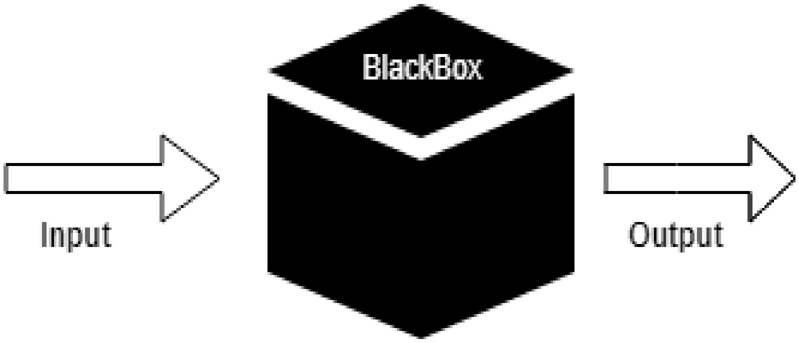
TASK-7 BLACK BOX TESTING

Black box is testing of **functionality** of the software without peering into the internal structure of coding. In black box testing we check the functionality by giving an input and check whether we get the expected output. If we don't get the correct output, we report to the developer to correct the mistake.

The primary source for black box testing is – specification of customer requirements. In the test procedure of black box testing the tester has some knowledge about how the software works and develops the testcases to check the accuracy. This doesn't need the knowledge of coding.



In white box testing the developers will check does the source code is working as expected or not, whereas in backbox testing the testers will check whether the application is working according to the end user or not. The objective of Whitebox testing is Code coverage and the objective of Blackbox testing is Requirement coverage.

TECHNIQUES USED IN BLACK BOX TESTING ARE:

1. DECISION TABLE TECHNIQUE
2. BOUNDARY VALUE TECHNIQUE
3. STATE TRANSITON TECHNIQUE
4. ALL-PAIR TESTING TECHNIQUE
5. CAUSE EFFECT TECHNIQUE
6. EQUIVALENT PARTIONING TECHNIQUE
7. ERROR GUESSING TECHNIQUE

8. USE CASE TECHNIQUE

Decision table technique:

Design table technique is a widely used case technique in black box testing. This is a systematic approach where various input combinations and their respective system behavior is captured in tabular form. That's why it is called the cause effect table. This technique is used to pick the testcases systematically, saving testing time and giving good coverage to the area.

EX: -If we login an email then we may have various input testcases in decision table where we might have a right input to email but not a right password, we get a particular error message. Similarly for every mistake we get its error message.

Boundary Value Analysis:

Boundary value analysis is one of the widely used case design techniques for black box testing. It is used to test boundary values because the values near the boundary may have high chances of error. Whenever we do the testing by boundary value analysis, the tester focuses on while entering the boundary value whether the software is working properly or not.

State Transition Technique:

The general meaning of state transition is different forms of the same situation, and according to the meaning, the state transition 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 such as the login function of an application which gets locked after the specified number of incorrect attempts. In this testing we check the state of the application is changing after the change in input.

All-Pairs Testing Technique:

All-pairs testing technique is also known as pairwise testing. It 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 (radio button is used when you must select only one option for example when you select gender male or female, you can select only one option), list box, text box, etc.

Cause Effect Technique:

Cause effect graph comes under the black box testing technique which underlines the relationship between a given result and all the factors affecting the result. It is used to write dynamic test cases. The dynamic test cases are used when code works dynamically based on user input. For example, while using email account, on entering valid email, the system accepts it but, when you enter invalid email, it throws an error message. In this technique, the input conditions are assigned with causes and the result of these input conditions with effects. Cause-Effect graph technique converts the requirements specification into a logical relationship between the input and output conditions by using logical operators like AND, OR and NOT.

Equivalent partitioning Technique:

Equivalence partitioning is a technique in software testing in which the input data is classified into valid and invalid input data and the condition for all partitions must be similar. The equivalence partitions are derived from requirements and specifications of the software. The advantage of this approach is it helps to reduce the time of testing due to a smaller number of test cases from infinite to finite. It is applicable at all levels of the testing process.

EX: -otp of 6 digits, Mobile number of 10 digits. In both cases in valid range, we get the other page in invalid data we get invalid message.

Error Guessing Technique:

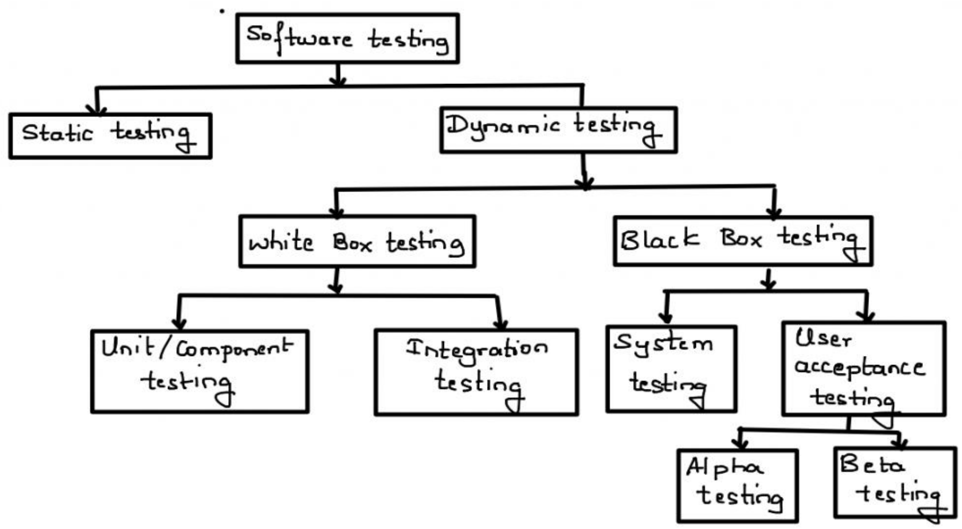
Error guessing is a technique in which there is no specific method for identifying the error. It is based on the experience of the test analyst, where the tester uses the experience to guess the problematic areas of the software. It is a type of black box testing technique which does not have any defined structure to find the error.

In this approach, every test engineer will derive the values or inputs based on their understanding or assumption of the requirements, and we do not follow any kind of rules to perform error guessing technique.

EX: Divide by zero, Null pointer exception, Enter invalid parameters.

Use case Technique: The use case is functional testing of the black box testing 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. The client provides the customer requirement specification for the application, then the development team will write the use case according to the CRS, and the use case is sent to the customer for their review.

SOFTWARE TESTING TYPES:


Software testing
Static testing
Dynamic testing
white Box testing
Black Box testing
Unit/Component testing
Integration testing
System testing
User acceptance testing
Alpha testing
Beta testing