CHAPTER - 5

TESTING - --

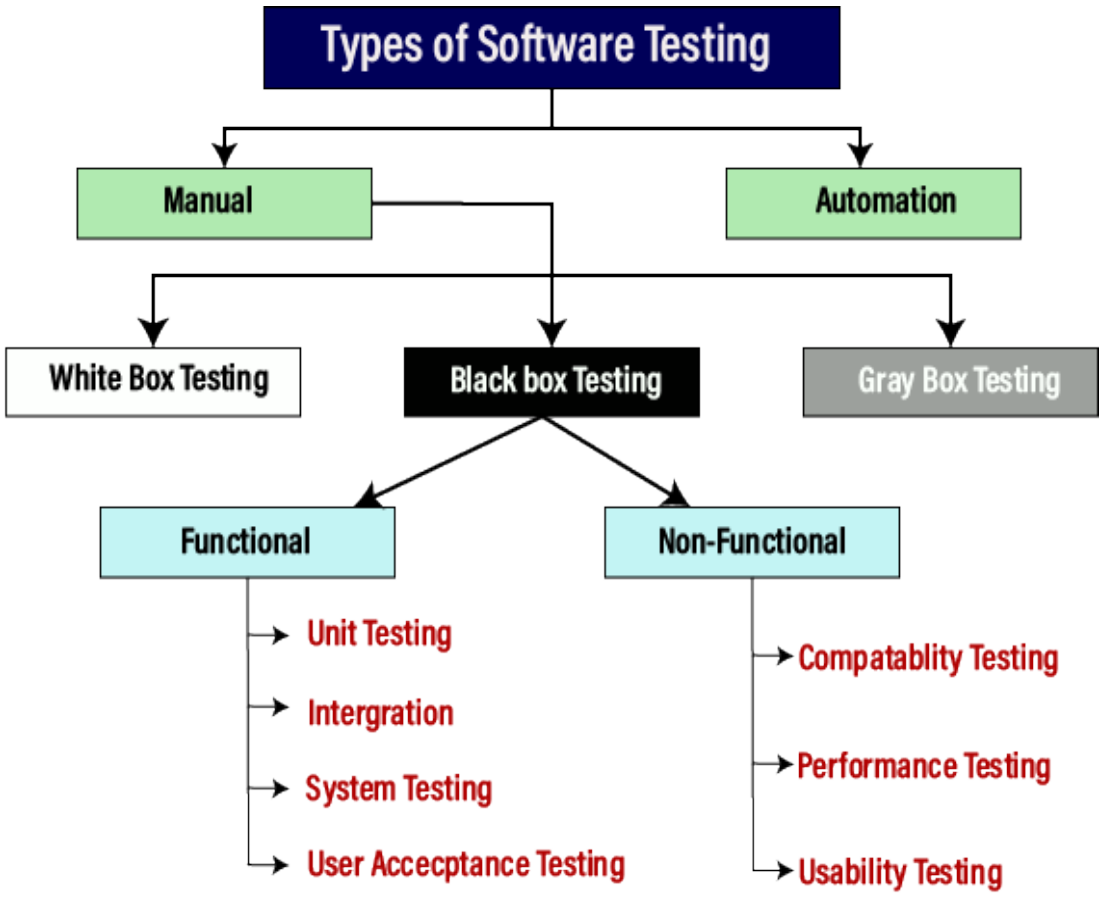
Software testing is an activity performed to uncover error. It is a critical element of software quality assurance and represents the ultimate review of specification, design and coding.

Testing objectives-

According to Glen Myers the testing objectives are –

Testing is a process of executing a program with the intend of finding an error.

A good test case is one that has probability of finding an undiscovered error.



Manual Testing

Manual testing includes testing a software manually, i.e., without using any automated tool or any script. In this type, the tester takes over the role of an end-user and tests the software to identify any unexpected behavior or bug. There are different stages for manual testing such as unit testing, integration testing, system testing, and user acceptance testing.

Automation Testing

Automation testing, which is also known as Test Automation, is when the tester writes scripts and uses another software to test the product. This process involves automation of a manual process. Automation Testing is used to re-run the test scenarios that were performed manually, quickly, and repeatedly.

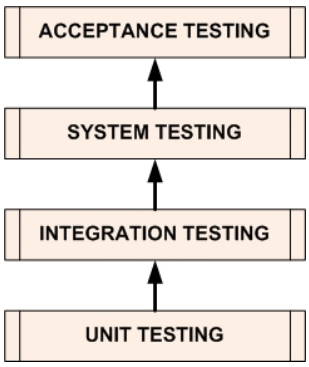
LEVELS OF TESTING / Testing Strategy

1. Unit testing - In this type of testing techniques are applied to detect the errors from each software component individually.

2. Integration testing - It focuses on issues associated with verification and program construction as components begin interacting with one another.

3. Validation testing - It provides assurance that the software validation criteria (established during requirements analysis) meets all functional, behavioural, and performance requirements.

4. System testing - In system testing all system elements forming the system is tested as a whole.



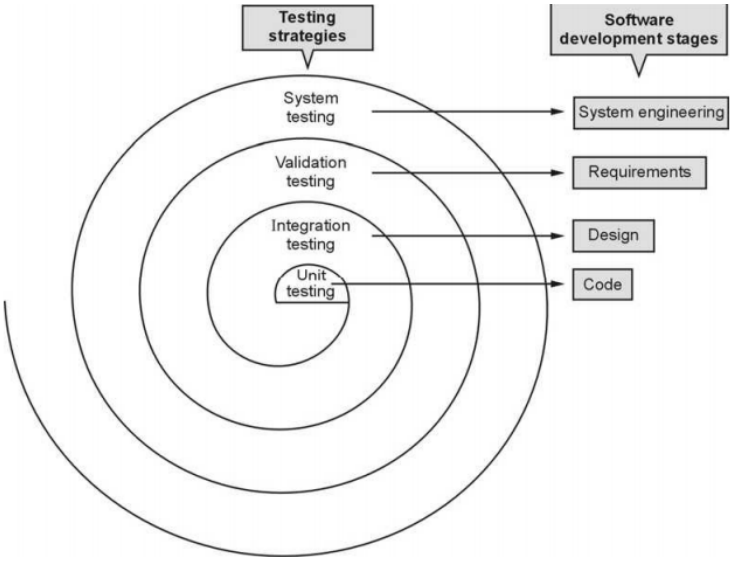


Fig. 4.6.1 Testing strategy

Unit Testing

- The focus is to uncover the errors in design and implementation.
- The various tests that are conducted during the unit test are described as below.

1. Module interfaces are tested for proper information flow in and out of the program.
2. Local data are examined to ensure that integrity is maintained.
3. Boundary conditions are tested to ensure that the module operates properly at boundaries established to limit or restrict processing.
4. All the basis (independent) paths are tested for ensuring that all statements in the module have been executed only once.
5. Drivers and stub software need to be developed to test incomplete software. The "driver" is a program that accepts the test data and prints the relevant results. And the "stub" is a subprogram that uses the module interfaces and performs the minimal data manipulation if required.

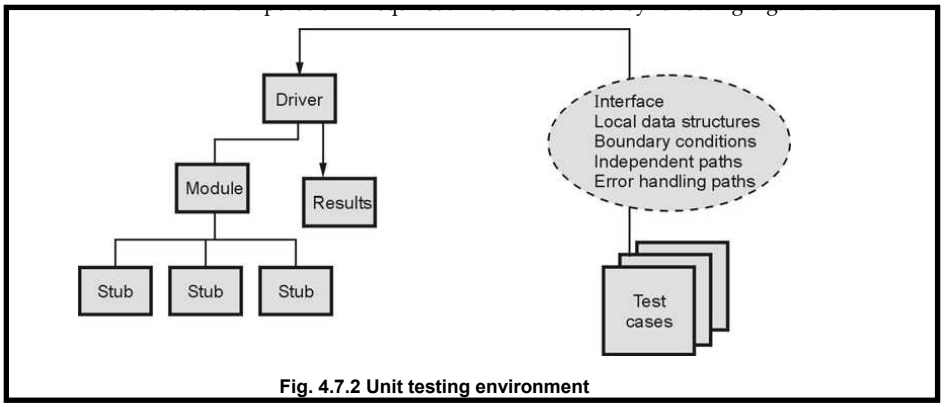


Fig. 4.7.2 Unit testing environment

INTEGRATION TESTING -

1. A group of dependent components are tested together to ensure their quality of their integration unit.

2. The objective is to take unit tested components and build a program structure that has been dictated by software design.

The focus of integration testing is to uncover errors in :

- ➢ Design and construction of software architecture.
- ➢ Integrated functions or operations at subsystem level.
- ➢ Interfaces and interactions between them.
- ➢ Resource integration and/or environment integration.

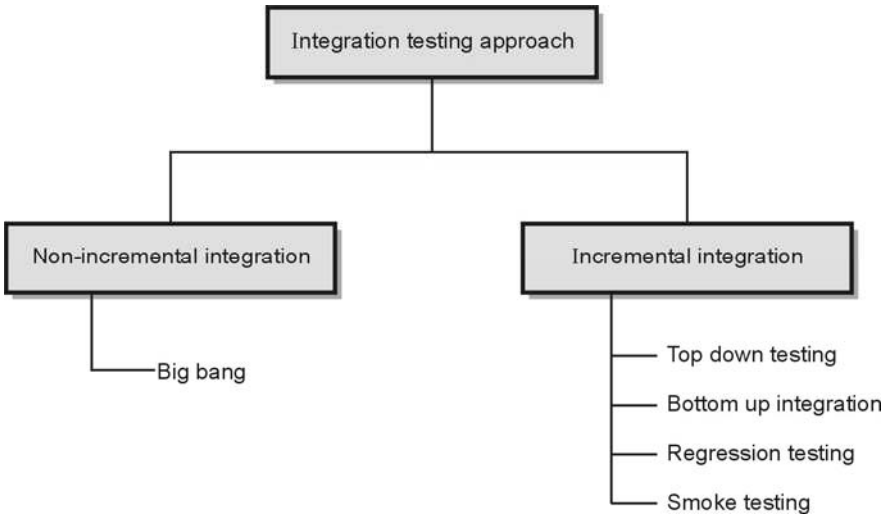


Fig. 4.8.1 Integration testing approach

/

Non incremental integration is given by the "big bang" approach. All components are combined in advance. The entire program is tested as a whole. And chaos usually results. A set of errors is tested as a whole. Correction is difficult because isolation of causes is complicated by the size of the entire program. Once these errors are corrected new ones appear.

An incremental construction strategy includes

(1) Top down integration

(2) Bottom up integration

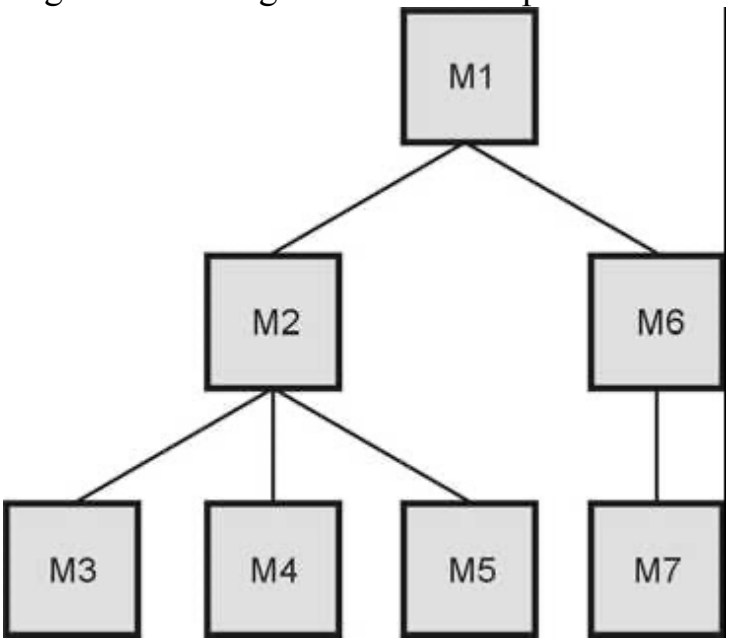
(3) Regression testing

(4) Smoke testing

TOP DOWN TESTING -

- Top down testing is an incremental approach in which modules are integrated by moving down through the control structure.
- Modules subordinate to the main control module are incorporated into the system in either a depth first or breadth first manner.
- Integration process can be performed using following steps.

1. The main control module is used as a test driver and the stubs are substituted for all modules directly subordinate to the main control module.
2. Subordinate stubs are replaced one at a time with actual modules using either depth first or breadth first method.
3. Tests are conducted as each module is integrated.
4. On completion of each set of tests, another stub is replaced with the real module.
5. Regression testing is conducted to prevent the introduction of new errors.



Program structure

In top down integration if the depth first approach is adopted then we will start integration from module M1 then we will integrate M2 then M3, M4, M5, M6 and then M7.

If breadth first approach is adopted then we will integrate module M1 first then M2, M6. Then we will integrate module M3, M4, M5 and finally M7.

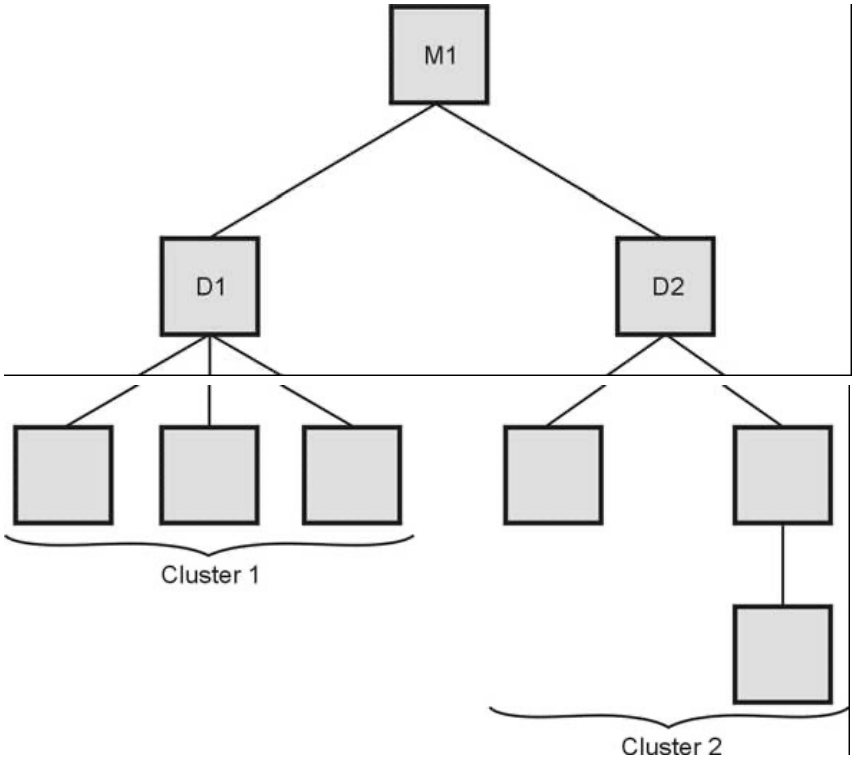
Bottom Up Integration Testing

In bottom up integration the modules at the lowest levels are integrated at first, then

integration is done by moving upward through the control structure.
The bottom up integration process can be carried out using following steps.

1. Low-level modules are combined into clusters that perform a specific software subfunction.

2. Drivers are removed and clusters are combined moving upward in the program structure.



Regression Testing

- Regression testing is used to check for defects propagated to other modules by changes made to existing program. Thus regression testing is used to reduce the side effects of the changes.

- After product had been deployed, regression testing would be necessary because after a change has been made to the product an error that can be discovered and it should be corrected. Similarly for deployed product addition of new feature may be
  requested and implemented. For that reason regression testing is essential.

Smoke Testing

The smoke testing is a kind of integration testing technique used for time critical projects wherein the project needs to be assessed on frequent basis.

Smoke testing benefits

1. Integration risk is minimized.
2. The quality of the end product is improved.
3. Error diagnosis and correction are simplified.
4. Assessment of progress is easy.

Acceptance Testing

The acceptance testing is a kind of testing conducted to ensure that the software works correctly in the user work environment.
The acceptance testing can be conducted over a period of weeks or months. The types of acceptance testing are

1. Alpha test - The alpha testing is a testing in which the version of complete software is tested by the customer under the supervision of developer. This testing is performed at developer's site. The software is used in natural setting in presence of developer. This test is conducted in controlled environment.

2. Beta test - The beta testing is a testing in which the version of software is tested by the customer without the developer being present. This testing is performed at customer's site. As there is no presence of developer during

testing, it is not controlled by developer. The end user records the problems and report them to developer. The developer then makes appropriate modification.

SYSTEM TESTING-

The system test is a series of tests conducted for fully the computer based system. Various types of system tests are

1. Recovery testing

2. Security testing

3. Stress testing

4.Performance testing

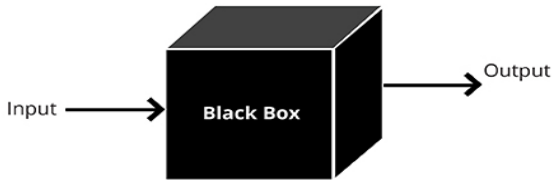
Internal and External Views of Testing—

There are two views of the testing. The internal view and external view. The internal view is also known as white box testing and the external view is also known as black box testing.

BLACK BOX TESTING -

1. The black box testing is used to demonstrate that the software functions are operational.
2. In black box testing, it is tested whether the input is accepted properly and output is correctly produced.
3. The major focus of black box testing is on functions, operations, external interfaces, external data and information.
4. The black box testing is also called as behavioural testing.
5. Black box testing methods focus on the functional requirements of the software.
6. Test sets are derived that fully exercise all functional requirements.
7. The black box testing is not an alternative to white box testing and it uncovers different class of errors than white box testing

BLACK BOX TESTING APPROACH



Why to perform black box testing ?
Black box testing uncovers following types of errors.
1.Incorrect or missing functions
2. Interface errors
3.Errors in data structures
4.Performance errors
5.Initialization or termination errors

BLACK BOX TECHNIQUE-

1.Equivalence Partitioning -

- It is a black box technique that divides the input domain into classes of data. From this data test cases can be derived.
- An ideal test case uncovers a class of errors that might require many arbitrary test cases to be executed before a general error is observed.
- In equivalence partitioning the equivalence classes are evaluated for given input condition. Equivalence class represents a set of valid or invalid states for input conditions.

Equivalence class guidelines can be as given below :

➢ If input condition specifies a range, one valid and two invalid equivalence classes are defined.
➢ If an input condition requires a specific value, one valid and two invalid equivalence classes are defined.
➢ If an input condition specifies a member of a set, one valid and one invalid equivalence class is defined.
➢ If an input condition is Boolean, one valid and one invalid equivalence class is defined.

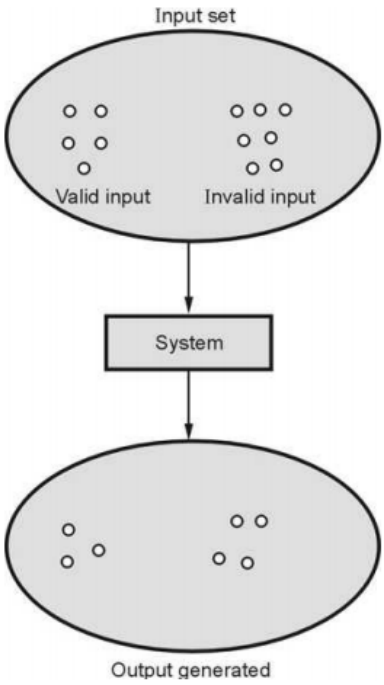


Fig. 4.4.1

For example :

Area code : Input condition, Boolean - The area code may or may not be present.

Input condition, range - Value defined between 200 and 700.

Password : Input condition, Boolean - A password may or may not be present.

Input condition, value - Seven character string.

Command : Input condition, set - Containing commands noted before.

Boundary Value Analysis (BVA)-

Boundary value analysis is done to check boundary conditions.

- · A boundary value analysis is a testing technique in which the elements at the edge of the domain are selected and tested.
- · Using boundary value analysis, instead of focusing on input conditions only, the test cases from output domain are also derived.
- · Boundary value analysis is a test case design technique that complements equivalence partitioning technique.

Guidelines for boundary value analysis technique are

1. If the input condition specified the range bounded by values x and y, then test cases should be designed with values x and y. Also test cases should be with the values above and below x and y.
2. If input condition specifies the number of values then the test cases should be designed with minimum and maximum values as well as with the values that are just above and below the maximum and minimum should be tested.
3. If the output condition specified the range bounded by values x and y, then test cases should be designed with values x and y. Also test cases should be with the values above and below x and y.
4. If output condition specifies the number of values then the test cases should be designed with minimum and maximum values as well as with the values that are just above and below the maximum and minimum should be tested.
5. If the internal program data structures specify such boundaries then the test cases must be designed such that the values at the boundaries of data structure can be tested.

✓ **Advantages and Disadvantages of Black Box Testing**

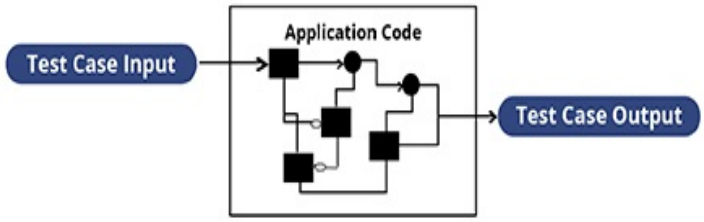
Advantages :
1. The black box testing focuses on fundamental aspect of system without being concerned for internal logical structure of the software.
2. The advantage of conducting black box testing is to uncover following types of errors.
i. Incorrect or missing functions
ii. Interface errors
iii. Errors in external data structures
iv. Performance errors
v. Initialization or termination errors

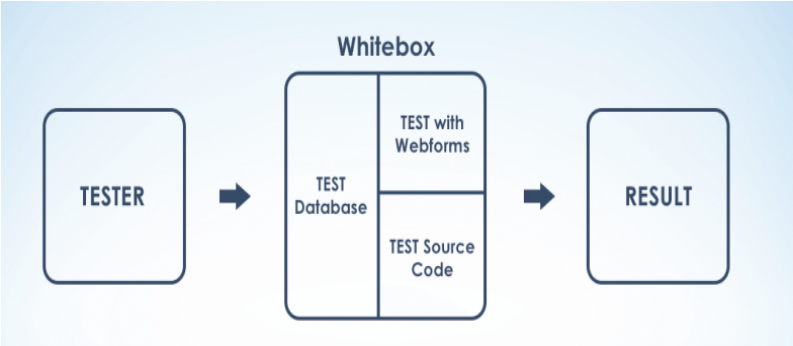
Disadvantages :
1. All the independent paths within a module cannot be tested.
2. Logical decisions along with their true and false sides cannot be tested.
3. All the loops and the boundaries of these loops cannot be exercised with black box testing.
4. Internal data structure cannot be validated.

WHITE BOX TESTING -

➢ In white box testing the procedural details are closely examined.
➢ In this testing the internals of software are tested to make sure that they operate according to specifications and designs.

➢ Thus, major focus of the white box testing is on internal structures, logic paths, control flows, data flows, internal data structures, conditions, loops, etc.
➢ White box testing is also called glass box testing.





Why to perform white box testing?

➢ To detect and correct logical errors in coding procedural details need to be examined.
➢ To uncover the errors on logical path, white box testing is must.
➢ There are certain typographical errors that remain undetected even after syntax and type checking mechanisms. Such errors can be uncovered during this testing.

1)CONDITION TESTING

To test logical conditions in the program module the condition testing is used. This condition can be Boolean condition or a relational expression.

The condition is incorrect in following situations –
➢ Boolean operator is incorrect, missing, or extra.

- ➢ Boolean variable is incorrect.
- ➢ Boolean parenthesis may be missing, incorrect or extra.
- ➢ Error in relational operator.
- ➢ Error in arithmetic expression.

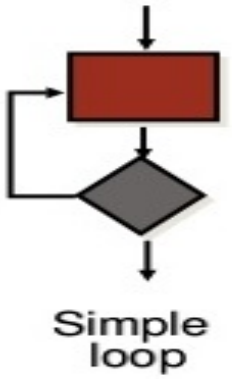
2)LOOP TESTING -

Loop testing is a white box testing technique, which is used to test the loop constructs.

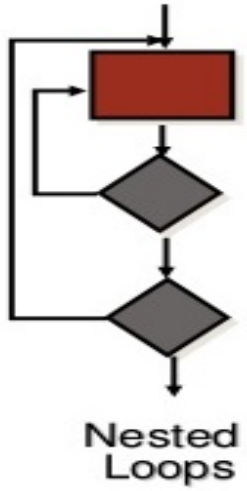
There are four types of loops –

- ➢ Simple loops
- ➢ Nested loops
- ➢ Concatenated loops
- ➢ Unstructured loops

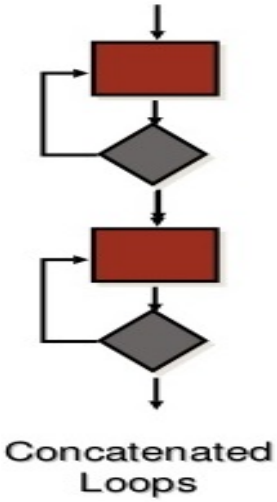
SIMPLE LOOP



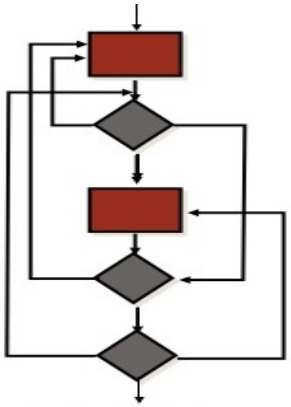
NESTED LOOP



CONCATENATED LOOP



UNSTRUCTURED LOOP



Unstructured Loops

3)PATH TESTING

Path testing is a structural testing strategy.
This method is intended to exercise every independent execution path of a program atleast once.

Following are the steps that are carried out while performing path testing-
Step 1 : Design the flow graph for the program or a component.
Step 2 : Calculate the cyclomatic complexity.
Step 3 : Select a basis of path.
Step 4 : Generate test cases for these paths.

4.5 Comparison between Black Box Testing and White Box Testing

AU : May-04,07,19, Dec.-05,17, Marks 6

Sr. No.	Black box testing	White box testing
1.	Black box testing is called behavioural testing.	White box testing is called glass box testing.
2.	Black box testing examines some fundamental aspect of the system with little regard for internal logical structure of the software.	In white box testing the procedural details, all the logical paths, all the internal data structures are closely examined.
3.	During black box testing the program cannot be tested 100 percent.	White box testing lead to test the program thoroughly.
4.	This type of testing is suitable for large projects.	This type of testing is suitable for small projects.