Types of manual Testing

(1)White box Testing

White Box Testing is software testing technique in which internal structure, design and coding of software are tested to verify flow of input-output and to improve design, usability and security. In white box testing, code is visible to testers so it is also called Clear box testing, Open box testing,Transparent box testing, Code-based testing and Glass box testing.

The goal of WhiteBox testing in software engineering is to verify all the decision branches, loops, statements in the code.

White Box Testing Techniques

(a) Statement Testing and Coverage

Statement testing exercises the potential executable statements in the code.

This technique requires every possible statement in the code to be tested at least once during the testing process of software engineering.

• Coverage is measured as the number of statements executed by the tests divided by the total number of Executable statements in the test object, normally expressed as a percentage.

• Code is executed in such a manner that every statement of the application is executed at least once 100% statement coverage doesn’t guarantee 100% branch or path coverage

(b)Branch Testing and Coverage

Decision testing exercises the decisions(eg : if and else

conditions) in the code and tests the code that is executed based on the decision outcomes.
Each branch should be executed at least once for 100% branch coverage
100% branch coverage guarantees 100% statement coverage.
100% branch coverage doesn’t guarantee 100% path Coverage

(c)Path Testing and Coverage

Each path has to be executed at least once for 100% path coverage.Execute all possible control flow paths through the program.100% path coverage guarantees 100% statement and branch coverages

(2) Black Box Testing

Black-box testing is a test approach in which the QA doesn’t have any knowledge about the underlying code or structure of the application. The QA interacts with the software application just like an end-user to test its functional and nonfunctional behavior. This helps to discover some bugs typically overlooked in the earlier stages

Black Box Testing Techniques

(a)Equivalence Partitioning

Equivalence partitioning divides data into partitions (also known as equivalence classes) in such a way that all the members of a given partition are expected to be processed in the same way. There are equivalence partitions for both valid and invalid values.
Valid values are values that should be accepted by the component or system. An equivalence partition containing

valid values is called a “valid equivalence partition.” Invalid values are values that should be rejected by the component or system. An equivalence partition containing invalid values is called an “invalid equivalence partition.” When invalid equivalence partitions are used in test cases, they should be tested individually, i.e., not combined with other invalid equivalence partitions

Example
Let us consider an example of any college admission process. There is a college that gives admissions to students based upon their percentage.
Consider percentage field that will accept percentage only between 50 to 90 %, more and even less than not be accepted, and application will redirect user to an error page.
If percentage entered by user is less than 50 %or more than 90 %, that equivalence partitioning method will show an invalid percentage. If percentage entered is between 50 to 90%, then equivalence partitioning method will show valid percentage.

Equivalence Partitioning		
Invalid	Valid	Invalid
<=50	50-90	>=90

(b) Boundary Value Analysis

Boundary value analysis (BVA) is an extension of equivalence partitioning, but can only be used when the partition is ordered, consisting of numeric or sequential data.

The minimum and maximum values (or first and last values) of a partition are its boundary values partitions.

BVA is also called Range Checking, in this technique, you test boundaries between equivalence partitions.

This technique is generally used to test requirements that call for a range of numbers (including dates and times)

2 way BVA – Boundary Value and value that is just over the boundary by the smallest possible increment are used

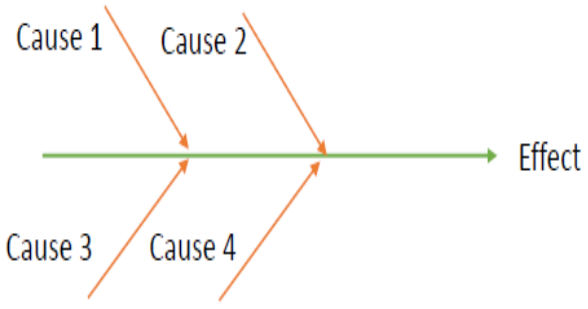
3 way BVA – Values before, on and over the boundary are used.

(c)Cause Effect Graph or Decision Table Testing

Cause Effect Graphing based technique is a technique in which a graph is used to represent the situations of combinations of input conditions. The graph is then converted to a decision table to obtain the test cases. Cause-effect graphing technique is used because boundary value analysis and equivalence class partitioning methods do not consider the combinations of input conditions.

Cause Effect Graph or Decision Table Testing

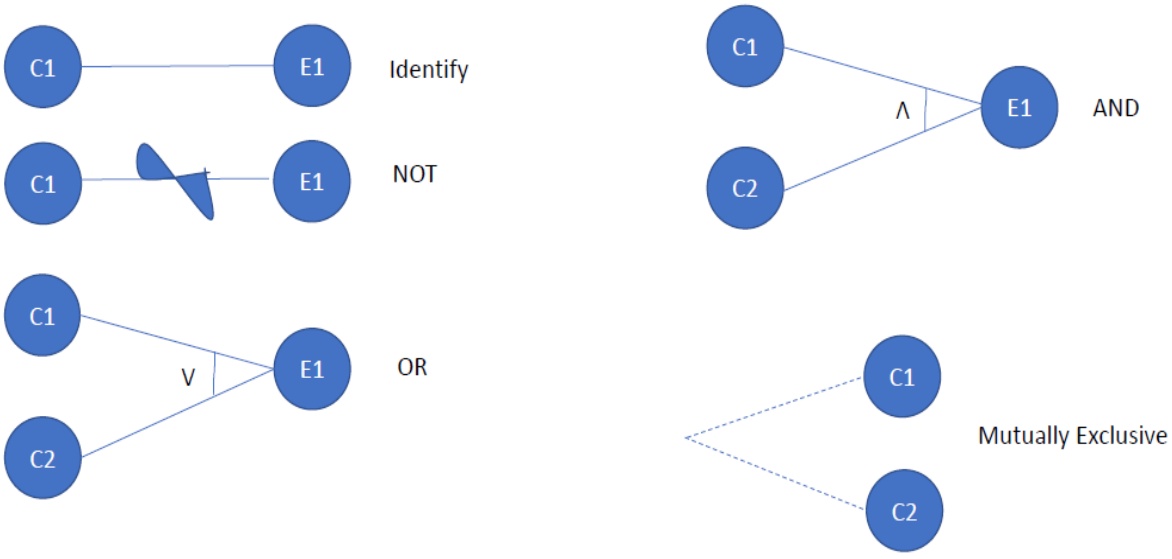
- It graphically shows the connection between a give outcome and all issues or conditions that cause that outcome. It is also known as **Ishikawa** diagram or **fish bone** diagram because the way it looks.
  - **Cause** - input condition that fetches about an internal change in the system
  - **Effect** – output condition, s system transformation or a state resulting from a combination of causes



Steps to Proceed on Test Case design using Cause – Effect Diagram

- Recognise and describe causes and effects
- Build up a cause effect graph
- Convert cause-effect graph into a decision table
- Convert decision table rules to testcases. Each column of decision table represents a test case

Symbols Used in Cause Effect Graph



Identify Function

If C1 = 1, E1 = 1

If C1 = 0, E1 = 0

NOT Function

If C1 = 1, E1 = 0

If C1 = 0, E1 = 1

OR Function

If C1 or C2 = 1, E1 = 1, else E1 = 0

AND Function

Both C1 and C2 = 1, E1 = 1, else E1 = 0

Mutually Exclusive

When only one of the cause will hold true

Advantages

It helps to determine the root causes of a problem or quality.

It indicates possible causes of variation in a process.

It identifies those areas where data should be collected for further study.

It utilizes the team knowledge of the process by encouraging team participation.

(d)State Transition Testing

• State transition testing is used for menu-based applications and is widely used within the embedded software industry.
• The technique is also suitable for modeling a business scenario having specific states or for testing screen navigation.
• A state transition table shows all valid transitions and potentially invalid transitions between states, as
Well as the events, and resulting actions for valid transitions.
• State transition diagrams normally show only the valid transitions and exclude the invalid transitions.
• Tests can be designed to cover a typical sequence of states, to exercise all states, to exercise every transition, to exercise specific sequences of transitions, or to test invalid transitions .

(e)Use Case Testing

Use Case Testing is a software testing technique that helps to identify test cases that cover entire system on a transaction by transaction basis from start to end. Test cases are the interactions between users and software application.
Use case testing helps to identify gaps in software application that might not be found by testing individual software components.

Negative Testing

Negative Testing is a software testing type used to check the software application for unexpected input data and conditions.

Unexpected data or conditions can be anything from wrong data type to strong hacking attack. The purpose of negative testing is to prevent the software application from crashing due to negative inputs and improve the quality and stability.

Example of Negative Testing

Consider the case of a lift which is a commonly considered example of negative testing.

We all know the functionality of a lift. These will be considered as the requirements of a lift like pressing the floor number make the lift go to that particular floor.

The door opens automatically once the lift reaches the specified floor and so on.

Now let's consider some negative scenarios for lift. Some of them are,

Negative Testing	Positive Testing
• What happens if the number of persons (weight) exceeds the specified limit?	• Assumes the only specified number of persons will enter the lift
• What happens if someone smokes or cause a fire inside the lift?	• There won't be smoke or fire inside the lift
• What happens if there is a power failure during operation?	• There won't be a power failure during the working of the lift