**MANUAL TESTING**

**https://www.javatpoint.com/software-testing-tutorial**

**WHAT IS 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. 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. It also makes sure that reported defects are fixed by developers and re-testing has been performed by testers on the fixed defects. This testing checks the quality of the system and delivers bug-free product to the customer.

* Manual testing is an activity where the tester needs to be very patient, creative & open minded.
* Manual testing is a vital part of user-friendly software development because humans are involved in testing software applications and end-users are also humans. They need to think and act with an End User perspective.
* Manual Testing is one of the most fundamental testing processes as it can find both visible and hidden defects of the software.

## How to perform Manual Testing:

* Analyze requirements from the software requirement specification document.
* Create a clear test plan.
* Write test cases that cover all the requirements defined in the document.
* Get test cases reviewed by the QA lead.
* Execute test cases and detect any bugs.
* Report bugs, if any, and once fixed, run the failed tests again to re-verify the fixes.

**Tools to Automate Manual Testing:**

* [Selenium](https://www.guru99.com/selenium-tutorial.html)
* [QTP](https://www.guru99.com/quick-test-professional-qtp-tutorial.html)
* [J meter](https://www.guru99.com/jmeter-tutorials.html)
* [Load runner](https://www.guru99.com/loadrunner-v12-tutorials.html)
* [Test Link](https://www.guru99.com/testlink-tutorial-complete-guide.html)
* Quality Center (ALM)
* JIRA

**HOW TO GET STRATED WITH TESTING AN SOFTWARE APPLICATION:**

## Test Strategy:

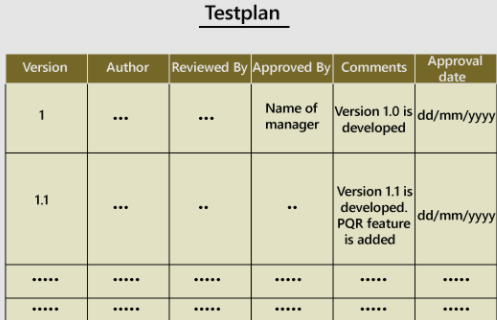
## 

* Your goal is to be as effective as possible a good test strategy lays out the overview of how to be effective.
* which types of testing are to be performed, and which entry and exit criteria apply. They are created based on development design documents.

**Types of testing strategies:** Some of the testing methodologies that may be part of an organization’s testing strategy are:

* [Analytical strategy](http://tryqa.com/what-is-test-strategy-types-of-strategies-with-examples/#analytical_strategy)
* [Model based strategy](http://tryqa.com/what-is-test-strategy-types-of-strategies-with-examples/#model_based_strategy)
* [Methodical strategy](http://tryqa.com/what-is-test-strategy-types-of-strategies-with-examples/#methodical_strategy)
* [Standards compliant or Process compliant strategy](http://tryqa.com/what-is-test-strategy-types-of-strategies-with-examples/#standards_or_process_compliant_strategy)
* [Reactive strategy](http://tryqa.com/what-is-test-strategy-types-of-strategies-with-examples/#reactive_strategy)
* [Consultative strategy](http://tryqa.com/what-is-test-strategy-types-of-strategies-with-examples/#consultative_strategy)
* [Regression averse strategy](http://tryqa.com/what-is-test-strategy-types-of-strategies-with-examples/#regression_averse_strategy)

1. **TESTING PLAN:**



* A test plan is a detailed document which describes software testing areas and activities. It outlines the test strategy, objectives, test schedule, required resources (human resources, software, and hardware), test estimation and test deliverables.
* The test plan is a base of every software's testing. It is the most crucial activity which ensures availability of all the lists of planned activities in an appropriate sequence.
* The test plan is a template for conducting software testing activities as a defined process that is fully monitored and controlled by the testing manager. The test plan is prepared by the Test Lead (60%), Test Manager (20%), and by the test engineer (20%).

**Types of Test Plan:** There are three types of the test plan

* Master Test Plan – Test plan has multiple levels of testing that includes complete test strategy.
* Phase Test Plan – That addresses any one phase of test strategy.
* Testing Type Specific Test Plans – Designed for major types of testing like security testing, load testing, performance testing etc.

**How to write a Test Plan:** Making a test plan is the most crucial task of the test management process First, analyze product structure and architecture.

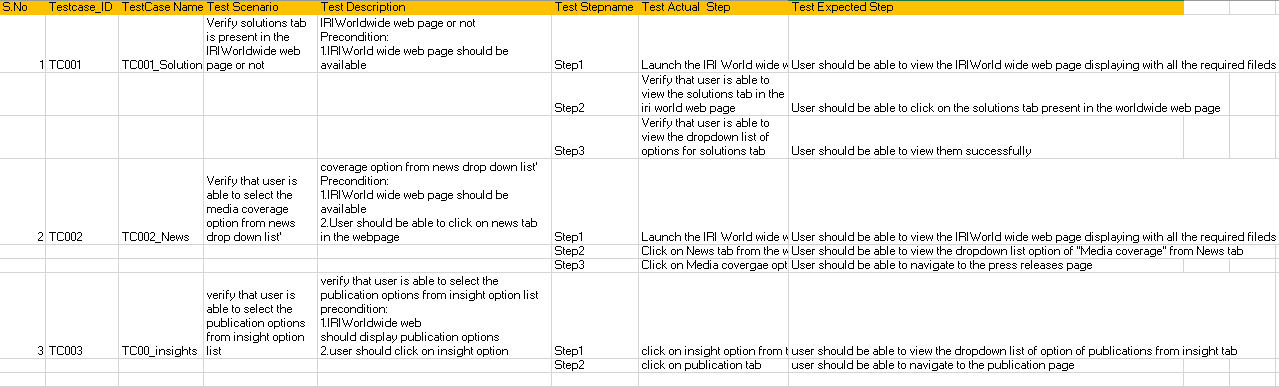
* Now design the test strategy.
* Define all the test objectives.
* Define the testing area.
* Define all the useable resources.
* Schedule all activities in an appropriate manner.
* Determine all the Test Deliverables.

## Test plan components or attributes:

## Test plan

1. **TEST CASE:** A test case is exactly what it sounds like a test scenario measuring functionality across a set of actions or conditions to verify the expected result. They apply to any software application, can use manual testing or an automated test.

* A Test Case contains test steps, test data, precondition, postcondition developed for specific test scenario to verify any requirement.



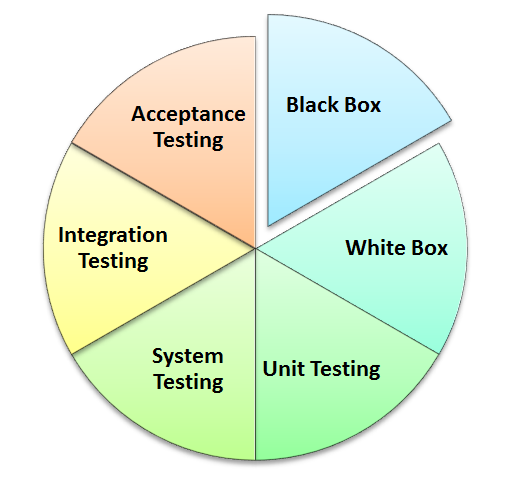
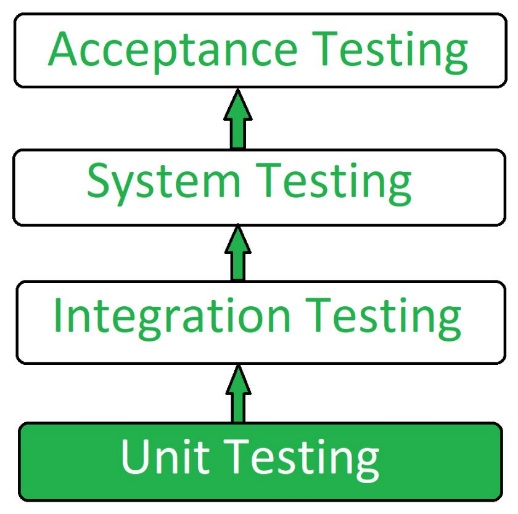
* **TEST SCENARIO**: The test scenario is a detailed document of test cases that cover end to end functionality of a software application in liner statements. In the test scenario, testers need to put themselves in the place of the user because they test the software application under the user's point of view.
* **TEST DESCRIPTION:** A test case is usually a single step, or occasionally a sequence of steps, to test the correct behaviour/functionality, features of an application. An expected result or expected outcome is usually given. Test case Description/Summary - This field describes the test case objective.

1. **TEST DATA: “**Data used for testing purposes”. Data created or selected to satisfy the execution preconditions and inputs to execute one or more test cases.
   * + In your tests, you’ll be creating, updating, and deleting data. You don’t want to delete or update real data from real users from your app, so it’s important to have a set of test data that can be altered to make sure each one of your functions are working.
     + Test data development is done simultaneously with test case development.
2. **TEST ENVIRONMENT:** A test environment is a server that allows you to run the test cases you have defined.

It also involves hardware and network configuration.

The purpose of test environments is to mimic the processes and user experiences of regular usage in the live environment (PROD).

**Types and stages in Manual Testing:** This can be executed both manually and as well as using automated testing tools.

* **UNIT TESTING:** Unit Testing is defined as software testing where individual components of a software are tested. Unit testing is a type of white box testing.

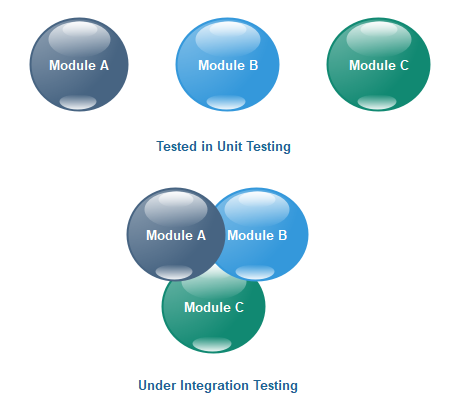
* This is also known as developer testing. developer can also test the application.
* The purpose of unit testing is to test the correctness of isolated code. Whenever the application is ready and given to the Test engineer, he/she will start checking every component of the module or module of the application independently or one by one, and this process is known as **Unit testing or components testing** .unit testing is the first level of testing process.

**Some crucial reasons are listed below:**

* Unit testing helps tester and developers to understand the base of code that makes them able to change defect causing code quickly.
* Unit testing helps in the documentation.
* Unit testing fixes defects very early in the development phase that's why there is a possibility to occur a smaller number of defects in upcoming testing levels.
* It helps with code reusability by migrating code and test cases.
* **INTEGRATION TESTING:**

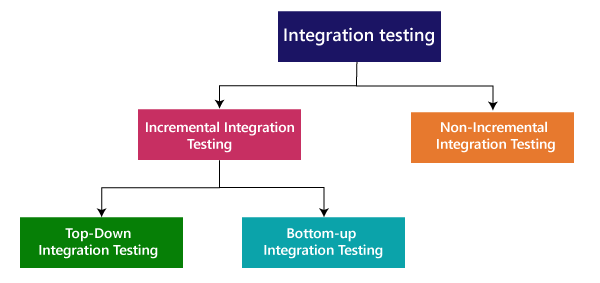
Integration testing is the second level of the software testing process comes after unit testing. In this testing, units or individual components of the software are tested in a group. The focus of the integration testing level is to expose defects at the time of interaction between integrated components or units.

[Unit testing](https://www.javatpoint.com/unit-testing) uses modules for testing purpose, and these modules are combined and tested in integration testing. The Software is developed with a number of software modules that are coded by different coders or programmers. The goal of integration testing is to check the correctness of communication among all the modules.



**TYPES OF INTEGRATION TESTING:**

* Incremental integration testing.
* Non-incremental integration testing.



* **SYSTEM TESTING:**
* System testing is testing conducted on a complete integrated system to evaluate the system's compliance with its specified requirements. System testing takes, as its input, all of the integrated components that have passed integration testing.
* System testing is an end-to-end testing.it verifies the entire software
* At this level, the whole system validates with unit level performance.
* **ACCEPTANCE TESTING:**
* Acceptance Testing is the final level of software testing. The main aim of this testing is to determine the working process of the system by satisfying the required specifications and it is acceptable for delivery. It is also known as End-User Testing. It also works under the Black Box Testing Method.

**Types of acceptance testing include:**

* Alpha & Beta Testing.
* Contract Acceptance Testing.
* Regulation Acceptance Testing.
* Operational Acceptance testing.