Experiment No 8

To study testing techniques.

Prepared by

Abhishek Vasant Girkar	VU4F1718022
Mohit Khambayat	VU4F1718072
Ajay Waaghmare	VU2T4SF1718028
Rasika Mahadik	VU4F1718004

Instructor: Prof. Vinod Sapkal
Course : SEPM
Class: TE-IT (A)/Batch A

Experiment No 8

Aim: To study testing techniques.

Theory:

Testing is a process of executing a program with the aim of finding error. To make our software perform well it should be error free.If testing is done successfully it will remove all the errors from the software.

Types of software testing:

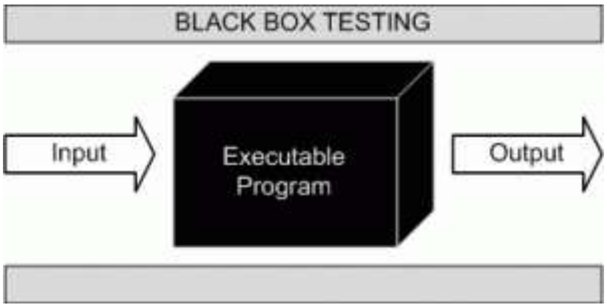
Black Box Testing is a software testing method in which the internal structure/design/ implementation of the item being tested is not known to the tester. These tests can be functional or non-functional, though usually functional. Test design techniques include: ***Equivalence partitioning, Boundary Value Analysis, Cause Effect Graphing.***

White Box Testing is a software testing method in which the internal structure/design/ implementation of the item being tested is known to the tester. Test design techniques include: ***Control flow testing, Data flow testing, Branch testing, and Path testing.***

BLACK BOX TESTING Fundamentals

DEFINITION

Black Box Testing, also known as Behavioral Testing, is a software testing method in which the internal structure/ design/ implementation of the item being tested is not known to the tester. These tests can be functional or non-functional, though usually functional.



This method is named so because the software program, in the eyes of the tester, is like a black box; inside which one cannot see. This method attempts to find errors in the following categories:

- Incorrect or missing functions
- Interface errors
- Errors in data structures or external database access
- Behaviour or performance errors
- Initialization and termination errors

EXAMPLE

A tester, without knowledge of the internal structures of a website, tests the web pages by using a browser; providing inputs (clicks, keystrokes) and verifying the outputs against the expected outcome.

LEVELS APPLICABLE TO

Black Box Testing method is applicable to the following levels of software testing:

- Integration Testing
- System Testing
- Acceptance Testing

The higher the level, and hence the bigger and more complex the box, the more black box testing method comes into use.

BLACK BOX TESTING TECHNIQUES

Following are some techniques that can be used for designing black box tests.

- **Equivalence partitioning:** It is a software test design technique that involves dividing input values into valid and invalid partitions and selecting representative values from each partition as test data.
- ***Boundary Value Analysis:*** It is a software test design technique that involves determination of boundaries for input values and selecting values that are at the boundaries and just inside/ outside of the boundaries as test data.
- ***Cause Effect Graphing:*** It is a software test design technique that involves identifying the cases (input conditions) and effects (output conditions), producing a Cause-Effect Graph, and generating test cases accordingly.

BLACK BOX TESTING ADVANTAGES

- ➢ Tests are done from a user's point of view and will help in exposing discrepancies in the specifications.
- ➢ Tester need not know programming languages or how the software has been implemented.
- ➢ Tests can be conducted by a body independent from the developers, allowing for an objective perspective and the avoidance of developer-bias.
- ➢ Test cases can be designed as soon as the specifications are complete.

BLACK BOX TESTING DISADVANTAGES

- ➢ Only a small number of possible inputs can be tested and many program paths will be left untested.
- ➢ Without clear specifications, which is the situation in many projects, test cases will be difficult to design.
- ➢ Tests can be redundant if the software designer/ developer has already run a test case.
- ➢ Ever wondered why a soothsayer closes the eyes when foretelling events? So is almost the case in Black Box Testing.

WHITE BOX TESTING Fundamentals

DEFINITION

White Box Testing (also known as Clear Box Testing, Open Box Testing, Glass Box Testing, Transparent Box Testing, Code-Based Testing or Structural Testing) is a software testing method in which the internal structure/ design/ implementation of the item being tested is known to the tester. The tester chooses inputs to exercise paths through the code and determines the appropriate outputs. Programming know-how and the implementation knowledge is essential. White box testing is testing beyond the user interface and into the nitty-gritty of a system.

This method is named so because the software program, in the eyes of the tester, is like a white/ transparent box; inside which one clearly sees.

Definition by ISTQB

- **White-Box testing:** Testing based on an analysis of the internal structure of the component or system.
- **White-Box test design technique:** Procedure to derive and/or select test cases based on an analysis of the internal structure of a component or system.

EXAMPLE

A tester, usually a developer as well, studies the implementation code of a certain field on a webpage, determines all legal (valid and invalid) AND illegal inputs and verifies the outputs against the expected outcomes, which is also determined by studying the implementation code.

White Box Testing is like the work of a mechanic who examines the engine to see why the car is not moving.

LEVELS APPLICABLE TO

White Box Testing method is applicable to the following levels of software testing:

- **Unit Testing:** For testing paths within a unit.
- **Integration Testing:** For testing paths between units.
- **System Testing:** For testing paths between subsystems.

However, it is mainly applied to Unit Testing.

WHITE BOX TESTING ADVANTAGES

- Testing can be commenced at an earlier stage. One need not wait for the GUI to be available.
- Testing is more thorough, with the possibility of covering most paths.

WHITE BOX TESTING DISADVANTAGES

- Since tests can be very complex, highly skilled resources are required, with thorough knowledge of programming and implementation.
- Test script maintenance can be a burden if the implementation changes too frequently.

Since this method of testing it closely tied with the application being testing, tools to cater to every kind of implementation/platform may not be readily available.

Conclusion: Hence , We studied different testing technique.