**Manual Testing Class notes**

**Software** : A software is collection of computer programs that helps us to perform a task.

**Types of Software**

1. Application Software

ex. Web Application, Mobile Apps, Desktop Application

2. System Software

ex. Device Drivers, Operating Software, Servers, Utilities

3. Prgramming Software

ex. compilers, debuggers, interpreters.

**Software Testing**

Software testing is part of Software Development Process.

It is an activity to detect and identify the defects in the software.

The objective of testing to to release quality product to the client.

**Software Quality**

Bug Free

Deleivered in Time

Within Budget

Meets requirements and expectations

Maintainable

**Project vs Product**

If software application is developed for a specific customer based on the requirements then it is called Project.

If software application is developed for multiple customersbased on the market requirement it is calld as Product.

**Error , Bug and Failure**

**Error** : Error is an incorrect human action or mistake. This is mainly identified by the developers

Ex. some mistake while writing the codes

**Bug/ Defect** : The deviation between actual and expected behaviour of the software is called bug/Defect

ex. If we are using any application and if there is some difference between the actual and expected thing

**Failure**: The deviation identified by the enduser is called Failure.

Ex. once we have developed the product and the custtomer is using the product but it gets failed is called failure.

**Why the software has bugs??**

Miscommunication or no communication

Software complexity

programming error

chnaging requirements

Lack of skilled Tester

**Questions Till Now**

What is Software? Types of Software’s?

What is Software Testing?

What is Software Quality?

Project Vs Product

Why do we need Testing?

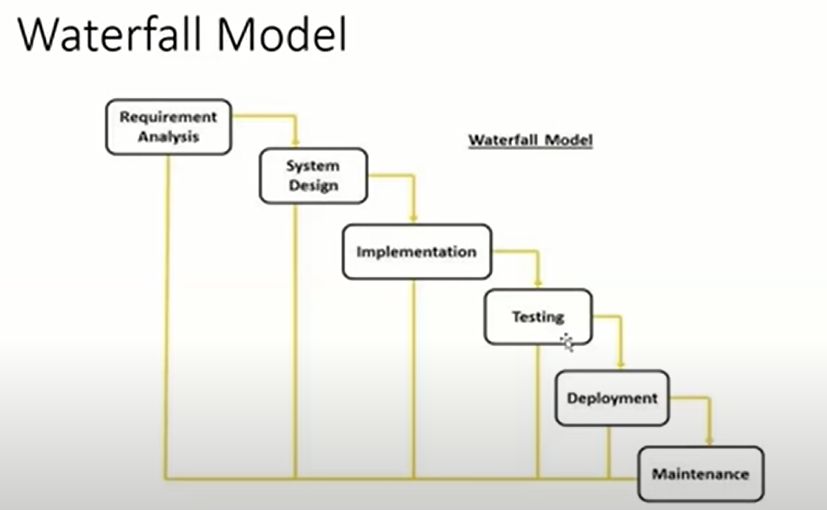
Error, Bug & Failure

Why the software has bugs?

**SDLC : Software Development Life Cycle**

SDLC is a process used by the software industry to design develop and test softwares.

**Waterfall Model**

****

**Advantages**

1. Quality of Product will be good

2. Since Requirement changes are not allowed, chamces of finding bugs will be less

3. Initial investment is less since the testers are hired at the later stages.

4. Preffered for smaller pojects where requirements are freezed.

**Disadvantages**

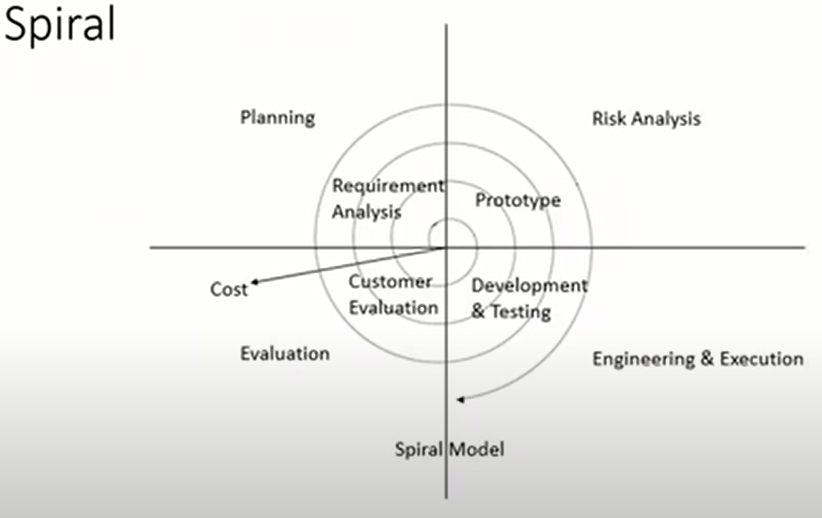
Requirement changes are not allowed.

If there is defect in requirement than it will be continued in later phases

Total insvestment is more as time taking for rework on defect is time consuming which leads to high investment

Testing will start only after coding.

**Spiral Model**



Spiral model is Iterative model.

Spiral Model overcome drawbacks of Waterfall Model

We follow spiral model where there is dependency on the modules.

In every cycle new software will be released to the customer.

Software will be released in multiple version. So it is called version control model.

**Advantages**

Testing is done in every cycle, before going to next cycle.

Customer will get to use the software for every module.

Requirement changes are allowed after every cycle before Going to Next cycle

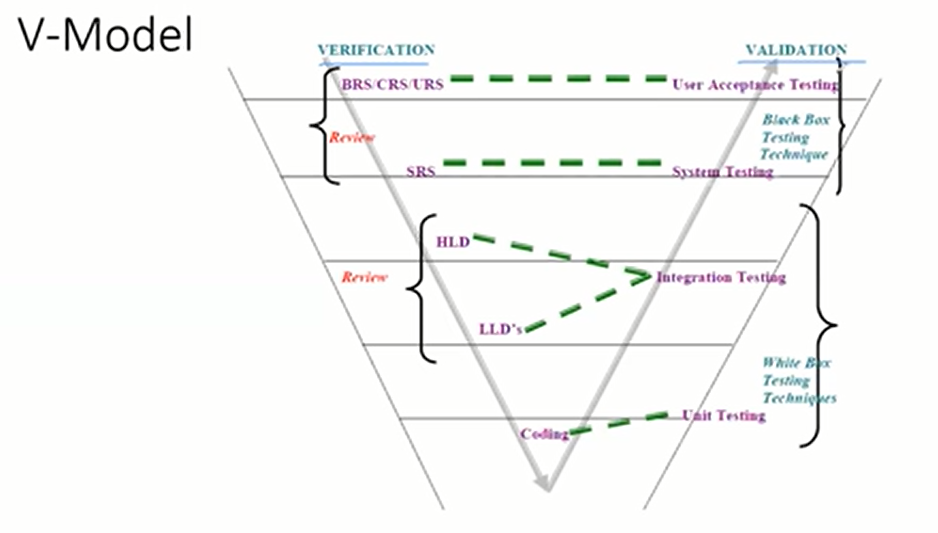
**Disadvantages**

Requirement changes are not allowed in between the cycle.

Every cycle of spiral model looks like waterfall model.

There is no testing in requirement and design phase.

**V Model**

****

**Verification vs Valiadtion**

**Verification**

Verification checks whether we are building the right product.

Focus on Documentation

Verification involves

Reviews

Walkthrough

Inspection

**Validation**

Validations checks whether we are building the product right.

Takes place after verification is completed

Focus on Software

Validation typically involves actual testing like Unit Testing, Integration testing, System testing and Acceptance Testing

**Advantages**

Testing is involved in each and every phase.

**Disadvantages**

Documentation is more.

Initial investment is more.

**Static Testing**

Testing the project related documents is called static testing.

**Techniques**

Reviews

Walkthrough

Inspection

**Reviews** -- We genrally conduct on documents to ensure the correctness and completeness.

**Types of Reviews**

Requirement Review -- BRS, SRS

Design Review ---- HLD, LLD

Code Review ---- Coding

Test Plan Review ---

Test Case Review ---

**Walkthrough**

It is a informal review

Author reads the documents or codes and discuss with peers(collegues)

It not planned and can be done anytime.

**Inspection**

Its the most formal review

In this atleast 3- 8 people are involved reader , writer , moderator

Inspection will be having a proper schedule.

**Dynamic Testing**

Tetsing the actual software is called dynamic testing

**Techniques**

Unit Testing

Integration Testing

System Testing

User Acceptance Testing

**Unit Testing**

A unit is a single component or module of a software.

Unit Testing conduct on a single program or single module.

Unit Testing is a white Box testing Technique.

Unit Testing is conducted by the developer.

**Unit Testing Techniques**

1. Basis Path Testing : Every path in the testing should be considered

2. Control structure Testing : that we have the wrtten codes and we know what is going to be the result

3. Conditional Coverage : If else conditions or any conditional statements

if A =3

Print 'A'

Else

Print 'B'

4. Loop Coverage : If we have various loops in the testing

if A =3

Print 'A'

Else

Print 'B'

And

if B>=4

print A

else print B

5. Mutation testing -- when we test the codes by passing various inputs

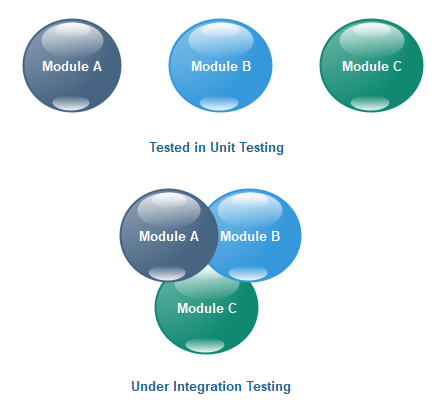
**Integration Tetsing.**

-------------------

Integration Testing is performed between 2 or more module.

Integration testing focuses on checking data communication between two or more modules

Integration testing is a type of White Box testing



**Types of Integration Testing**

1.Incremental Integration testing

2.Non Incremental Integration testing

**Incremental Integration Testing**

Icrementally adding the module and testing the data flow between the modules

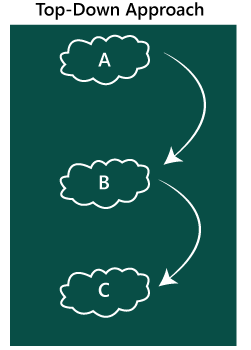
2 types of Approaches

1. Top Down Approach

2. Bottom Up Approach

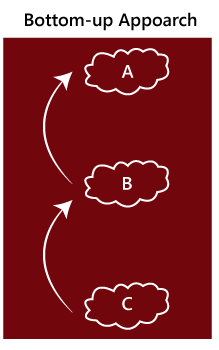
**1. Top Down Approach**

Incrementally adding the modules and testing the data flow between the modules and ensure that the module added is the child of of previous module



**2. Bottoms Up Approach**

Incrementally adding the modules and testing the data flow between the modules and ensure that the module added is the parent of of previous module.

****

**3. Sandwitch Approach**

Combination of Top Down and Bottoms up approach

**NON -Incremental Integration Testing**

Adding all the modules in a single shot and test the data between the modules.

**Drawback of Non Incremental Integration testing**

1. We might miss data flow between some of the modules.

2. If you find any defect we cant understand the root cause of the defect.

**System Testing**

Testing overall functionality of the software with respect to the client requirement.

It is Black box technique

This testing is conducted by testing team.

After completeting components and integration testing we start System testing

Before conducting sysytem testing we should know the customer requirement

**System testing mainly focuses on below aspect**

> User Interface testing(GUI) - Graphical User Interface

> Funtional testing

> Non Funtional testing

> Usability Testing

**User Interface testing (GUI)**

Graphical User interface testing is process of testing the user interface of the application.

A graphical user interface includes all the elements like menus,checkbox, buttons, color, fonts, icon, content and images.

**GUI testing Checklist**

Testing the size, position, width and height of the elements.(we will be able to get the details from the design documents)

* Testing the error, messages, that are getting displayed.
* Testing the diffferent section of screen.
* Testing the font whether it is readable or not.
* Testing the of screen in different resolution with the zoooming and zooming out
* Testing of allignment of the the text and other elemts like icon, imagesand other things.
* Testing the color of the font.
* Testing whether the image has good clarity or not.
* Testing of spellings.
* The user must not get frustrated by using the system interface.
* Testing whether the interface is attractive or not.
* testing the scrollbar according to the size of page if any.
* Testing the disabled fields if any.
* Testing the size of images.
* Testing the headings whether it is properly allinged or not.
* Testing the color of the hyperlink.
* Testing UI elements like buttons, testxtbox, text area, checkbox, buttons, dropdown, links,etc.

**Usability Testing**

During this testing we validates application provided context sensitive help or not to the user.

Checks how easily the enduser are able to understand and operate the application is called Usability testing.

Eg. User manual should be prepared for the customer help to use the application and as a user we should be able to use the appilcation easily.

**Functional Testing**

Funtionality : the behaviour of the application i.e it is working as per the customer requirement.

Functional testing talks about how the feature of the application should work.

As part of Functional testing we generally do below testing.

* Object Properties Testing
* Database Testing
* Error Handling
* Calculations/Maninulation Testing.
* Links Existence and Links Execution
* Cookies and Sessions

**Object Properties Testing**

Checks the properties of object present on the Application.

Ex. Enable ,disable , visble and Focus.

**Database Testing/Backend Testing**

Checking database operations wrt to user operations.

Genarally we do the database testing with the help of sql

DML operations - Data manipualtion language

insert

update

delete

Database testing is the type of grey box testing technique.

various things we do as database tester

Table Level validation

Relation between the tables

Funtions

triggers

procedures

indexes etc.

**Error Handling Testing**

Testers verifies the error messages while performing incorrect action on the applications

Error messages should be readable which is it should be understandable by users

ex. Invalid user , incorrect data

**Calculation/ Maninulation Testing.**

Testers should verify the calculations.

**Links**

Where exactly the links are placed -- link Existence

Links are nevigating to proper page or not.

Internal Links

External Links

Broken links

**Cookies and Sessions Testing**

**Cookies** : Temprory files created by browser while browsing the page through internet.

**Sessions**: Sessions are time slots created by servers. Sessions will be expired after sometime.(If you are idle for sometime)

**Non Functional Testing**

Once the application functionality is stable then we do Non Functional testing.

Generally focuses on performance, load it can take, security etc,

**Types of Non Functional Testing**

* Performance Testing
  + Load testing
  + Stress Testing
  + Volume Testing
* Security testing
* Recovery Testing
* Comaptibility testing
* Configuration Testing
* Installation testing
* Sanitation/ Garbage Testing

**Performance Testing**

Performance -- Speed of the Application

* Load testing
* Stress Testing
* Volume Testing

**Load** : Gradually inreasing the load on the application slowly then check the speed of the application.

**Stress** : Suddenly incresing/decreasing the load on the application and check the speed of application.

**Volume** : Check how much data is able to handle by the application

Tools used for performance testing are like Jmeter, Loadrunner.

**Security Testing**: How secure the application is

Authentication : Users who are using the application is valid or not.

Autherization/Access Control : Verifying the permissions of valid users.

**Recovery Testing:**

Recovery testing generally testing whether application is overcoming from abnormal situation to normal.

**Comaptibility Testing**: Compatibility Testing is a type of Software testing to check whether your software is capable of running on different hardware, operating systems, applications, network environments or Mobile devices.

* **Forward Comaptibility** : A process to verify the behavior and compatibility of the developed hardware or software with the newer versions of the hardware or software.
* **Backward Comaptibility**: A technique to verify the behavior and compatibility of the developed hardware or software with their older versions of the hardware or software.
* **Hardware Comaptibility** : It checks software to be compatible with different hardware configurations.

**Installation Testing**: Testing the installation process of any software

Check screens are clear to understand

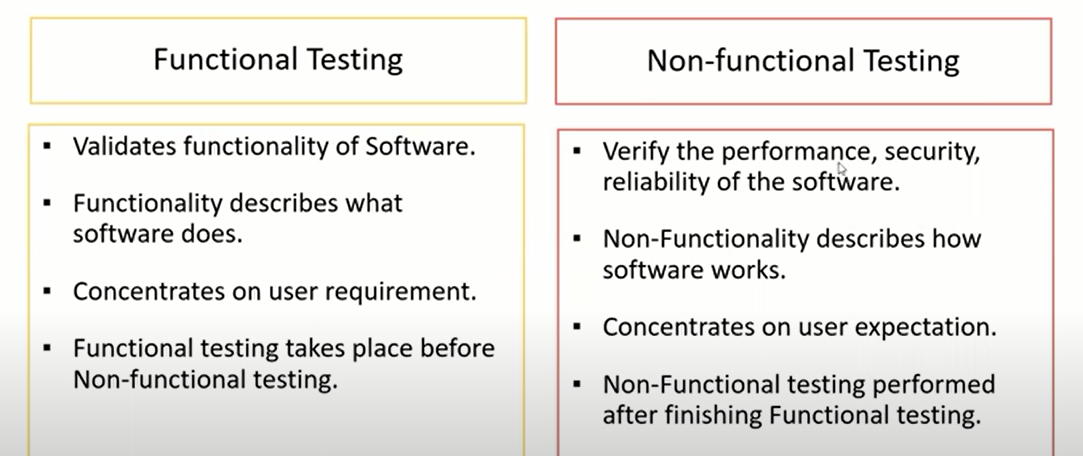
Screen navigation

Simple or Not

Uninstallation process is easy

**Garbage/Sanitation Testing**: If any application provides extra features/functionality than we consider them as bug.

**Difference Between Functional and Non- Functional Testing**



**Regression Testing**

Testing conducts on modified build to make sure that there will be no impact on existing functionality

because of changes like adding deleting or modifying any feature.

**Unit Regresssion testing**

Testing only the changes/modifications done by the developer.

**Regional Regression Testing**

Testing the modified module along with the impacted module

Impact analysis meetings conduct to identify impacted modules with qa and dev

**Full Regression testing**

Testing the main feature and remaining part of the application.

ex. Dev has changes in many modules, instead of identifying impacted modules, we perform one full round of regression.

**Retesting**

Whenever the developer fixed a bug, tester will test the bug fix is called retesting.

Tester close the the bug if it worked otherwise reopen and send to developer.

To ensure that the defects which were found and posted in the earlier build were fixed or not in the current build.

**Smoke Testing**

Somke test is done to make sure the build we received from development team is stable or not.

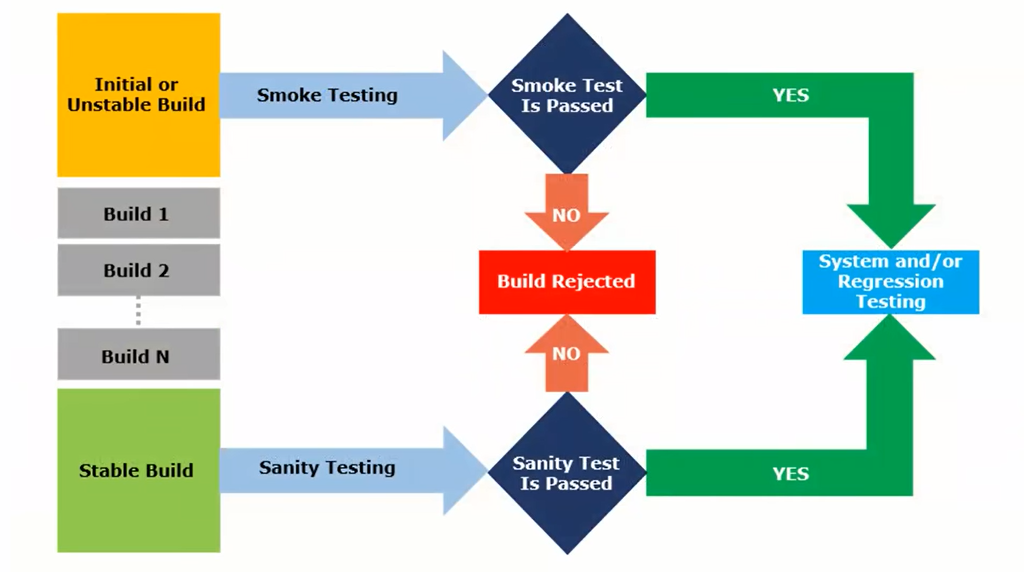
Smoke test is done by both tester and developers.

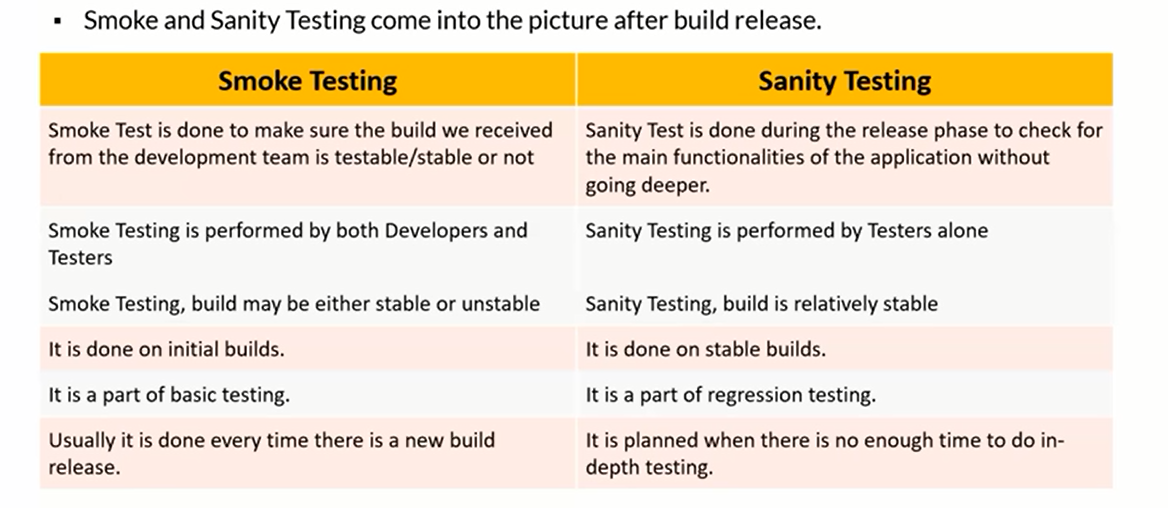
Smoke Testing, build may be either stable or unstable

**Sanity testing**

Sanity testing is done during the release phase to check for the main functionality of application without going deeper. Sanity Testing the buils is stable.

Sanity test is done by testers only



****

**Exploratory testing**

We have to explore the application, understand completely and test it.

Understand the application,identity all possible scenarios, document it and use it for testing.

We do exploratory testing when the Application ready but there is no requirement.

Test Engineer will do exploratory testing when there is no requirement.

**Drawbacks**

You might misunderstand any feature or bug any bug as feature since you do not have any requirement

Time Consuming

If there will be any bug in the sysytem you wil never know it

**Adhoc Testing**

Testing application randomly without any test cases or BRS document.

Adhoc Testing is an informal testing type i.e we can do this testing anytime, dnt need any particular system.

Tester should have the knowledge of application even though we dnt have document.

This is unplanned

**Monkey/Gorilla Testing**

Testing the application randomly and anywhere without having the knowledge of requirements.

Tester do not have the knowledge of application

Suitable for gaming application .

**Positive Testing**

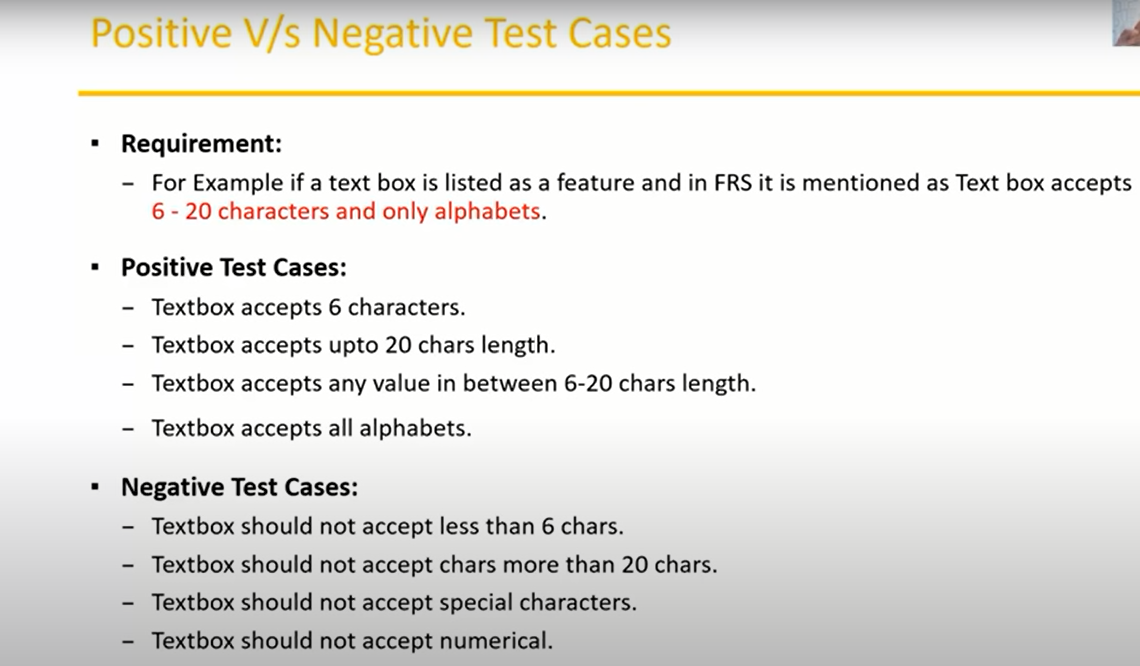
Testing application with valid inputs is called Positive testing.

It Checks whether the application behaves as expected with positive inputs.

**Negative Testing**

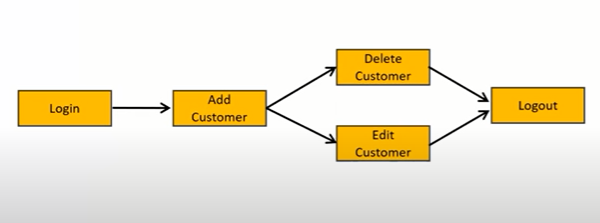
Testing application with invalid inputs is called Negative testing.

It Checks whether the application behaves as expected with Negative inputs.



**END to End testing**

Testing the overall functionality of the system including the data integration among the modules is called end to end testing.



**Globalisation Testing**

Performed to ensure the system or software application can run in any cultural or local environment.

Different aspects of the software are tested to ensure that it supports every language and different attributes.

It tests the different currency formats , mobile formats and address formats are supported by the application.

Ex. facebook supports many languages and it can be accused by people of different countries. Hence it is Globalised product.

**Localization testing**

Perform to check system or software application for a specific geographical and cultural environment.

Localized product only supports the specific kind of language and is usable only in specific region.

It test the specific currency formants , Mobile number format and address format is working properly or not

Ex. baidu.com is a chinese app which supports inly chineses language.

**Test Design Techniques**

Used to prepare data for testing.

1.Reduce the Data

2. More Coverage

**Types of test design Techniques**

1. Equivalence Class Partitioning

2. Boundary Value Analysis

3. Decision table based testing

4. State Transition

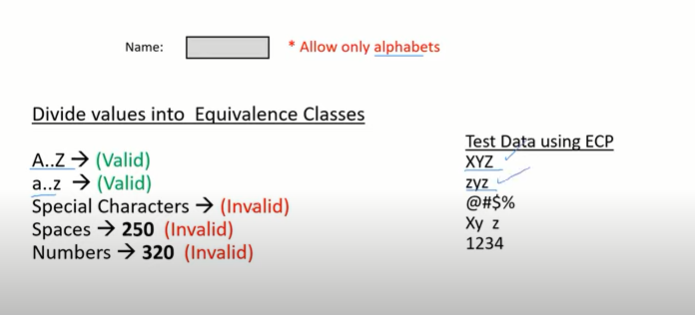
5. Error Guessing

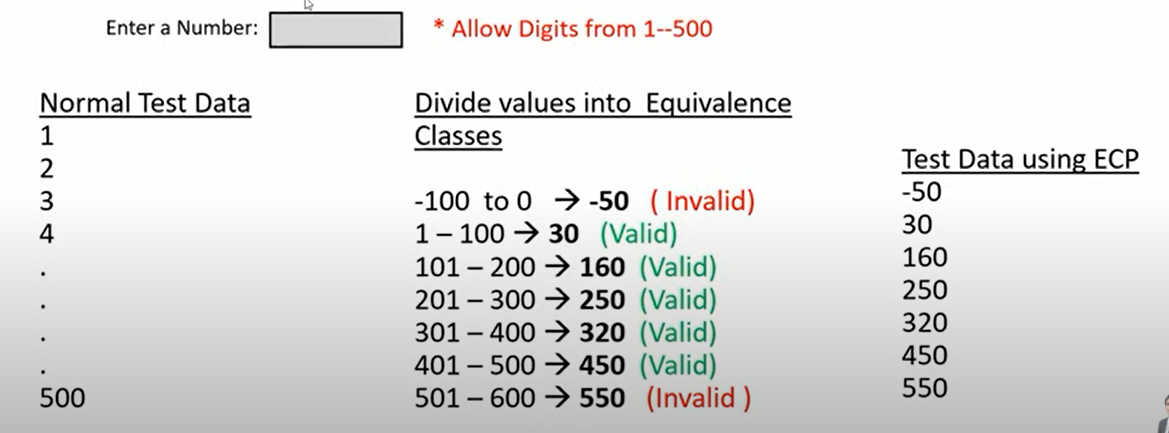
**1. Equivalence Class Partitioning**

Partition data into various classes and we can select data according to class then

test. It reduces the number of test cases and saves time for testing.

Examples:





Classify/divide/partition data in to multiple classes.

**2. Boudary value analysis(BVA)**

BVA technique is used to check the boundary of input.

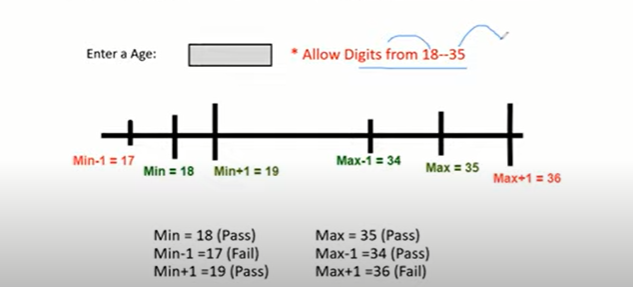
only six parameters will be considered

min max

min+1 max+1

min -1 max-1

Examples:



Note : Mainly BVA and ECP test technique is used in input domain testing.

**Input Domain Testing**

The valye will be provided in the text box/input field

**3. Decision Table**

Decision table is also called as Cause- Effect tables

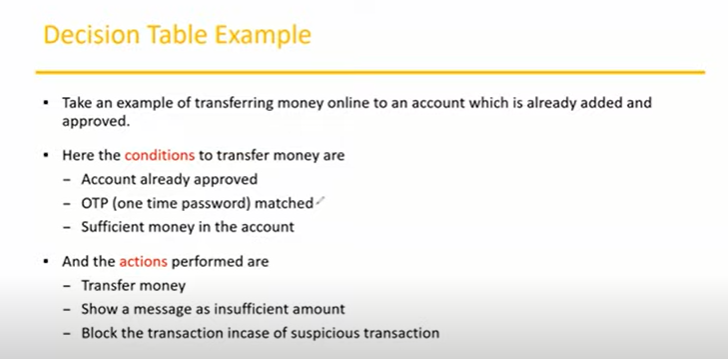
This technique is used if we have more conditions and corresponding actions.

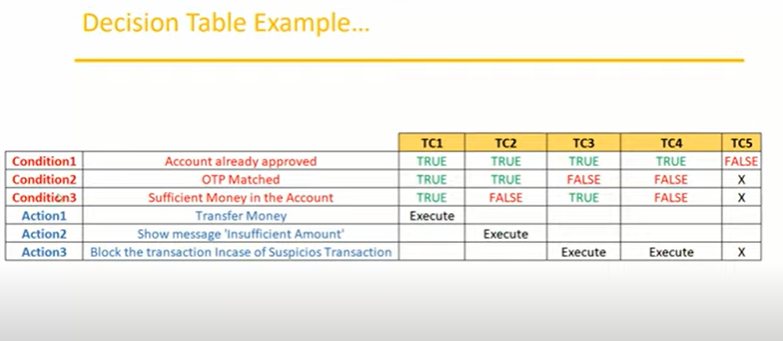
In decision table we deal with combination of inputs.

To identify the test case with decision table, we consider conditions and actions.

Examples:

**Requirement**





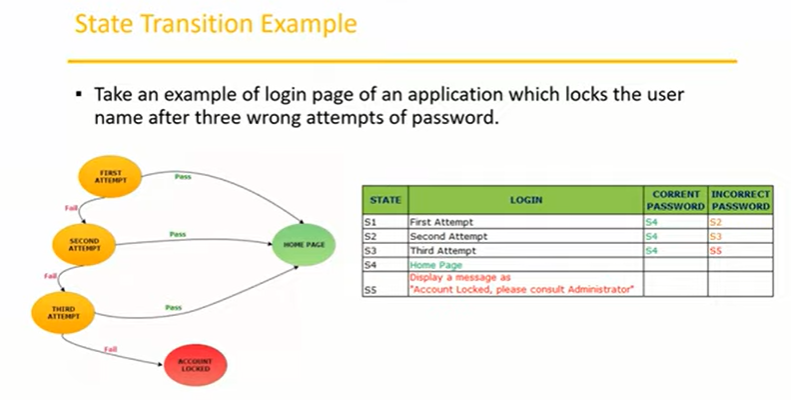
**5. State Transition**

In state transition technique changes in input conditions change the state of application.

This testing technique allows the tester to test the behaviour of AUT

The tester can perform this action by entering various input conditions in a sequence.

In state transition technique tester provide negative as well as negative input test values for evaluation of system behaviour.

****

**5. Error Guessing**

>Error guessing is one of the testing technique used to find the bugs in a software application based on tester prior experience.

> In error gussing we do not follow any specific rule.

>It depends on testers analytical skils and experience.

ex. Submiting a form without entering values

entering invalid values such as entering alphabet in the numneric field.

**STLC - Software Testing Life Cycle**

1. Requirement Analysis

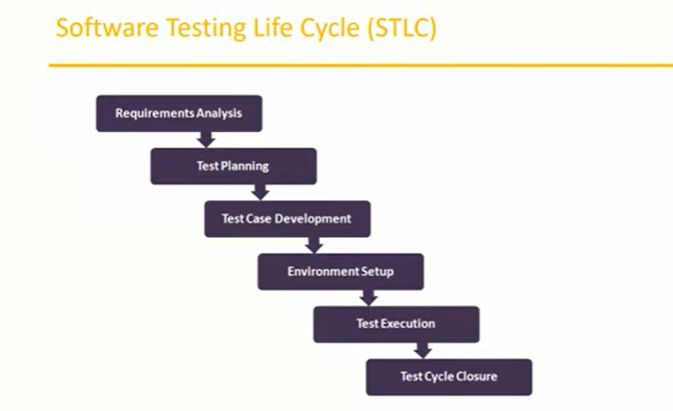
2. Test planning

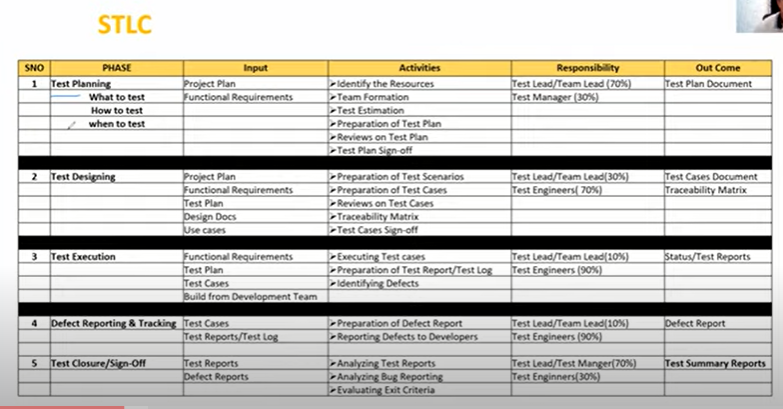
3.Test Design

4. Test Execution

5. Defect/Bug Tracking

6. Test Closure





**STLC - Software Testing Life Cycle**

1. Requirement Analysis

2. Test planning

3.Test Design

4. Test Execution

5. Defect/Bug Tracking

6. Test Closure

**Test Plan Document Content**

A test plan is a document that describes the test scope, test startegy, objectives, schedule, deliverables and resources required to perform testing of software product.

**Test Plan templates contents**

1. Overview

2. Scope

> Inclusion > Test Environment > Exclusion

3. Test Strategy

4. Defect Reporting procedure

5. Roles/Responsibility

6. Test Schedule

7.Test Deliverables

8. Pricing

9. Entry and Exit Criteria

10. Suspension and Resumption Criteria

11. Tools

12. Risks and Mitigation

13. Approvals

**Use Case**

Use Case describes the requirement which will help us to understand the requirement more clear.

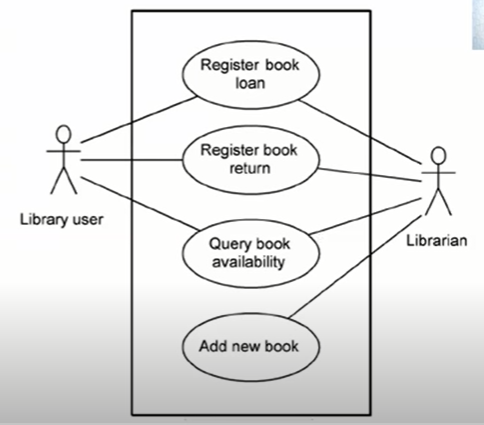
Use case contains THREE items

- Actor which is the user, which can be a single person or group of people, interacting with a process.

- Action, which is to reach the final outcome.

-Goal/Outcome which is successful user outcome.

Project managers or BA writes the usecases.



**Test Scenario**

A possible area to be tested(what to test)

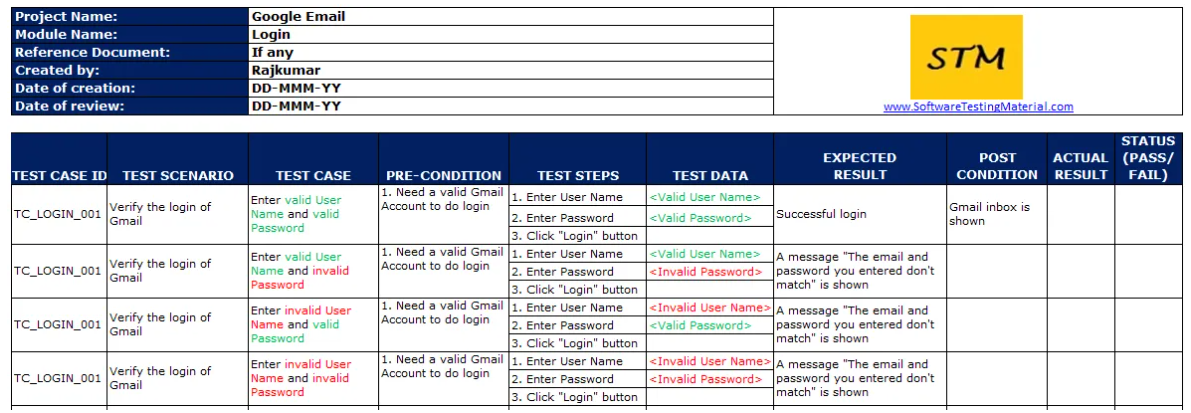
**Test Case**

Step by step action to be performed to validate the functionality of AUT(How to test)

Test case contains test steps, expected and actual results

Mainly prepared by the Test Engineers

A test Case is a set of actions executed to validate the particular feature or functionality of the software application.



**Test Case Content**

Test case ID

Test Scenario

Test Case Title

Precondition

Test Steps

Priority

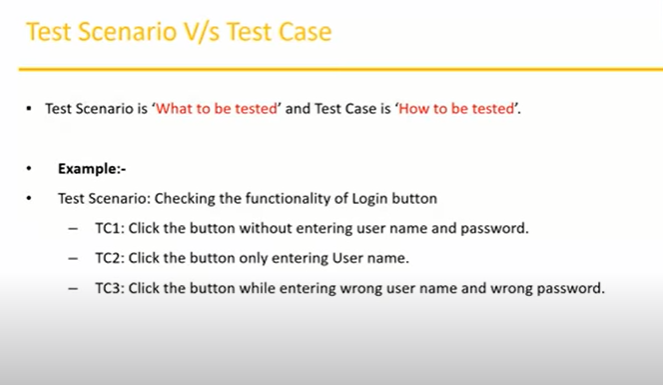
Test Data

Expected Result

Actual result

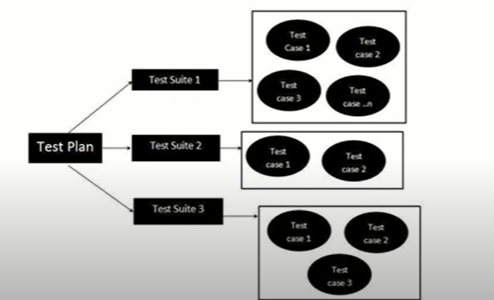
Status({Pass/fail)

**Test scenario vs test case**

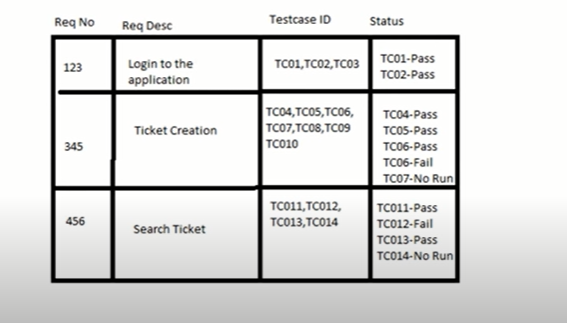
****

**Test Suite**

A group of test cases which belongs to same category is called test suite.

****

**Requirement Traceabilitity Matrix(RTM)**



RTM describes the mapping of requirements with the test cases

The main Purpose of RTM is to see that all the test cases are covered so that no functionality should miss while testing.

It mainly includes the

Requirement ID

Requirement Description

test case ID

**Test Environment**

Test environment is a platform specially build for test case execution on the software product

It is created by integrating the required software and hardware along with proper network configuration

Test Environment stimulates production/real time environment.

Another name for Test environment is Test BED.

**Test Execution**

During this phase test team will carry out the testing based on the test plans and the test case prepared.

Entry Criteria: Test case, test data, test plan

Activities:

- Test Cases are executed based on the test planning.

- Status of test cases are marked like Passed, Failed , Blocked, Run and others

- Documentation of test results and log defects failed cases is done.

- All blocked and failed test cases are assigned bug ids.

- Retesting once the defect is fixed.

- defetcs are tracked till closure.

Deliverables: Provided defect and test case execution report with completed result.

**Guideline for Test Execution**

> The build being deployed to the QA environment is the most improtant part of test execution cycle.

> Test execution is done in QA environmaent.

> Test Execution happens in multiple cycle.

> Test Execution phase consists Executing the Test cases + test scripts

**Defects/Bugs**

> Any mismatched functionality found in a application is called as bug/defect.

> During test Execution Test Engineers are reporting mismatched defects to developers templates or using Tools.

**Defect reporting tools**

Quality Center

Jira

Bug Jilla

Clear quest.

**Defect Classifications**



