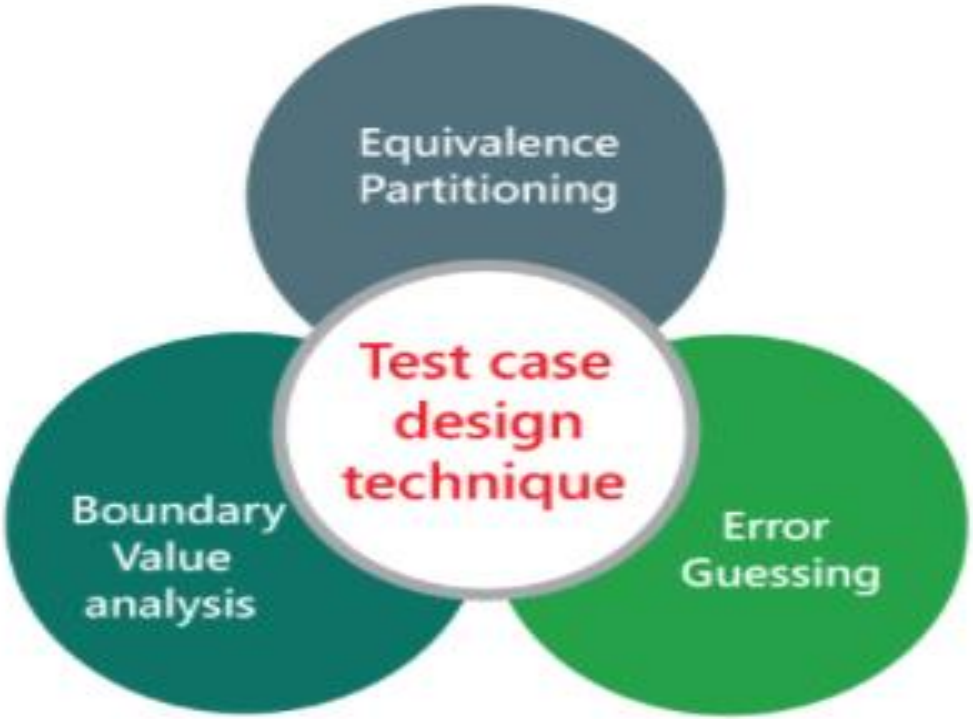
White Box Techniques

- Data flow Testing
- Control flow Testing
- Branch Coverage Testing
- Statement Coverage Testing
- Decision Coverage Testing

Black Box Techniques

- Decision Table
- All-pair Testing
- Cause-Effect Testing
- State Transition
- Use Case

Test CaseTechniques



Error GuessingTechniques

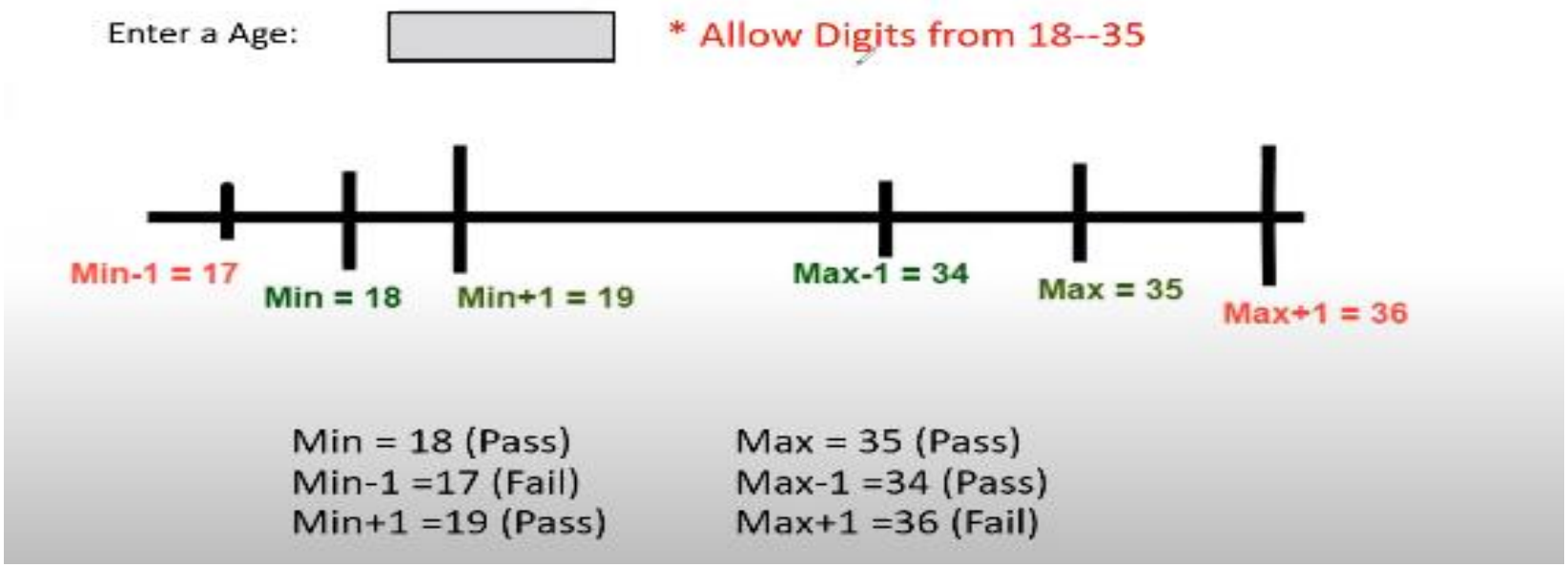
- It is used to find the bug in the application based on the testers prior experience.
- We don't follow any specific rule
- It depends on the Tester Analytical skill and experience.

Example: we have to deposit some money in bank account, but the amount will be accepted on a particular range of which is 5000-7000.

Value	Decription
6000	Accepted
5555	Accepted
4000	Error message
8000	Error message
blank	Error message
100$	Error message

Boundry value Analysis

- It is one of the widely used testing technique.
- It is used to test boundaries of the input values.
- **Example:** Age = 18-35



Equivalance Class PartitioningTechniques (ECP)

- It is a technique of software testing in which input data is divided into partitions of valid and invalid values/class.
- We can select the data according to the class.
- It reduces the number of test-cases and saves the time

Exar

Enter a Number: [ ] * Allow Digits from 1--500

Normal Test Data	Divide values into Equivalence Classes	Test Data using ECP
1		
2		
3	-100 to 0 → **-50** (Invalid)	-50
4	1 – 100 → **30** (Valid)	30
.	101 – 200 → **160** (Valid)	160
.	201 – 300 → **250** (Valid)	250
.	301 – 400 → **320** (Valid)	320
.	401 – 500 → **450** (Valid)	450
500	501 – 600 → **550** (Invalid)	550

ECP & BVA

- **Requirement:-**
  - User name field allows only lower case with min 6 max 8 letters.

ECP for User Name

Valid	Invalid
a….z	A….Z 0…9 Special characters (@ ,#,$,& etc..)

BVA for User Name

Parameters	Value	Result
Min	6	Valid
Min+1	7	Valid
Min-1	5	Invalid
Max	8	Valid
Max+1	9	Invalid
Max-1	7	Valid

Bug/Defect

- It means that software or application is not working as per the requiremet .
- The bug occurred when developers made any mistake or error while developing the product.

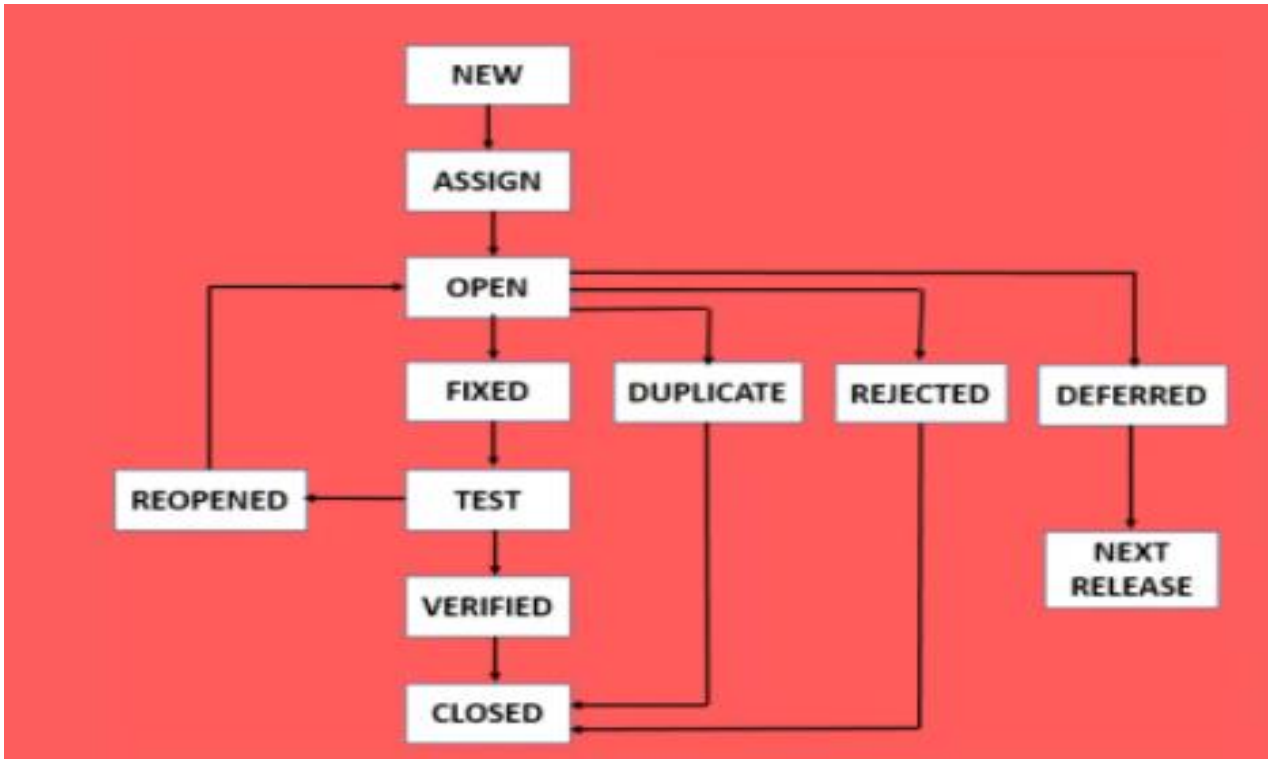
Terms	Description	Raised by
Defect	When the application is not working as per the requirement.	Test Engineer
Bug	Informal name of defect	Test Engineer
Error	Problem in code leads to the errors.	Developer, Automation Test Engineer
Issue	When the application is not meeting the business requirement.	Customer

Bug tracking tool

- Jira
- Bugzilla
- Redmine
- Mantis
- Backlog

Bug/Defect Life Cycle

- **The bug life cycle** is also known as the **Defect life cycle**.
- The bug life cycle varies depends upon the tools (Backlog, JIRA, etc.,) used
- The bug starts when the defect is found and ends when a defect is closed.



Bug/Defect Life Cycle

#1. New

- When a tester finds a new defect.
- He should provide a proper Defect document to the Development team to reproduce and fix the defect.
- In this state, the status of the defect posted by the tester is "New"

#2. Assigned

- Defects that are in the status of New will be approved (if valid) and assigned to the development team by Test Lead/Project Lead/Project Manager.
- Once the defect is assigned then the status of the bug changes to "Assigned"

#3. Open

- The development team starts analyzing and works on the defect fix

#4. Fixed

- When a developer makes the necessary code change and verifies the change, then the status of the bug will be changed as "Fixed" and the bug is passed to the testing team.

#5. Test

- If the status is "Test", it means the defect is fixed and ready to do test whether it is fixed or not.

Bug/Defect Life Cycle

#6. Verified

- The tester re-tests the bug after it got fixed by the developer. If there is no bug detected in the software, then the bug is fixed and the status assigned is “verified.”

#7. Closed

- After verified the fix, if the bug is no longer exits then the status of the bug will be assigned as “Closed.”

#8. Reopen

- If the defect remains the same after the retest, then the tester posts the defect using the defect retesting document and changes the status to “Reopen”. Again the bug goes through the life cycle to be fixed.

#9. Duplicate

- If the defect is repeated twice or the defect corresponds to the same concept of the bug, the status is changed to “duplicate” by the development team.

Bug/Defect Life Cycle

#10. Deferred: In some cases, the Project Manager/Lead may set the bug status as deferred.

- If the bug found during the end of the release and the bug is minor or not important to fix immediately.
- If the bug is not related to the current build.
- If it is expected to get fixed in the next release.
- The customer is thinking to change the requirement.
- In such cases the status will be changed as “deferred” and it will be fixed in the next release.

- **#11. Rejected**
- If the system is working according to specifications and the bug is just due to some misinterpretation (such as referring to old requirements or extra features) then the Team lead or developers can mark such bugs as “Rejected”