What is Software Testing?

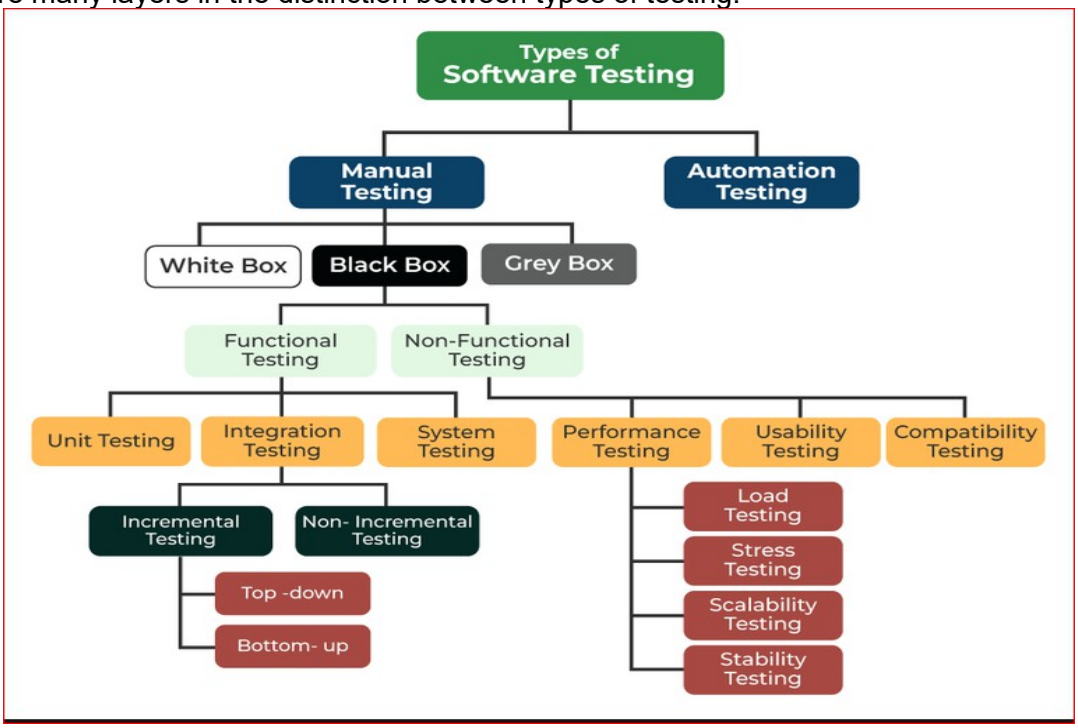
Software Testing is the process of checking the correct functioning of software applications and systems. It is really important in software development and it should be continuous throughout the development process, to ensure every functionality is safe, seamlessly operating and user-friendly. It is beneficial to have a proper test management plan in place, that permits to find out the highest number of bugs or, more generally speaking, flaws, with the fewest numbers of tests.

There are many reasons why testing is fundamental nowadays:

1. Considering the majority of software is developed to be sold, we should think of the main damage of a software with bugs, or an unreliable, unstable or inefficient software as a loss of clients and a bad reputation for the business;
2. Security is a crucial aspect of testing. We want our software to be safe from malicious attacks and data breaches; since we are likely dealing with customer's details, we must ensure these are safely stored and check every possible system vulnerability;
3. We also need to make sure the software is compliant with regulations and standards to avoid legal and financial consequences;
4. Testing also helps in optimizing the performance of our software. This, apart from boosting reputation, increases interest in the product, as it meets customer expectations and enhances their satisfaction, leading to increased sales;
5. Finally, an efficient testing plan, can cut costs during the production and development process, giving developers the chance to fix the potential bugs earlier, saving resources and time.

Types Of Software Testing:

There are many layers in the distinction between types of testing.



Functional and Non-Functional testing are both types of Black-Box testing, a subset of Manual Testing.

Functional Testing is result-oriented, it focuses on the functionality and the requirements of the system, without making assumptions about its internal structures.

The three main examples of functional testing are:
Unit Testing: small units of code, like a function or a method, are tested in isolation;
Integration Testing: it verifies that different units of software correctly interact with each other;
System Testing: evaluation of the complete product, it tests the functionality, requirements and the suitability for delivery.

Non-Functional Testing aims at verifying if a system is performing properly and is reliable under load.

There are three types of Non-Functional Testing:
Performance Testing: evaluates the operational efficiency and scalability of a system, focusing on optimization;
Usability Testing: simulates user interactions to ensure that specific tasks can be completed as intended, focusing on the user experience.
Compatibility Testing: tests the software on different platforms and within different environments to ensure this works across various systems.

-