  1. Risk Analysis
  2. Test Estimation
  3. Test Planning
  4. Test Organization
* Execution
  1. Test Monitoring and Control
  2. Issue Management
  3. Test Report and Evaluation
* Test Case
* The test case is defined as a group of conditions under which a tester determines whether a software application is working as per the customer's requirements or not. Test case designing includes preconditions, case name, input conditions, and expected result. A test case is a first level action and derived from test scenarios.
* Test case gives detailed information about testing strategy, testing process, preconditions, and expected output. These are executed during the testing process to check whether the software application is performing the task for that it was developed or not.

**What is Risk Analysis**

The probability of any unwanted incident is defined as Risk. In[Software Testing](https://www.edureka.co/blog/what-is-software-testing/), risk analysis is the process of identifying the risks in applications or software that you built and prioritizing them to test. After that, the process of assigning the level of risk is done. The categorization of the risks takes place, hence, the impact of the risk is calculated.

## Quality Assurance

Software quality assurance is (also known as QA) a sequence of tasks to prevent defects and ensure that the techniques, methods, approaches, and processes are designed for a specific application must be implemented correctly. This is an ongoing process within the development of a software system.

The development of units of an application is checked under the quality assurance specifications in the sequence of their development.

Quality assurance test ensures the development of high-quality software because of its main focus on the high-quality processes, good quality management system and periodic conformance audit during the development of software. It is a managerial tool includes planned and systematic activities and documentation to prevent problems related to quality.

The responsibility of quality assurance is not of any specific team, but it is a responsibility of each member of the development team.

## What is quality assurance?

In the software testing industry, quality assurance ensures that your product is high-quality and effective for your customers to use. Put another way, if you’re developing an app or a website, quality assurance checks such things as the functionality, reliability, and design of your product.

## What is quality engineering?

Quality engineering also focuses on the overall quality of your product. It applies a series of strict quality tests as you move through each phase of product development. QE focuses on anything that impacts your product’s quality, including the design, development, and manufacturing proces

## What is Defect Life Cycle?

Defect life cycle, also known as Bug Life cycle is the journey of a defect cycle, which a defect goes through during its lifetime. It varies from organization to organization and also from project to project as it is governed by the software testing process and also depends upon the tools used.

## Defect Life Cycle States:

* **New -**Potential defect that is raised and yet to be validated.
* **Assigned -**Assigned against a development team to address it but not yet resolved.
* **Active -**The Defect is being addressed by the developer and investigation is under progress. At this stage there are two possible outcomes; viz - Deferred or Rejected.
* **Test -**The Defect is fixed and ready for testing.
* **Verified -**The Defect that is retested and the test has been verified by QA.
* **Closed -**The final state of the defect that can be closed after the QA retesting or can be closed if the defect is duplicate or considered as NOT a defect.
* **Reopened -**When the defect is NOT fixed, QA reopens/reactivates the defect.
* **Deferred -**When a defect cannot be addressed in that particular cycle it is deferred to future release.
* **Rejected -**A defect can be rejected for any of the 3 reasons; viz - duplicate defect, NOT a Defect, Non Reproducible.

### **What is Testing Pyramid?**

Testing Pyramid is a framework that can help both developers and QAs create high-quality software. It reduces the time required for developers to identify if a change they introduced breaks the code. It can also be helpful in building a more reliable test suite.

Essentially, the testing pyramid also referred to as the test automation pyramid, lays out the types of tests that should be included in an automated test suite. It also outlines the sequence and frequency of these tests. The whole point is to offer immediate feedback to ensure that code changes do not disrupt existing features.

## What is Verification in Software Testing?

|  |  |
| --- | --- |
| **Verification** | **Validation** |
| * The verifying process includes checking documents, design, code, and program | * It is a dynamic mechanism of testing and validating the actual product |
| * It does **not** involve executing the code | * It always involves executing the code |
| * Verification uses methods like reviews, walkthroughs, inspections, and desk- checking etc. | * It uses methods like Black Box Testing, [White Box Testing](https://www.guru99.com/white-box-testing.html), and non-functional testing |
| * Whether the software conforms to specification is checked | * It checks whether the software meets the requirements and expectations of a customer |
| * It finds bugs early in the development cycle | * It can find bugs that the verification process can not catch |
| * Target is application and software architecture, specification, complete design, high level, and database design etc. | * Target is an actual product |
| * QA team does verification and make sure that the software is as per the requirement in the SRS document. | * With the involvement of testing team validation is executed on software code. |
| * It comes before validation | * It comes after verification |

**Verification in Software Testing** is a process of checking documents, design, code, and program in order to check if the software has been built according to the requirements or not. The main goal of verification process is to ensure quality of software application, design, architecture etc. The verification process involves activities like reviews, walk-throughs and inspection.

## What is Validation in Software Testing?

**Validation in Software Engineering** is a dynamic mechanism of testing and validating if the software product actually meets the exact needs of the customer or not. The process helps to ensure that the software fulfills the desired use in an appropriate environment. The validation process involves activities like unit testing, integration testing, system testing and user acceptance testing

## What is Requirements based Testing?

Requirements-based testing is a testing approach in which test cases, conditions and data are derived from requirements. It includes functional tests and also non-functional attributes such as performance, reliability or usability.

**Explicit Requirements: The Things You Wrote Down**Our first type of requirement is the explicit requirement. This is the simplest type and the easiest to test. Explicit requirements are most commonly found in documents communicated by stakeholders to the development team. They might take the form of an elaborate design specification, a set of acceptance criteria, or a set of wireframes.

**Implicit Requirements: The Things Your Customers Will Expect**Implicit requirements are the second type. These are all the things that users are going to expect that were not captured explicitly. Examples include performance, usability, availability, and security. Users expect that their password will not be stored in plain text; that requirement need not be written down by anyone.

Consider a cloud-based storage product that lets you store your files online. The product gets a new explicit requirement: Users should be able to share private content to other users via URL, using a share button. However, while testing it is discovered that by modifying a value in the generated URL, it’s possible for other users to view all of the sharing user’s private content. This violates an implicit requirement that only shared content should be accessible to other users, resulting in a show-stopping bug.

**Latent Requirements: Things That Will Delight Your Customers**Lastly, we have latent requirements. Latent requirements represent behaviors that users do not expect based on their previous experiences but which will make them like the software more.

High elevel document

User story – user req

Unit testing

Deliberate

1. What is ut
2. Integration
3. Uat
4. Preproduction testing
5. production
6. Regg resting
7. Performance
8. Black box
9. White box

Local environment

Conncted through vm

Testing concepts

Quality center-tool

Waterfall: the next phase starts after the earlier phase has fully completed

Agile: in the form of sp

The input for testing team is they have

Req Tracebility matrix:

-table format

-mapping btw requirements and testcases

R1 C1 link to the test case doc -> XLS sheet

RTM is been created by your testing engineer /test lead to have a track and mapping between all the requirements and testcases-status

Test strategy:

Test approach – defines how testing should be carried out

Test plan : it is detailed doc which describes the test strategy

Schedule,estimates,deliverables and test data – test plan

Test scenario- grp og test cass

Given a req /usecase there can be many possibilities – test scenario

When test scenario broken up- test case

Test case – emphasizes one particular flow/has steps to execute one single flow. Written in xls file

STLC

Software Testing Life Cycle

Req is confirmed –test planning RTM-test strategy-test plan-test scenarios-test cases/test scripts-test execution-go/no go{defect fixing by developers)

Bug n defect

Bug is mistake in pgn- unit

STLC Phases

Acceptance criteria:

Expected behavior of system as per the requirement

Epic- piece of user history

Entry criteria

Exit criteria

Risk n issue

Leave risk as it is it becomes issue

Smoke testing

Smoke Testing comes into the picture at the time of receiving build software from the development team. The purpose of smoke testing is to determine whether the build software is testable or not. It is done at the time of "building software." This process is also known as "Day 0".

It is a time-saving process. It reduces testing time because testing is done only when the key features of the application are not working or if the key bugs are not fixed. The focus of Smoke Testing is on the workflow of the core and primary functions of the application.

If at all the basic functionalities are working fine

Sanity testing

Whenever a minor change is made we check the entire application if its still working fine

Functional testing

To see if all the functionalities working fine or not

Perfomance testing

How fast the page is rendered . not more than 3 mins(as specified by client)

Regression testing:

Any new code has not impacted the existing functionality

BlackBox testing

You don’t care on internal code you test the functionalities of the application

WhiteBox testing/open box testing/code based testing

Test management

Using a tool u need to manage the testcases and defects

Risk analysis

Stakeholders

Defects life cycle:

New- assign –in progress – test-re test

QA/QE

Quality assurance team:

SQA

QA-> testing role

QE-> best approaches one can use to

He puts in the stringent rules to what tools to be used and policing on the quality regulations so that quality is assured

Test pyramid:

Some model or structure that shd be followed to have a quality product which is made faster and more efficient

Integration /component/modules testing

V&V

Verification & validation testing

User stoires

FRD: Functional Requirement Doc

bRD: Business Req Doc

HLD: High Leve; Design Doc

Waterfall mode

Req testing

Agile

scrum

Requirement Testing, types if it, & static testing

Req testing is based on req provide by the client

All my test cases,test scenarios, test data are inclined from requirements

Functional(based on req) and non functional (performance, system hardware etc) testing

Types of requirement testing:-

-impicit requirements

-explicit reqyuremnts

-latent requirements

Static testing☹done without executing application

Through dummy features:

* 1. Review the code
  2. Static analysis

Alpha testing, Beta testing