**Manual Testing**

## Functional Testing

* **Functional Testing** is a type of software testing that **validates the software system** against the **functional requirements/specifications**.
* The purpose of Functional tests is to **test each function of the software application, by providing appropriate input, verifying the output against the Functional requirements**.
* Functional testing is **executed first**
* Functional testing **describes what the product does**
* **Easy to do Manual Testing**

**Examples of Functional testing are**

* [Unit Testing](https://www.guru99.com/unit-testing-guide.html)
* Smoke Testing
* Sanity Testing
* [Integration Testing](https://www.guru99.com/integration-testing.html)
* White box testing
* Black Box testing
* User Acceptance testing
* [Regression Testing](https://www.guru99.com/regression-testing.html)

**Non-Functional Testing**

* **Non-Functional testing** checks the **Performance, reliability, scalability and other non-functional aspects of the software system.**
* Non-functional testing should be **performed after functional testing**
* Non-functional testing **describes how good the product works.**
* **Tough to do Manual Testing**

**Examples of Non-functional testing are**

* [Performance Testing](https://www.guru99.com/performance-testing.html)
* Load Testing
* Volume Testing
* Stress Testing
* Security Testing
* Installation Testing
* Penetration Testing
* Compatibility Testing
* Migration Testing

**Manual Testing**

* **Manual Testing** is a type of software testing in which **test cases** are **executed manually** by a tester **without using any automated tools**.
* The purpose of Manual Testing is to **identify the bugs, issues, and defects** in the software application.
* Manual software testing is the **most primitive technique** of all testing types and it helps to find **critical bugs in the software application**.

**Black Box testing**

* In [**Black-box testing**](https://www.guru99.com/black-box-testing.html), a tester **doesn’t have any information about the internal working** of the software system.
* Black box testing is a **high level of testing** that focuses on the **behavior of the software**.
* It involves testing from an external or end-user perspective.
* This type of testing is ideal for **higher levels of testing** like **System Testing, Acceptance testing.**
* **Programming knowledge not required.**

**White Box testing**

* [**White-box testing**](https://www.guru99.com/white-box-testing.html) is a testing technique which **checks the internal functioning of the system**.
* In this method, testing is based on **coverage of code statements, branches, paths or conditions**.
* White-Box testing is considered as **low-level testing**.
* It is also called **glass box, transparent box, clear box or code base testing**.
* The white-box Testing method assumes that the **path of the logic in a unit or program is known.**
* Testing is best suited for a **lower level of testing** like **Unit Testing, Integration testing.**
* **Programming knowledge required.**

## Software Testing Hierarchy

## Unit Testing

* **Unit Testing** is a type of software testing where **individual units or components** of a software are tested.
* The purpose is to **validate** that each unit of the **software code performs as expected**.
* **Unit Testing** is done during the **development (coding phase)** of an application by the developers.
* It **isolates a section of code and verify its correctness**.
* A **unit** may be an **individual function, method, procedure, module, or object.**
* In SDLC, STLC, V Model, Unit testing is **first level of testing** **done before integration testing.**
* Unit testing is a **WhiteBox testing technique** that is usually **performed by the developer**.
* It helps to fix bugs early in the development cycle and save costs.

## Integration Testing

* **Integration Testing** is defined as a type of testing where **software modules are integrated logically and tested as a group**.
* A typical software project consists of multiple software modules, coded by different programmers.
* The purpose of this level of testing is to **expose defects in the interaction between these software modules when they are integrated**
* Integration Testing focuses on checking **data communication** amongst these modules. Hence it is also termed as **‘I & T’** **(Integration and Testing),** **‘String Testing’** and sometimes **‘Thread Testing’**.

## System Testing

* **System Testing** is a level of testing that **validates** the complete and fully integrated software product.
* The purpose of a system test is to **evaluate the end-to-end system specifications**.
* Usually, the software is only **one element** of a **larger computer-based system**.Ultimately, the software is **interfaced with other software/hardware systems**.
* System Testing is actually a **series of different tests** whose sole purpose is to **exercise the full computer-based system.**
* System test falls under the **black box testing** category of software testing.

# **Acceptance Testing**

* **Acceptance Testing** is a method of **software testing** where a system is **tested for acceptability.**
* The major aim of this test is to **evaluate the compliance** **of the system** with the **business requirements** and **assess whether it is acceptable for delivery or not**.
* It is a **formal testing** according to **user needs, requirements and business processes** conducted to determine whether a **system satisfies the acceptance criteria or not** and to **enable the users, customers or other authorized entities** to determine whether to accept the system or not.
* Acceptance Testing is the **last phase** of software testing performed after System Testing and before making the system available for actual use.
* **Helps in finding the defects missed during the functional testing phase**.

**Smoke and Sanity Testing**

[**Smoke Testing**](https://www.geeksforgeeks.org/types-software-testing/)

* It is a type of testing which is done **to assure that the acute functionalities of program is working fine.**
* It is also known as **subset of acceptance testing** and it is used to **test all over function of the system/product.**
* This is **documented and scripted.**
* **Performed by either developers or testers.**

[**Sanity Testing**](https://practice.geeksforgeeks.org/problems/what-is-sanity-testing-in-software-testing-process)

* It is done to **check the bugs have been fixed after the build**.
* It is also known as **subset of regression testing**.
* This is **not** **documented and usually not scripted.**
* **Performed by testers.**

**Regression Testing**

* **Regression Testing** is defined as a type of software testing to **confirm that a recent program or code change has not adversely affected existing features**.
* Regression Testing is nothing but a **full or partial selection** of **already executed test cases which are re-executed to ensure existing functionalities work fine.**
* This testing is done to make sure that new code changes **should not have side effects** on the existing functionalities.
* **It ensures that the old code still works once the latest code changes are done.**

**Performance Testing**

* **Performance Testing** is a software testing process used for **testing the speed, response time, stability, reliability, scalability and resource usage of a software application** under particular workload.
* The main purpose of performance testing is to **identify and eliminate the performance bottlenecks in the software application**.
* The focus of Performance Testing is checking a software program’s
* **Speed** – Determines whether the application responds quickly
* **Scalability** – Determines maximum user load the software application can handle.
* **Stability** – Determines if the application is stable under varying loads

**SDET**

* **Software Development Engineer in Test (SDET)** is a developer with the primary responsibility development of **software product** as well as **writing framework** and **tools to test that product**. A **(Software Development Engineer in Test) SDET** takes part in the complete **Software Development Process** mainly we term it as [**Software Development Life Cycle (SDLC)**](https://www.geeksforgeeks.org/software-development-life-cycle-sdlc/). SDETs are able to understand **software development** as well as **software testing**.
* [**Software Development Life Cycle (SDLC)**](https://practice.geeksforgeeks.org/problems/software-development-life-cycle) is a sequence of different activities performed during the **software development process**.
* [**Software Testing Life Cycle (STLC)**](https://www.geeksforgeeks.org/software-testing-life-cycle-stlc/)is a sequence of different activities performed during the **software testing process**.

**RTM**

* **Requirement Traceability Matrix (RTM)** is a document that maps and traces **user requirement with test cases**.
* It captures all requirements proposed by the client and requirement traceability in a **single document**, **delivered at the conclusion of the Software development life cycle**.
* The main purpose of Requirement Traceability Matrix is to validate that **all requirements are checked** via **test cases** such that no functionality is unchecked during Software testing.

**Some of the Parameters included in Requirement Traceability Matrix are :-**

* Requirement ID
* Requirement Type and Description
* Test Cases with Status

### **Test Plan**

* A[**Test Plan**](https://www.guru99.com/what-everybody-ought-to-know-about-test-planing.html)is defined as a document which **outlines the scope, objective, method and weight** on a **software testing task.**
* It contains **schedule, estimates, deliverables and test data.**
* Carried out by the **testing manager.**
* Narrates about the **specification.**
* It **can** change

## Test Strategy

* **Test Strategy** in software testing is defined as a **set of guiding principles** that determines the **test design & regulates how the software testing process will be done.**
* The objective of the[Test Strategy](https://www.guru99.com/how-to-create-test-strategy-document.html)is to provide a **systematic approach to the software testing process** in order to ensure **the quality, traceability, reliability and better planning.**
* Carried out by the **project manager.**
* Narrates about the **general approaches.**
* It **cannot** change

## Test Case

* A **Test Case** is a set of actions executed to **verify a particular feature or functionality of your software application**. A Test Case contains **test steps, test data, precondition, postcondition** developed for **specific test scenario** to verify any requirement.
* The test case includes **specific variables or conditions**, using which a testing engineer can **compare expected and actual results** to determine whether a **software product is functioning as per the requirements of the customer**.
* **What to test and how to test.**
* Requires **more** resources and time.

## Test Scenario

* A **Test Scenario** is defined as **any functionality** that can be tested.
* It is a **collective set of test cases** which helps the testing team to **determine the positive and negative characteristics** of the project.
* Test Scenario gives a **high-level idea** of what we need to test.
* **What to test.**
* Requires **less** resources and time**.**

## Example of Test Scenario :-

For an **eCommerce Application**, a few test scenarios would be

**Test Scenario 1:**Check the Search Functionality

**Test Scenario 2:**Check the Payments Functionality

**Test Scenario 3:**Check the Login Functionality

**Test Script**

* **Test Scripts** are a **line-by-line description** containing the **information about the system** transactions that should be performed to **validate the application or system under test**.
* Test script should **list out each step that should be taken with the expected results.**
* Same as test case but created **programmatically.**

**Test Data**

* **Test Data in Software Testing** is the input given to a **software program** during **test execution.**
* It represents **data** **that affects or affected by** **software execution while testing**.

**Agile software development**

* Agile software development refers to a **group of software development methodologies based on iterative development**, where **requirements and solutions** evolve through **collaboration between self-organizing cross-functional teams.**
* The Agile methodology is **a way to manage a project by breaking it up into several phases**.
* Once the work begins, teams cycle through a process of **planning, executing, and evaluating.**
* **Scrum** is a framework that allows for **more effective collaborations** among teams working on complex projects.
* Agile and scrum are two similar project management systems with a few key differences.
* **Agile is more flexible** and **promotes leadership teams**, while **scrum is more rigid** and **promotes cross-functional teams**.

**Agile scrum methodology** is a[**project management system**](https://www.businessnewsdaily.com/9977-best-online-project-management-software.html)that relies on **incremental development.** Each iteration consists of **two- to four-week sprints**, where each sprint's goal is to build the **most important features first** and **come out with a potentially deliverable product.**

**Quality Analyst (QA)**

One who ensures/maintains the quality of a product by executing on CodeScience's quality procedures.

**Quality Engineer** **(QE)**

One who automates quality procedures to minimize manual testing efforts.

# **Software Testing Life Cycle (STLC)**

* The **Software Testing Life Cycle (STLC)** is a sequence of **specific actions** performed during the **testing process** to ensure that the **software quality objectives** are met.
* The STLC includes both **verification and validation**.

The **STLC** is a **high-quality** strategy directly associated with and part of the **Software Development Life Cycle (SDLC)**, which in turn is a framework with 6 Phases:

1. **Requirement Analysis**
2. **Test Planning**
3. **Test Case Development**
4. **Test Environment Setup**
5. **Test Execution**
6. **Test Cycle Closure**.

### **Requirement Analysis**

* During this phase, **feature requirements** collected in the SDLC process are **evaluated to identify testable aspects**.
* If necessary, testing teams may need to consult with stakeholders to clarify requirements.
* These requirements can either be **functional or non-functional,** defining **what a feature can do or it’s characteristics** respectively.
* The ability to [**automate testing**](https://www.sealights.io/blog/best-practices-in-automated-regression-testing/) is also evaluated during this phase.
* **Entry Criteria—documented requirements, acceptance criteria, and intended product architecture.**
* **Exit Criteria—approved requirement traceability matrix (RTM) and automation feasibility report.**

## Test Planning

## 

* **Test Planning in STLC** is a phase in which a Senior QA manager **determines the test plan strategy along with efforts and cost estimates for the project**.
* The **resources, test environment, test limitations and the testing schedule** are also determined.
* The **Test Plan** gets prepared and finalized in the same phase.

## Test Case Development

* The **Test Case Development Phase** involves the **creation, verification and rework of test cases & test scripts** after the test plan is ready.
* Initially, the [**Test data**](https://www.guru99.com/software-testing-test-data.html)**is identified then created and reviewed** and **then reworked based on the preconditions.**
* Then the QA team starts the development process of test cases for individual units.

## Test Environment Setup

* **Test Environment Setup** decides the **software and hardware conditions** under which a work product is tested.
* It is one of the critical aspects of the testing process and **can be done in parallel with the Test Case Development Phase**.
* Test team may not be involved in this activity if the development team provides the test environment. **The test team is required to do a readiness check** (smoke testing) of the given environment.

## Test Execution

* **Test Execution Phase** is carried out by the testers in which **testing of the software build is done based on test plans and test cases prepared.**
* The process consists of **test script execution, test script maintenance and bug reporting.**
* If bugs are reported then it is reverted back to **development team** for correction and retesting will be performed.

## Test Cycle Closure

* **Test Cycle Closure** phase is **completion of test execution** which involves several activities like **test completion reporting, collection of test completion matrices and test results.**
* Testing team members meet, discuss and analyze testing artifacts to identify strategies that have to be implemented in future, taking lessons from current test cycle.
* **The idea is to remove process bottlenecks for future test cycles.**

**Risk and Issue**

The key difference is an **“issue”** **already has occurred** and a **“risk” is a potential issue** that **may or may not happen** and **can impact the project positively or negatively**.

## Verification and Validation

## 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**.
* It does ***not* involve** executing the code.

## Validation

* **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.**
* It **involves** executing the code**.**

**User Story**

* A **user story** is an **informal, general explanation** of a **software feature** written from the **perspective of the end user**.
* Its purpose is to **articulate how a software feature will provide value to the customer**.

**Functional Requirements Document (FRD)**

The **Functional Requirements Document (FRD)** is a **formal statement of an application’s functional requirements.**

**Business Requirements Document(BRD)**

It is a formal document that **outlines the goals and expectations an organization hopes** to achieve by partnering with a vendor to complete a specific project.

**High Level Design (HLD)**

**High Level Design** (HLD) is a **general system design** and includes the **description of the System architecture and design.**

**Low-Level Design Document (LLDD)**

**Low-Level Design Document (LLDD)** is to give the **internal logical design of the actual program code**.

## 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.

## Static Testing

* **Static Testing** is a software testing technique which is used to check **defects in software application without executing the code.**
* Static testing is done to **avoid errors at an early stage of development** as it is easier to identify the errors and solve the errors.
* **It also helps finding errors that may not be found by Dynamic Testing.**

.

**Alpha and Beta Testing**

## Alpha Testing

* **Alpha Testing** is a type of **acceptance testing**.
* It is performed to identify all **possible issues and bugs before releasing the final product to the end users.**
* Alpha testing is **carried out by the testers** who are internal employees of the organization.
* The main goal is to **identify the tasks that a typical user might perform and test them.**

## Beta Testing

* **Beta Testing** is **performed by “real users”** of the software application in **“real environment”** and it can be considered as a form of **external**[**User Acceptance Testing**](https://www.guru99.com/user-acceptance-testing.html).
* It is the **final test before shipping** a product to the customers.
* **Direct feedback from customers** is a major advantage of Beta Testing. This testing helps to test products in customer’s environment.
* **Beta version** of the software is **released to a limited number of end-users** of the product to obtain **feedback on the product quality**.
* Beta testing **reduces product failure risks** and provides **increased quality** of the product through customer validation.