Introduction:-

Testing is a process of executing a program with the aim of finding error. To make our software perform well it should be error free.If testing is done successfully it will remove all the errors from the software.

Principles of Testing:-

(i) All the test should meet the customer requirements
(ii) To make our software testing should be performed by third party
(iii) Exhaustive testing is not possible. As we need the optimal amount of testing based on the risk assessment of the application.
(iv) All the test to be conducted should be planned before implementing it
(v) It follows pareto rule(80/20 rule) which states that 80% of errors comes from 20% of program components.
(vi) Start testing with small parts and extend it to large parts.

Types of Testing:-

1. Unit Testing

It focuses on smallest unit of software design. In this we test an individual unit or group of inter related units.It is often done by programmer by using sample input and observing its corresponding outputs.
Example:

a) In a program we are checking if loop, method or function is working fine

b) Misunderstood or incorrect, arithmetic precedence.

c) Incorrect initialization

2. Integration Testing

The objective is to take unit tested components and build a program structure that has been dictated by design.Integration testing is testing in which a group of components are combined to produce output.

Integration testing is of four types: (i) Top down (ii) Bottom up (iii) Sandwich (iv) Big-Bang
Example

(a) Black Box testing:- It is used for validation.
In this we ignore internal working mechanism and focus on **what is the output?**.

(b) White Box testing:- It is used for verification.
In this we focus on internal mechanism i.e.**how the output is achieved?**

3. Regression Testing

Every time new module is added leads to changes in program. This type of testing make sure that whole component works properly even after adding components to the complete program.

Example

In school record suppose we have module staff, students and finance combining these modules and checking if on integration these module works fine is regression testing

4. Smoke Testing

This test is done to make sure that software under testing is ready or stable for further testing
It is called smoke test as testing initial pass is done to check if it did not catch the fire or smoked in the initial switch on.
Example:

If project has 2 modules so before going to module make sure that module 1 works properly

5. Alpha Testing

This is a type of validation testing.It is a type of *acceptance testing* which is done before the product is released to customers. It is typically done by QA people.
Example:

When software testing is performed internally within the organization

6. Beta Testing

The beta test is conducted at one or more customer sites by the end-user of the software. This version is released for the limited number of users for testing in real time environment
Example:

When software testing is performed for the limited number of people

7. System Testing

In this software is tested such that it works fine for different operating system.It is covered under the black box testing technique. In this we just focus on required input and output without focusing on internal working.
In this we have security testing, recovery testing , stress testing and performance testing
Example:

This include functional as well as non functional testing

8. Stress Testing

In this we gives unfavorable conditions to the system and check how they perform in those condition.
Example:

(a) Test cases that require maximum memory or other resources are executed

(b) Test cases that may cause thrashing in a virtual operating system

(c) Test cases that may cause excessive disk requirement

9. Performance Testing

It is designed to test the run-time performance of software within the context of an integrated system. It is used to test speed and effectiveness of program. It is also called load testing. In it we check , what is the performance of the system in the given load.

Example:

Checking number of processor cycles.

Levels of Testing

Testers are following various levels during the time of testing. The primary two levels of testing divided into four levels of testing. These levels of software testing comprise dissimilar methodologies which can be implemented at the time of testing software.

The two significant levels of software testing are:

- Functional Testing
- Non-functional Testing

Functionality Test

This is a kind of black box testing which is specification oriented testing which is done by putting the input and then the output is examined whether it comes the way it is expected. Five steps of testing are there which involved the testing of an application for functionality. They are:

- Determine the functionality which a projected software is meant to perform
- Creation of test data depending on requirements of the application
- Check for output depending on the data being tested along with specifications of your application
- Scripting of scenarios by testing and then executing those test cases
- Compare the actual as well as expected output depending on the executed tests

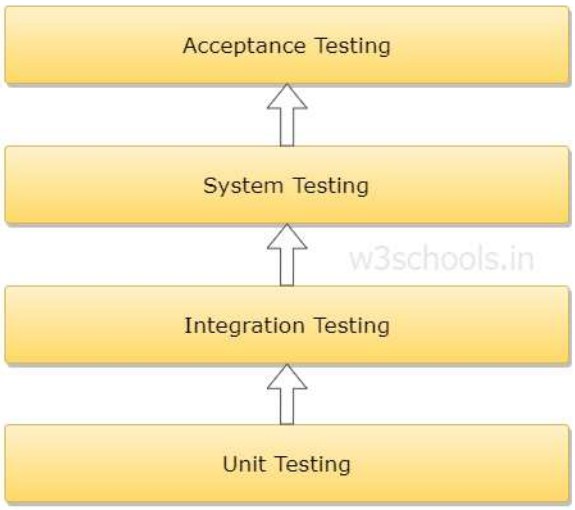
Non-Functional Testing

This type of testing deals with testing any product from the requirement point of view that are nonfunctional in type and important at the same time. Nonfunctional testing can be categorized as:

- Performance
- Security
- User interface
- Levels of Testing

The four Levels of software testing goes from Unit testing towards Integration testing on the following list:

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



Levels of Testing

1. **Unit Testing:** In this testing level, individual sections or parts of software or product to being tested. The idea of this is to confirm every parts or unit of the product after the test.
2. **Integration Testing:** In this software testing level, individual parts need to combine as well as a test as a single cluster. The main idea of this testing level is for exposing the faults while interacting between integrated units of the project.
3. **System Testing:** In this software testing level, the whole, integrated software or project is tested. The principle for this testing is to assess the system's conformity with its intended requirements.
4. **Acceptance Testing:** At this software testing level, a system needs to be tested for adequacy. This test is purposefully done for evaluating the compliance of the system with business its requirements.

The testing sequence

These four types of testing cannot be applied haphazardly during development. There is a logical sequence that should be adhered to in order to minimize the risk of bugs cropping up just before the launch date.

Any testing team should know that testing is important at every phase of the development cycle.
By progressively testing the simpler components of the system and moving on the bigger, more complex groupings, the testers can rest assured they are thoroughly examining the software in the most efficient way possible.

The four levels of testing shouldn't only be seen as a hierarchy that extends from simple to complex, but also as a sequence that spans the whole development process from the early to the later stages. Note however that later does not imply that acceptance testing is done only after say 6 months of development work. In a more agile approach, acceptance testing can be carried out as often as every 2-3 weeks, as a part of the sprint demo. In an organization working more traditionally it is quite typical to have 3-4 releases per year, each following the cycle described here.