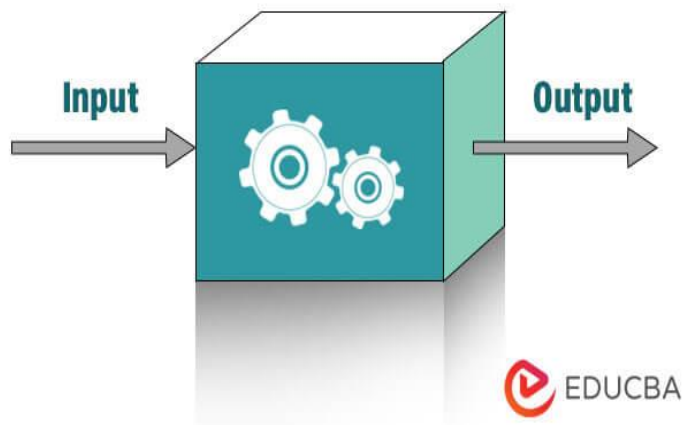


WHITE BOX TESTING: TASK-6

The box type testing is of three types of black box, grey box and white box. The white box testing is also called Glass box testing, clear box testing, structural testing, open box testing and transparent box testing. It tests the internal structure and working of the code and focuses on checking predefined inputs on expected or desired outputs. It is based on the internal working of the application. In this type of testing, we need coding knowledge for testing.

Developer does the white box testing. In this the developer tests every line of code and sends to the tester for black box testing. The Tester identifies the bug and sends it to the developer and the developer removes the bug and fixes the code. The tester should fix the code because fixing the code may change or affect other features of the code and fixing the code takes a lot of time so the tester may not find enough time to find out other bugs in it.

White Box Testing



White box testing has various kinds of testing like,

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

Rational purify:

We have a built-in tool, which helps us to test the needless variables and functions. And here we have the tool called Rational purify.

Rational Quantify:

Once the entire code is ready, the rational quantify tool will go through the code and execute it. And we can see the outcome in the result sheet in the form of thick and thin lines. The thick line specifies which section of code is time-consuming. When we double-click on the thick line, the tool will take us to that line or piece of code automatically, which is also displayed in a different color. We can change that code again and use this tool.

Reasons For White Box Testing:

- It identifies internal security holes.
- To check the way of input in the code
- To check the functionality of the conditional loops.
- To test function, object and statement at an individual level.

ADVANTAGES OF WHITE BOX TESTING:

- White box testing optimizes code so hidden errors can be identified.
- Test cases of white box testing can be easily automated.
- This testing is more thorough than other testing approaches as it covers all code paths.
- It can be started in the SDLC phase even without GUI.

DISADVANTAGES OF WHITE BOX TESTING:

- White box testing is too time consuming when it comes to large-scale programming applications.
- White box testing is much more expensive and complex.
- It can lead to production error because it is not detailed by the developers.
- White box testing needs professional programmers who have a detailed knowledge and understanding of programming language and implementation.

WHITE BOX TEST CASE DESIGN TECHNIQUES:

Data Flow Testing: Data flow testing uses the control flow graph to detect illogical things that can interrupt the flow of data. Anomalies in the flow of data are detected at the time of associations between values and variables due to:

- If the variables are used without initialization.
- If the initialized variables are not used at least once.

CONTROLFLOW TESTING:

In this technique, a particular part of a large program is selected by the tester to set the testing path. It is mostly used in unit testing. Test cases represented by the control graph of the program.

BRANCH COVERAGE TESTING:

Branch coverage technique is used to cover all branches of the control flow graph. It covers all the possible outcomes (true and false) of each condition of decision point at least once. Branch coverage technique is a whitebox testing technique that ensures that every branch of each decision point must be executed.

However, branch coverage technique and decision coverage technique are very similar, but there is a key difference between the two. Decision coverage technique covers all branches of each decision point whereas branch testing covers all branches of every decision point of the code

CODE COVERAGE:

The percentage of code tested during white box testing is called code coverage. The source code behind a unit may contain a set of statements, a set of conditions and a set of branches. If all these are tested, then we say 100% code coverage is achieved.

Statement coverage:

The percentage of statements tested during white box testing is called statement coverage.

Statement coverage= (Number statements tested/total number of statements) x 100

Path coverage:

The percentage of paths or branches exercised or tested during white box testing is called path coverage.

Path coverage= (Number of paths executed/total number of paths) x 100%

CONDITION COVERAGE:

Condition coverage testing is a type of white box testing that tests all the conditional expressions in a program for all possible outcomes of the conditions. It is also called predicate coverage. In branch coverage, all conditions must be executed at least once. On the other hand, in condition coverage, all possible outcomes of all conditions must be tested at least once

FUNCTIONAL COVERAGE:

The number of functions that are called and executed at least once in the source code

Function Coverage = (Number of functions called)/(Total number of functions) *100.