Black-Box Testing and White-Box Testing

Black box testing (also called functional testing) is testing that ignores the internal mechanism of a system or component and focuses solely on the outputs generated in response to selected inputs and execution conditions. White box testing (also called structural testing and glass box testing) is testing that takes into account the internal mechanism of a system or component.

What is White-Box Testing

Testing a system in a "black box" is doing so without knowing anything about how it operates within. A tester inputs data and monitors the output produced by the system being tested. This allows for the identification of the system's reaction time, usability difficulties and reliability concerns as well as how the system reacts to anticipated and unexpected user activities.

Because of the system's internal viewpoint, the phrase "white box" is employed. The term “clear box," "white box" or "transparent box" refers to the capability of seeing the software's inner workings through its exterior layer.

Developers carry it out before sending the program to. the testing team, who then conducts black-box testing. Testing the infrastructure of the application is the primary goal of white-box testing. As it covers unit testing and integration testing, it is performed at lower levels. Given that it primarily focuses on the code structure, pathways, conditions and ‘branches of a programed or piece of software, it necessitates programming skills. Focusing on the inputs and outputs via the program and enhancing its security are the main objectives of white-box testing:

It is also referred to as transparent testing, code-based testing, structural testing and clear box testing. It is a good fit and is recommended for testing algorithms.

Types of White Box Testing in Software Testing

White box testing is a type of software testing that examines the internal structure and design of a program or application. The following are some common types of white box testing:

1. Unit testing: Tests individual units or components of the software to ensure they function as intended.

2. Integration testing: Tests the interactions between different units or components of the software to ensure they work together correctly.

3. Functional testing: Tests the functionality of the software to ensure it meets the requirements and specifications.

4. Performance testing: Tests the performance of the software under various loads and conditions to ensure it meets performance requirements.

5. Security testing: Tests the software for vulnerabilities and weaknesses to ensure it is secure.

6. Code coverage testing: Measures the percentage of code that is executed during testing for ensure that all parts of the code are tested.

7. Regression testing: Tests the software after changes have been made to ensure that the changes did not introduce new bugs or issues.

Techniques of White Box Testing

There are some techniques which is used for white box testing -

1. Statement coverage: This testing approach involves going over every statement in the code to make sure that each one has been run at least once. As a result, the code is checked line by line.

2. Branch coverage: Is a testing approach in which test cases are created to ensure that each ranch is tested at léast once. This method examines all potential configurations for the system.

3. Path coverage: Path coverage is a software testing 2pproach that defines and covers all potential pathways. From system entrance to exit points, pathways are statements that may ‘be executed. It takes a lot of time.

4. Loop testing: With the help of this technique, loops and values in both independent and dependent code are examined. Errors often happen at the start and conclusion of loops. This method included testing loops

- Concatenated loops
- Simple loops
- Nested loops

5. Basis path testing: Using this methodology, control flow diagrams are created from code and subsequently calculations are made for cyclomatic complexity. For the purpose of designing the fewest possible test cases, cyclomatic complexity specifies the quantity of separate routes.

Advantages of White Box Testing

1. Complete coverage.
2. Better understanding of the system.
3. Improved code quality.
4. Increase efficiency.
5. Early detection of error.

Disadvantages of White Box Testing

1. This testing is very expensive and time-consuming.
2. Redesign of code needs test cases to be written again.
3. Missing functionalities cannot be detected.
4. This technique can be very complex and at times not realistic.
5. White-box testing requires a programmer with a high level of knowledge due for the complexity of the level of testing that needs to be done.

What is Black Box Testing

Testing a system in a "black box" is doing so without knowing anything about how it operates within. A tester inputs data and monitors the output produced by the system being tested. This allows for the identification of the system's reaction time, usability difficulties and reliability concerns as well as how the system reacts to anticipated and unexpected user activities.

Because it tests a system from beginning to finish, black box testing is a potent testing method. A tester may imitate user action to check if the system fulfills its promises, much as end users "don't care” how a system is programmed or designed and expect to get a suitable answer to their requests.

A black box test assesses every important subsystem along the route, including the UVUX, database, dependencies and integrated systems, as well as the web server or application server.

Black Box Testing Pros and Cons

S. No	Pros	Cons
1	Testers do not require technical knowledge, programming of IT skills.	Difficult to automate.
2	Testers do not need to learn implementation details of the system.	Requires prioritization, typically infeasible to tests all user paths.
3	Tests can be executed by crowdsourced or outsourced testers.	Difficult to calculate test coverages.
4	Low chance of false positives.	If a test fails, it can be difficult to understand the root cause of the issues.
5	Tests have lower complexity, since they simply model common user behavior	Tests may be conducted at low scale or on a non-production like environment.

Types of Black Box Testing

Black box testing can be applied to three main types of tests : Functional, non-functional and regression testing.

1. Functional testing:

Specific aspects or operations of the program that is being tested may be tested via black box testing. For instance, make sure that the right user credentials may be used to log in and that the incorrect ones cannot.

Functional testing might concentrate on the most important features of the program (smoke testing/sanity testing), on how well the system works as a whole (system testing) or non the integration of its essential components.

2. Non-functional testing:

- Beyond features and functioning, black box testing allows for the inspection of extra software components. A non-functional test examines "how" rather than "if" the program can carry out a certain task.
- Black box testing may determine whether software is:

a) Usable and simple for its users to comprehend;

b) Performant under predicted or peak loads; Compatible with relevant devices, screen sizes, browsers or operating systems;

c) Exposed to security flaws or frequent security threats.

3. Regression testing:

To determine if a new software version displays a regression or a decrease in capabilities, from one version to the next. black box testing may be employed. Regression testing may be used to evaluate both functional and non-functional features of the program, such as when a particular feature no longer functions as anticipated in the new version or when a 8 formerly fast-performing action becomes much slower in the new version.

Black Box Testing Techniques

1. Equivalence partitioning:

Testing professionals may organize potential inputs into "partitions" and test just one sample input from each category. For instance, it is sufficient for testers to verify one birth date in the "under 18" group and one date in the "over 18" group if a system asks for a user's birth date and returns the same answer for users under the age of 18 and a different response for users over 18.

2. Boundary value analysis:

Testers can determine if a system responds differently around a certain boundary value. For instance, a particular field could only support values in the range of 0 and 99.

Testing personnel may concentrate on the boundary values (1, 0, 99 and 100) to determine if the system is appropriately accepting and rejecting inputs.

Decision Table Testing

Numerous systems provide results depending on a set of parameters. Once rules” that are combinations of criteria have been identified, each rule's conclusion can then be determined and test cases may then be created for each r

Differences between Black Box Testing vs White Box Testing:

Black Box Testing	White Box Testing
It is a way of software testing in which the internal structure or the program or the code is hidden and nothing is known about it.	It is a way of testing the software in which the tester has knowledge about the internal structure or the code or the program of the software.
Implementation of code is not needed for black box testing.	Code implementation is necessary for white box testing.
It is mostly done by software testers.	It is mostly done by software developers.
No knowledge of implementation is needed.	Knowledge of implementation is required.
It can be referred to as outer or external software testing.	It is the inner or the internal software testing.
It is a functional test of the software.	It is a structural test of the software.
This testing can be initiated based on the requirement specifications document.	This type of testing of software is started after a detail design document.
No knowledge of programming is required.	It is mandatory to have knowledge of programming.
It is the behavior testing of the software.	It is the logic testing of the software.
It is applicable to the higher levels of testing of software.	It is generally applicable to the lower levels of software testing.
It is also called closed testing.	It is also called as clear box testing.
It is least time consuming.	It is most time consuming.
It is not suitable or preferred for algorithm testing.	It is suitable for algorithm testing.
Can be done by trial and error ways and methods.	Data domains along with inner or internal boundaries can be better tested.
Example: Search something on google by using keywords	**Example:** By input to check and verify loops
Black-box test design techniques- • Decision table testing • All-pairs testing • Equivalence partitioning • Error guessing	**White-box test design techniques-** • Control flow testing • Data flow testing • Branch testing

Types of Black Box Testing:	Types of White Box Testing:

• Functional Testing • Non-functional testing • Regression Testing	• Path Testing • Loop Testing • Condition testing
It is less exhaustive as compared to white box testing.	It is comparatively more exhaustive than black box testing.