* **Software Testing**

Software Testing is evaluation of the software against requirements gathered from users and system specifications. Testing is conducted at the phase level in software development life cycle or at module level in program code. Software testing comprises of Validation and Verification.

**Validation:**

Validation is process of examining whether or not the software satisfies the user requirements. It is carried out at the end of the SDLC. If the software matches requirements for which it was made, it is validated.

* Validation ensures the product under development is as per the user requirements.
* Validation answers the question – "Are we developing the product which attempts all that user needs from this software ?".
* Validation emphasizes on user requirements.

**Verification:**

Verification is the process of confirming if the software is meeting the business requirements, and is developed adhering to the proper specifications and methodologies.

* Verification ensures the product being developed is according to design specifications.
* Verification answers the question– "Are we developing this product by firmly following all design specifications ?"
* Verifications concentrates on the design and system specifications.

**Classification of Software Testing:**

Software Testing can be broadly classified into two types:

* **Manual:**

This testing is performed without taking help of automated testing tools. The software tester prepares test cases for different sections and levels of the code, executes the tests and reports the result to the manager.

Manual testing is time and resource consuming. The tester needs to confirm whether or not right test cases are used. Major portion of testing involves manual testing.

* **Automated:**

This testing is a testing procedure done with aid of automated testing tools. The limitations with manual testing can be overcome using automated test tools.

**Testing approaches:**

Tests can be conducted based on two approaches –

1. **Functionality testing/Black-box testing:**

* The technique of testing in which the tester doesn’t have access to the source code of the software and is conducted at the software interface without any concern with the internal logical structure of the software is known as blackbox testing.



* It is carried out to test functionality of the program. It is also called ‘Behavioral’ testing.
* In this testing method, the design and structure of the code are not known to the tester, and testing engineers and end users conduct this test on the software.

Black-box testing techniques:

1. **Equivalence class :**

The input is divided into similar classes. If one element of a class passes the test, it is assumed that all the class is passed.

1. **Boundary values :**

The input is divided into higher and lower end values. If these values pass the test, it is assumed that all values in between may pass too.

1. **Cause-effect graphing:**

In both previous methods, only one input value at a time is tested. Cause (input) – Effect (output) is a testing technique where combinations of input values are tested in a systematic way.

1. **Pair-wise Testing :**

The behavior of software depends on multiple parameters. In pairwise testing, the multiple parameters are tested pair-wise for their different values.

1. **State-based testing :**

The system changes state on provision of input. These systems are tested based on their states and input.

1. **Implementation testing/White-box testing:**

* The technique of testing in which the tester is aware of the internal workings of the product, has access to its source code, and is conducted by making sure that all internal operations are performed according to the specifications is known as white box testing.



* It is conducted to test program and its implementation, in order to improve code efficiency or structure. It is also known as ‘Structural’ testing.
* In this testing method, the design and structure of the code are known to the tester. Programmers of the code conduct this test on the code.

White-box testing techniques:

1. **Control-flow testing -** The purpose of the control-flow testing to set up test cases which covers all statements and branch conditions. The branch conditions are tested for both being true and false, so that all statements can be covered.
2. **Data-flow testing -** This testing technique emphasis to cover all the data variables included in the program. It tests where the variables were declared and defined and where they were used or changed.