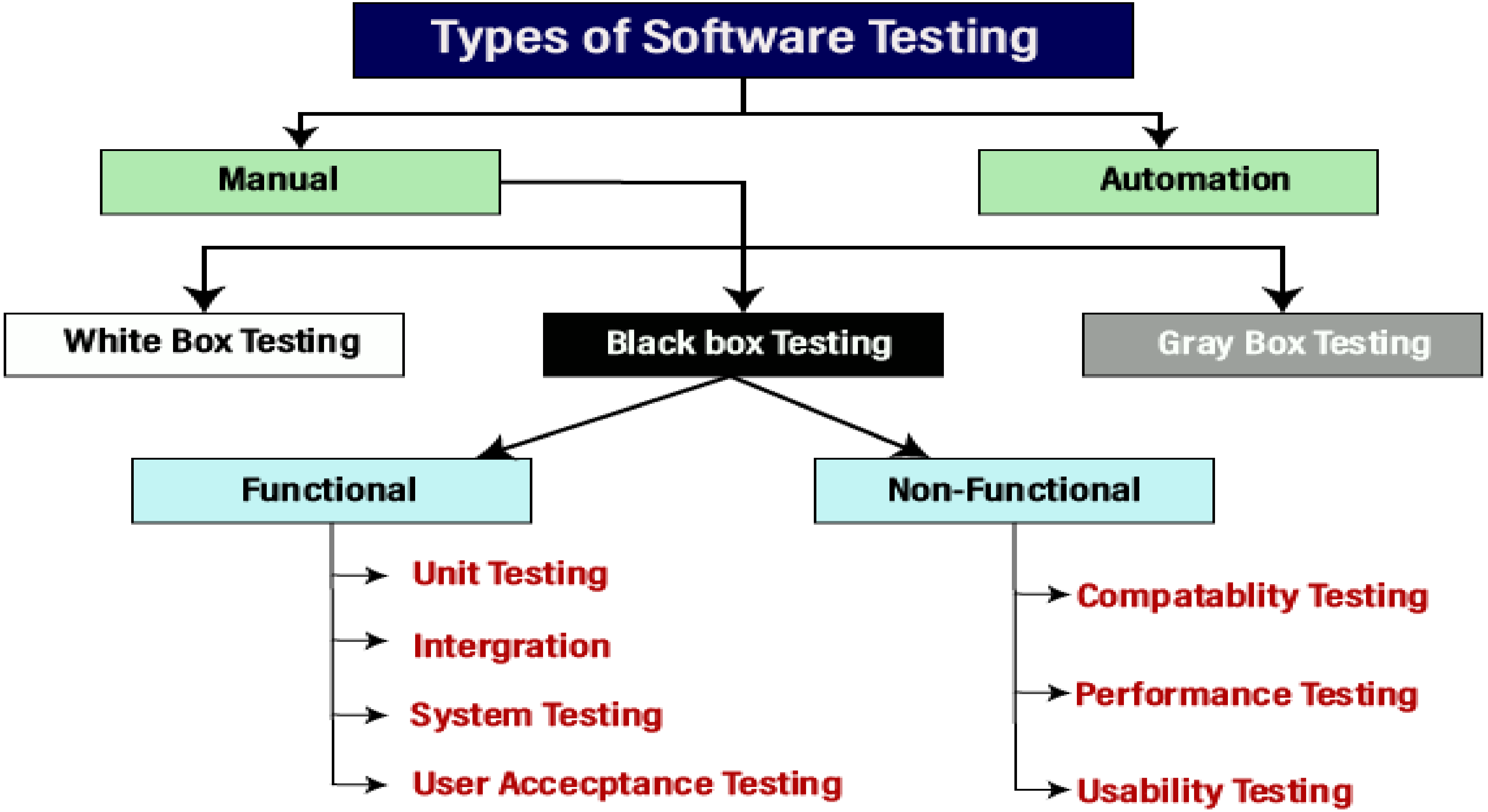
# Software Testing

**Rakibul Hassan**
**Lecturer**
**Dept. of ECE, RUET**

# Software Testing

**Software Testing** is a method to check whether the actual software product matches expected requirements and to ensure that software product is Defect free. It involves execution of software/system components using manual or automated tools to evaluate one or more properties of interest. The purpose of software testing is to identify errors, gaps or missing requirements in contrast to actual requirements.

# Types of Software Testing

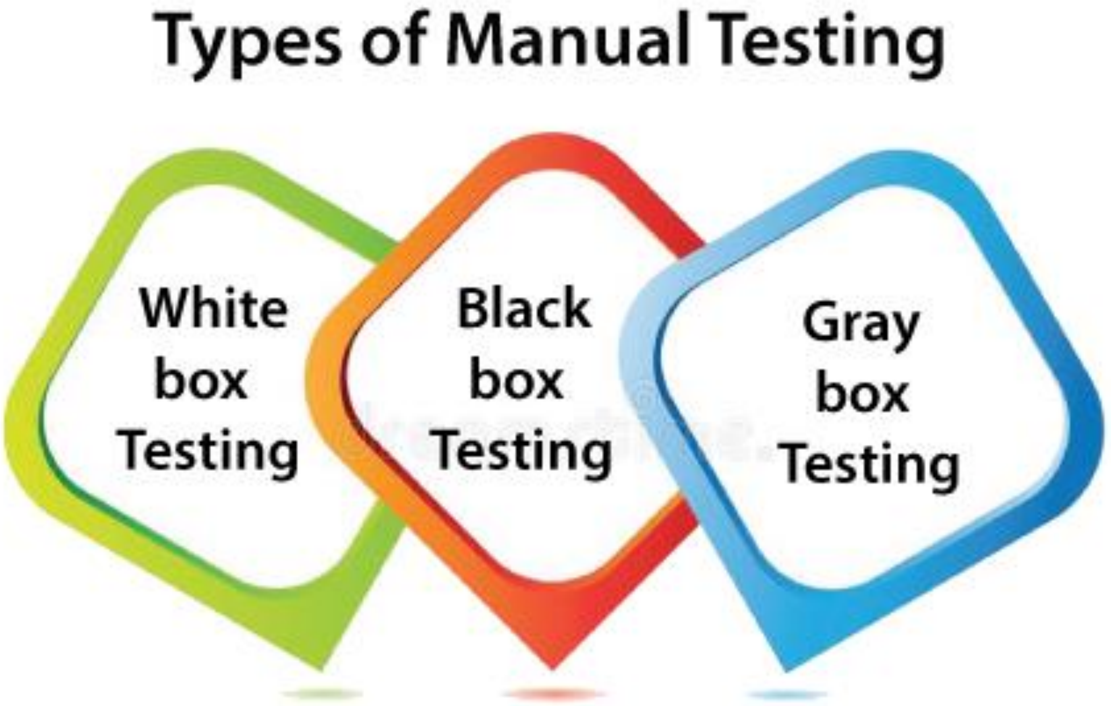


# Types of Software Testing

**Manual testing**: Manual testing is a software testing process in which test cases are executed manually without using any automated tool. All test cases executed by the tester manually according to the end user's perspective. It ensures whether the application is working, as mentioned in the requirement document or not. Test cases are planned and implemented to complete almost 100 percent of the software application. Test case reports are also generated manually.

# Types of Software Testing

**Manual testing** is mandatory for every newly developed software before automated testing. This testing requires great efforts and time, but it gives the surety of bug-free software.



# Types of Software Testing

## White-box testing

The white box testing is done by Developer, where they check every line of a code before giving it to the Test Engineer. Since the code is visible for the Developer during the testing, that's why it is also known as White box testing.

# Types of Software Testing

White-box testing involves

- Path testing
- Loop testing
- Condition testing
- Testing based on the memory perspective
- Test performance of the program

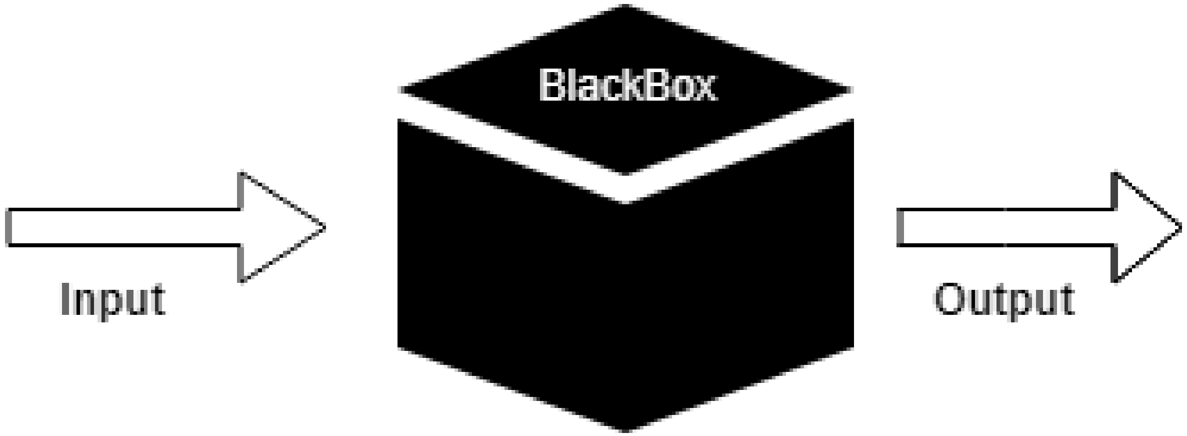
.

# Types of Software Testing

## Black box testing

The black box testing is done by the Test Engineer, where they can check the functionality of an application or the software according to the customer /client's needs. In this, the code is not visible while performing the testing; that's why it is known as black-box testing.

.



# Types of Software Testing

## Black box testing

## Generic steps of black box testing

- The black box test is based on the specification of requirements, so it is examined in the beginning.

- In the second step, the tester creates a positive test scenario and an adverse test scenario by selecting valid and invalid input values to check that the software is processing them correctly or incorrectly.

- In the third step, the tester develops various test cases such as decision table, all pairs test, equivalent division, error estimation, cause-effect graph, etc.

# Types of Software Testing

## Black box testing

### Generic steps of black box testing

- The fourth phase includes the execution of all test cases.

- In the fifth step, the tester compares the expected output against the actual output.

- In the sixth and final step, if there is any flaw in the software, then it is cured and tested again

# Types of Software Testing

## Gray Box testing

Gray box testing is a combination of white box and Black box testing. It can be performed by a person who knew both coding and testing. And if the single person performs white box, as well as black-box testing for the application, is known as Gray box testing.

# Types of Software Testing

**Automation testing**

Automation testing is a process of converting any manual test cases into the test scripts with the help of automation tools, or any programming language is known as automation testing. With the help of automation testing, we can enhance the speed of our test execution because here, we do not require any human efforts. We need to write a test script and execute those scripts.

# Types of Software Testing

**Testing Strategies in Software Engineering**

Here are important strategies in software engineering:

**Unit Testing:** This software testing approach is followed by the programmer to test the unit of the program. It helps developers to know whether the individual unit of the code is working properly or not.

**Integration testing:** It focuses on the construction and design of the software. You need to see that the integrated units are working without errors or not.

**System testing:** In this method, your software is compiled as a whole and then tested as a whole. This testing strategy checks the functionality, security, portability, amongst others.

# Types of Software Testing

## Verification and validation

Verification and Validation is the process of investigating that a software system satisfies specifications and standards and it fulfills the required purpose.

- Verification is the process of checking that a software achieves its goal without any bugs. It is the process to ensure whether the product that is developed is right or not. It verifies whether the developed product fulfills the requirements that we have. Verification is Static Testing.

- Validation is the process of checking whether the software product is up to the mark or in other words product has high level requirements. It is the process of checking the validation of product i.e. it checks what we are developing is the right product. it is validation of actual and expected product. Validation is the Dynamic Testing.

# Verification vs validation

| Verification | Validation |
|---|---|
| • Verification addresses the concern: "Are you building it right?" | • Validation addresses the concern: "Are you building the right thing?" |
| • Ensures that the software system meets all the functionality. | • Ensures that the functionalities meet the intended behavior. |
| • Verification takes place first and includes the checking for documentation, code, etc. | • Validation occurs after verification and mainly involves the checking of the overall product. |
| • Done by developers. | • Done by testers. |
| • It has static activities, as it includes collecting reviews, walkthroughs, and inspections to verify a software. | • It has dynamic activities, as it includes executing the software against the requirements. |
| • It is an objective process and no subjective decision should be needed to verify a software. | • It is a subjective process and involves subjective decisions on how well a software works. |

Thank you