Black Box Testing

Black box testing is a technique of software testing which examines the functionality of software without peering into its internal structure or coding.

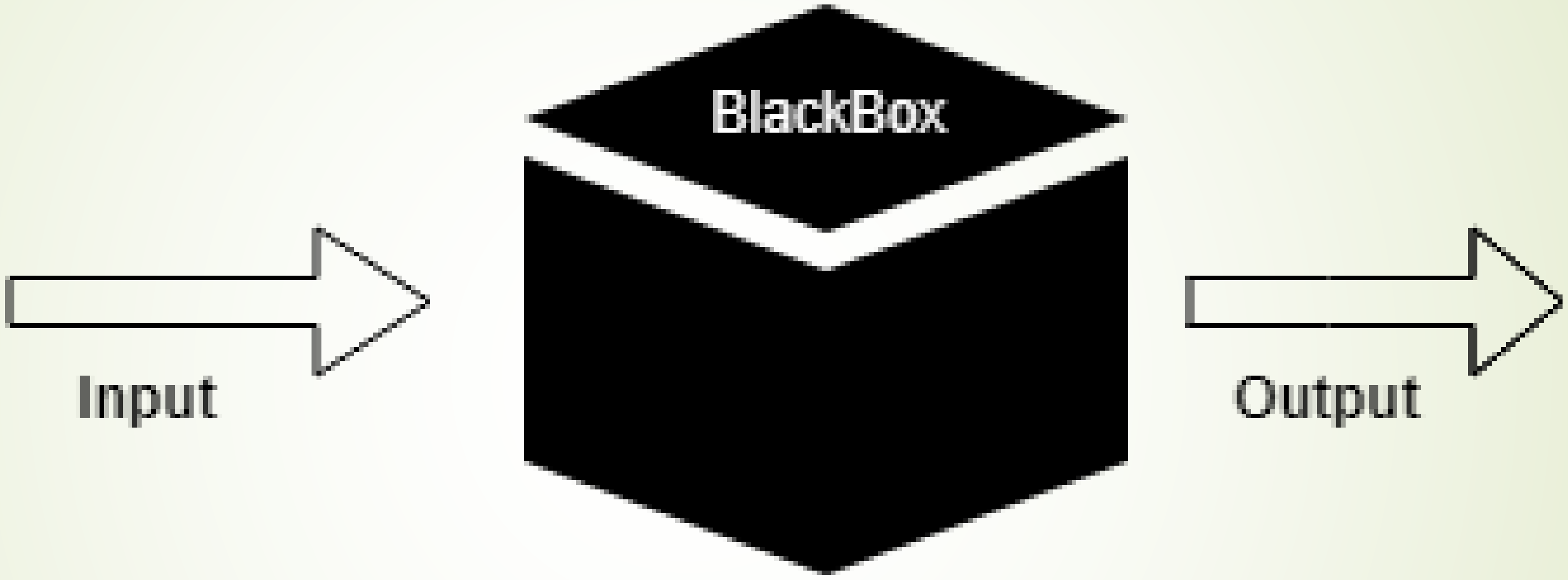
The primary source of black box testing is a specification of requirements that is stated by the customer.

In this method, tester selects a function and gives input value to examine its functionality, and checks whether the function is giving expected output or not.

If the function produces correct output, then it is passed in testing, otherwise failed.

The test team reports the result to the development team and then tests the next function.

After completing testing of all functions if there are severe problems, then it is given back to the development team for correction.



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.

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.

Test procedure

The test procedure of black box testing is a kind of process in which the tester has specific knowledge about the software's work, and it develops test cases to check the accuracy of the software's functionality.

It does not require programming knowledge of the software. All test cases are designed by considering the input and output of a particular function.

A tester knows about the definite output of a particular input, but not about how the result is arising.

There are various techniques used in black box testing for testing like decision table technique, boundary value analysis technique, state transition, All-pair testing, cause-effect graph technique, equivalence partitioning technique, error guessing technique, use case technique and user story technique.

Test cases

Test cases are created considering the specification of the requirements.

These test cases are generally created from working descriptions of the software including requirements, design parameters, and other specifications.

For the testing, the test designer selects both positive test scenario by taking valid input values and adverse test scenario by taking invalid input values to determine the correct output.

Test cases are mainly designed for functional testing but can also be used for non-functional testing.

Test cases are designed by the testing team, there is not any involvement of the development team of software.

Techniques Used in Black Box Testing

- **Decision Table Technique** - Decision Table Technique is a systematic approach where various input combinations and their respective system behavior are captured in a tabular form. It is appropriate for the functions that have a logical relationship between two and more than two inputs.

- **Boundary value Technique** - Boundary Value Technique is used to test boundary values, boundary values are those that contain the upper and lower limit of a variable. It tests, while entering boundary value whether the software is producing correct output or not.

- **State Transition Technique** - State Transition Technique is used to capture the behavior of the software application when different input values are given to the same function. This applies to those types of applications that provide the specific number of attempts to access the application.

- ❑ **All-pair testing technique** - All-pair testing Technique is used to test all the possible discrete combinations of values. This combinational method is used for testing the application that uses checkbox input, radio button input, list box, text box, etc.

- ❑ **Cause-Effect Technique** - Cause-Effect Technique underlines the relationship between a given result and all the factors affecting the result.It is based on a collection of requirements.

- ❑ **Equivalance partition technique** - Equivalence partitioning is a technique of software testing in which input data divided into partitions of valid and invalid values, and it is mandatory that all partitions must exhibit the same behavior.

- ❑ **Error guessing technique** - Error guessing is a technique in which there is no specific method for identifying the error. It is based on the experience of the test analyst, where the tester uses the experience to guess the problematic areas of the software.

- ❑ **Use case technique** - Use case Technique used to identify the test cases from the beginning to the end of the system as per the usage of the system. By using this technique, the test team creates a test scenario that can exercise the entire software based on the functionality of each function from start to end.

Various approaches to black-box testing

There are a set of approaches for black-box testing.

❑ **Manual UI Testing:** In this approach, a tester checks the system as a user. Check and verify the user data, error messages.

❑ **Automated UI Testing:** In this approach, user interaction with the system is recorded to find errors and glitches. Testers can set record demand as per schedule.

❑ **Documentation Testing:** In this approach, a tester purely checks the input and output of the software. Testers consider what system should perform rather than how. It is a manual approach to testing.

Black box testing example

A black box testing will not consider the specifications of the code, and it will test the valid username and password to login to the right account.

❑ A user is logging in with the correct username and password.

❑ A user receives an error message when enters incorrect username and password.

❑ A user might enter the password in the wrong format, and a user might not receive an error message on entering an incorrect password.

In this scenario, the software has failed to show the error message to the user. This failed scenario will be noted by the black box tester and then he will inform the developer about this.
This form of testing mitigates the risk of software failures at the user's end.

When we do Black Box testing?

Unlike traditional white box testing, black box testing is beneficial for testing software usability.

The overall functionality of the system under test
Black box testing gives you a broader picture of the software.

This testing approach sees an application from a user’s perspective.

To test the software as a whole system rather than different modules.

What are the benefits of Black Box testing?

The tester doesn’t need any technical knowledge to test the system. It is essential to understand the user’s perspective.

Testing is performed after development, and both the activities are independent of each other.

It works for a more extensive coverage which is usually missed out by testers as they fail to see the bigger picture of the software.

Test cases can be generated before development and right after specification.

Black box testing methodology is close to agile.