**Manual Testing:**

Manual Testing is a type of Software Testing where Testers manually execute test cases without using any automation tools.

 Any new application must be manually tested before its testing can be automated.

Manual Testing does not require knowledge of any testing tool. One of the  Software Testing Fundamental is "**100% Automation is not possible**".

This makes Manual Testing imperative.

**Goal of Manual Testing:**

The key concept of manual testing is to ensure that the application is error free and it is working in conformance to the specified functional requirements.

Test Suites or cases ,are designed during the testing phase and should have 100% test coverage.

It also makes sure that reported defects are fixed by developers and re-testing has been performed by testers on the fixed defects.

Basically, this testing checks the quality of the system and delivers bug-free product to the customer.

|  |  |
| --- | --- |
| **Manual Testing** | **Automated Testing** |
| Manual testing requires human intervention for test execution. | [Automation Testing](https://www.guru99.com/automation-testing.html) is use of tools to execute test cases |
| Manual testing will require skilled labour, long time & will imply high costs. | Automation Testing saves time, cost and manpower. Once recorded, it's easier to run an automated test suite |
| Any type of application can be tested manually, certain testing types like ad-hoc and monkey testing are more suited for manual execution. | Automated testing is recommended only for stable systems and is mostly used for [Regression Testing](https://www.guru99.com/regression-testing.html) |
| Manual testing can be become repetitive and boring. | The boring part of executing same test cases time and again, is handled by automation software in Automation Testing. |

**Conclusion**

Manual testing is an activity where the tester needs to be very patient, creative &  open minded.

They need to think and act with an End User perspective.

### What is Automation Testing?

[Manual Testing](https://www.guru99.com/manual-testing.html) is performed by a human sitting in front of a computer carefully executing the test steps.

Automation Testing means using an automation tool to execute your test case suite.

The automation software can also enter test data into the System Under Test, compare expected and actual results and generate detailed test reports. Test Automation demands considerable investments of money and resources.

Successive development cycles will require execution of same test suite repeatedly. Using a test automation tool, it's possible to record this test suite and re-play it as required.Once the test suite is automated, no human intervention is required.This improved ROI of Test Automation.The goal of Automation is to reduce the number of test cases to be run manually and not to eliminate Manual Testing altogether.

## Why Automated Testing?

Automated software testing is important due to following reasons:

* Manual Testing of all workflows, all fields, all negative scenarios is time and money consuming
* It is difficult to test for multilingual sites manually
* Automation does not require Human intervention. You can run automated test unattended (overnight)
* Automation increases the speed of test execution
* Automation helps increase Test Coverage
* Manual Testing can become boring and hence error-prone.

### What is Unit Testing?

Unit testing of software applications is done during the development (coding) of an application.

The objective of Unit Testing is to isolate a section of code and verify its correctness. In procedural programming a unit may be an individual function or procedure

The goal of Unit Testing is to isolate each part of the program and show that the individual parts are correct. Unit testing is usually performed by the developer.

### What is Integration Testing?

In integration testing, individual software modules are integrated logically and tested as a group.

A typical software project consists of multiple software modules, coded by different programmers.

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'.

### What is System Testing?

System Testing is the testing of a complete and fully integrated software product. Usually software is only one element of a larger computer based system. Ultimately, 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.

Two Category of Software Testing

* Black Box Testing
* White Box Testing

System test falls under the **black box testing** category of software testing.

## What is Smoke Testing?

[Smoke Testing](https://www.guru99.com/smoke-testing.html) is a kind of Software Testing performed after software build to ascertain that the critical functionalities of the program is working fine. It is executed "before" any detailed functional or regression tests are executed on the software build. The purpose is to reject a badly broken application, so that the QA team does not waste time installing and testing the software application.

## What is Sanity Testing?

Sanity testing is a kind of Software Testing performed after receiving a software build, with minor changes in code, or functionality, to ascertain that the bugs have been fixed and no further issues are introduced due to these changes. The goal is to determine that the proposed functionality works roughly as expected. If sanity test fails, the build is rejected to save the time and costs involved in a more rigorous testing.

## Smoke Testing Vs Sanity Testing - Key Differences

|  |  |
| --- | --- |
| **Smoke Testing** | **Sanity Testing** |
| Smoke Testing is performed to ascertain that the critical functionalities of the program is working fine | Sanity Testing is done to check the new functionality / bugs have been fixed |
| The objective of this testing is to verify the "stability" of the system in order to proceed with more rigorous testing | The objective of the testing is to verify the "rationality" of the system in order to proceed with more rigorous testing |
| This testing is performed by the developers or testers | Sanity testing is usually performed by testers |
| Smoke testing is usually documented or scripted | Sanity testing is usually not documented and is unscripted |
| Smoke testing is a subset of Acceptance testing | Sanity testing is a subset of [Regression Testing](https://www.guru99.com/regression-testing.html) |
| Smoke testing exercises the entire system from end to end | Sanity testing exercises only the particular component of the entire system |
| Smoke testing is like General Health Check Up | Sanity Testing is like specialized health check up |

### What is 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 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 old code still works once the new code changes are done.

### What is a Test Scenario?

A Test Scenario is any functionality that can be tested. It is also called **Test Condition or Test Possibility**. As a tester, you may put yourself in the end user’s shoes and figure out the real-world scenarios and use cases of the Application Under Test.

### What is Scenario Testing?

Scenario Testing is a variant of Software Testing where Scenarios are Used for Testing. Scenarios help in an Easier Way of Testing of the more complicated Systems

**How to create a Test Scenario**

As a tester, you can follow these five steps to create Test Scenarios-

* **Step 1**: Read the Requirement Documents like BRS, SRS, FRS, of the System Under Test (SUT).  You could also refer uses cases, books, manual, etc. of the application to be tested.
* **Step 2**: For each requirement, figure out possible users actions and objectives. Determine the technical aspects of the requirement. Ascertain possible scenarios of system abuse and evaluate users with hacker's mindset.
* **Step 3:** After reading the Requirements Document and doing your due Analysis, list out different test scenarios that verify each feature of the software.
* **Step 4:** Once you have listed all possible Test Scenarios, a [Traceability Matrix](https://www.guru99.com/traceability-matrix.html) is created to verify that each & every requirement has a corresponding Test Scenario
* **Step 5:** The scenarios created are reviewed by your supervisor. Later, they are also reviewed by other Stakeholders in the project.

**Tips to Create Test Scenarios**

* Each Test Scenario should be tied to a minimum of one Requirement or User Story as per the Project Methodology.
* Before creating a Test Scenario that verifies multiple Requirements at once, ensure you have a Test Scenario that checks that requirement in isolation.
* Avoid creating overly complicated Test Scenarios spanning multiple Requirements.
* The number of scenarios may be large, and it is expensive to run them all. Based on customer priorities only run selected Test Scenarios

**Test Basis:**

* Test Basis is defined as the source for creation of test Cases. It can be the Application itself or the requirement documents like SRS (Software Requirement Specification), BRS (Business Requirement Specification), etc.

### What is Test Data? Why is it Important?

Test data is actually the input given to a software program. It represents data that affects or is affected by the execution of the specific module. Some data may be used for positive testing, typically to verify that a given set of input to a given function produces an expected result. Other data may be used for negative testing to test the ability of the program to handle unusual, extreme, exceptional, or unexpected input. Poorly designed testing data may not test all possible test scenarios which will hamper the quality of the software.

## What Is State Transition in Testing?

State Transition testing is defined as the testing technique in which changes in input conditions cause's state changes in the Application under Test (AUT).

It is a black box testing technique in which the tester analyzes the behavior of an application under test for different input conditions in a sequence. In this technique, tester provides both positive and negative input test values and record the system behavior.

It is the model on which the system and the tests are based. Any system where you get a different output for the same input, depending on what has happened before, is a finite state system.

### What is a Test Plan?

A test plan is a detailed document that outlines the test strategy, [Testing](https://www.guru99.com/software-testing.html) objectives, resources (manpower, software, hardware) required for testing, test schedule, [Test Estimation](https://www.guru99.com/an-expert-view-on-test-estimation.html) and test deliverables.

The test plan serves as a blueprint to conduct software testing activities as a defined process which is minutely monitored and controlled by the test manager.

## What is the Test Report?

Test Report is a document which contains

* A **summary** of test activities and final test results
* An **assessment** of how well the [Testing](https://www.guru99.com/software-testing.html) is performed

Based on the test report, the stakeholders can

* **Evaluate** the **quality** of the tested product

### What is Bug?

* A bug is the consequence/outcome of a coding fault

### What is Defect?

* A defect is a variation or deviation from the original business requirements