

## i) **White box testing**

- It is a test case design method that uses the control structure of the procedural design to derive the test cases.
- It tests the internal structure of the software application.
- Knowledge of the program is needed to identify test cases
- White box testing is also called as Clear box testing, open box testing, transparent box testing, glass box testing, structural testing or code- based testing.

### **White Box Testing Techniques**

- i) Basis Path Testing
- ii) Condition testing
  - a) Branch testing
  - b) Domain testing
  - c) Branch and relational operator testing (BRO testing)
- iii) Data Flow Testing
- iv) Loop testing
  - a) Simple loops
  - b) Nested loops
  - c) Concatenated loops
  - d) Unstructured loops

#### **1) Basis path testing**

- Test case designer will derive a logical complexity measure of the design. This design is used as a guide to measure the basis set of execution paths.
- Flow graph represents the control flow.
  - i) Flow graph node- represents the procedural statements
  - ii) Edges or links- represents flow control
  - iii) Regions- Areas bounded by edges or nodes
- Predicate node- Node that contains a condition is called as predicate node and it has two or more edges from it.
- Logical complexity of a program or software can be measured using software metric.
- Cyclomatic complexity is a software metric that provide a quantitative measure of logical complexity of a program.
- Cyclomatic defines the number of independent paths in the basis set of a program
- It provides the upper bound for the number of test conducted to ensure that all statements have been executed at least once.
- 3 ways of computing cyclomatic complexity
  - i) Number of regions of the flow graph represents cyclomatic complexity
  - ii)  $V(G)=E-N+2$ 
    - E- Number of flow graph edges
    - N- Number of flow graph nodes
  - iii)  $V(G)=P+1$ 
    - p- Number of predicate nodes

## 2) Condition Testing

- It is a test case design method that checks the logical conditions present in a program module.
- It tests the each condition in the program
- Simple method
- Other tests performed are
  - i. Branch testing
  - ii. Domain testing
  - iii. Branch and relational operator testing

## 3) Data Flow Testing

- It test paths of the program based on the locations of definitions and uses of variables in the program
- It is useful for selecting test paths for a program containing if and loop statements
- It is effective method for error detection

## 4) Loop Testing

- It focuses on the validity of loop constructs
- It can fix the loop repetition issues
- It can reveal performance issues
- Uninitialized variables in the loop can be determined
- Different classes of loops:
  - i. Simple
  - ii. Nested
  - iii. Concatenated : if two loops are independent of each other then they are tested using simple or nested loops
  - iv. Unstructured loops- It needs restructuring of the design to reflect the use of the structured programming constructs

## ii) **Black Box testing**

- It is a way of software testing in which the internal structure or the program or the code is hidden and nothing is known about it
- It is done by software testers
- No knowledge of implementation is needed
- No knowledge of programming is required
- It is also called as closed testing
- Least time consuming testing method
- Black Box testing method are:
  - i. Equivalent partitioning
  - ii. Boundary Value Analysis
  - iii. Comparison testing
  - iv. Orthogonal array testing

**i. Equivalent partitioning**

- In this method the input domain of the system is partitioned into a number of equivalence classes such that each member of the class works in a similar way

**Two steps:**

- i. Identification of equivalence classes  
Partition of any input domain into two sets: Example valid and invalid values
- ii. Generating test cases
  - To each valid and invalid assign unique values
  - Write test cases covering valid and invalid test cases.

Example: To select the numbers between 1- 50

Valid set: 1-50

Invalid set: 60

**Software Implementation Techniques**

Different testing tools are

- i. Data acquisition tools- To acquire data to be used during testing
- ii. Static measurement tools- To analyse source code without executing test cases
- iii. Dynamic measurement tools- To analyse source code during execution of test cases
- iv. Simulation tools- To simulate function of hardware or other externals
- v. Test management tools- To assist in planning, development and control of testing
- vi. Cross functional tools- that cross the bounds of preceding categories

\*\*\*\*\*