**What is software testing?**

**Ans:**software testing is a part of development process it is an activity to identify the defects in the software,objective of testing is to release a quality product.

**Software quality?**

Quality is defined as justification all the requirements of the customer in a product

Quality is not defined in the product it is defined in the customer mind

Bug free

Delivery on time

With in a budget

Meets requirement and expecting maintenance

**What is project vs product**

If the software application is developed for the specific customer based on requirements is called project

If the software application developed for the multiple customer based om market requirement then it is called product

**Why do you need testing**

Bug free

To increase the quality

To meets the customers requirements

Fixing the bugs after release is more expensive

**Error:-**any incorrect human action that produces in the system is called error

**Defect/bug:-**deviation from expected behaviour to actual behaviour is called defect

**Failure:-**the defect found by the end user is called failure

**why the software has bugs normally?**

Miscommunication or no communication

Software complexity(software properties that effects internal interactions)

Lack of knowledge in testers

Error in programming

Changes of requirements in last minute or in the middle of testing or deploy phase

**Software development life cycle {SDLC}**

Software development life cycle is used by software industry to design develop and test the software.

Requirement analysis

Design

Development

Testing

Deploy

Maintenance

**Waterfall model**

This is old traditional model that we have develop the software first later on we have to test and deploy

**Advantages**

The product quality Is good

New requirements and changes are nit allowed so bugs will be less

Low investment

Preferred small projects

**Disadvantages**

Changes are not allowed in between the development or testing phase

Defect need more time to solve

Investment is high because the down time will be happen if the project is deployed or under maintenance

Testing has to be start after development only.

***The goal of a software tester is to find bugs, find them as early as possible, and make***

***sure they get fixed.***

**Why testing is necessery?**

Any level of testing cannot declare there is no defect in the product

Developer people assume that whatever they developed is right and they think its always work but in the real time scenario under take actual execution each level of software building including system level to check its actually working or not

The primary role Of software testing is not to demonstrate the correctness of

software product, but to expose hidden defects so that they can be fixed. Testing is done to protect the common users from any failure of system during usage.

Testing is a process Of demonstrating that errors are not present in the product

This approach is used in acceptance testing where if the application meets acceptance criteria, then it

must be accepted by the customer,

Testing gives number of detects present which indirectly gives a measurement of quality.

More number of defects Vindicate bad software and bad processes Of development,

The software doesn’t do something that the product specification says it should do.

The software does something that the product specification says it shouldn’t do.

The software does something that the product specification doesn’t mention.

**The number one cause for the bugs is specification**

**Next one id design**

**Next code**

**Test and release**

Bugs are caused for many reasons but the number one cause foe more bugs is specification

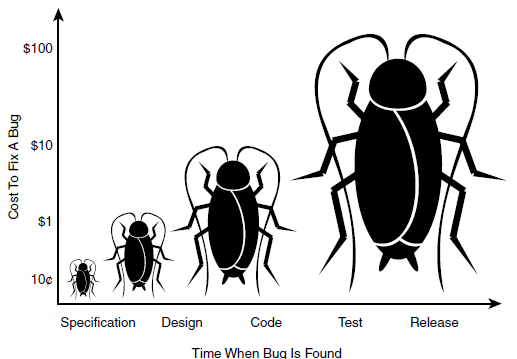
Specification is a largest bug producer its consistently changing or its not communicated to the entire developer team.

Planning the software is too important .if the planning was not done perfectly bugs will be created

The next largest bug producer is design

That means wrong design more bug

THE COST OF BUG



The bug costs are low when it is located in sepcification that gradually increases to design to code and code to test and relaese respectively.

If the coustmer finds the bug in relaese lev that will more expensive it will loss more the laks

Example: if amazon flipkart app is crashed for one hour due to the bug the operations will not happen that time then there in no servese also there they will loss crores of money that one hour.

Or

In amazon a single product cost is 2000 but in the application level its showing 20 due to bug after relaese then lost will happen that time.

***The goal of a software tester is to find bugs, find them as early as possible, and make***

***sure they get fixed.***

***SOFTWARE TESTING :-***

To check the software is ok.

The goal of the software tester is to find bugs

This is process of the application to find out error in it

Verifying that software or application is bug free

***TYPES OF TESTING***

**MANUAL TESTING AUTOMATION TESTING**

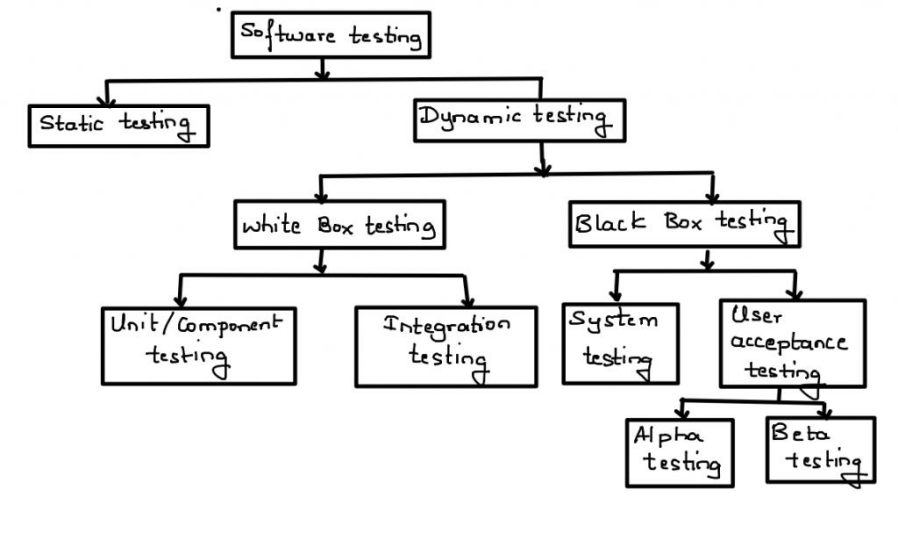
Manual testing:- manual testing the software manually.without using automation tools.

Automation testing:-automation test is known as using automation tools,like if the tester write the script and used different software to test is called automation testing.

This test involves automation of manual process

Automation Testing is used to re-run the test scenarios that were performed manually, quickly, and

repeatedly.



**Static testing techniques**

Analysis of a program program carried without excuting the program.

Done during verification process I.e before development

As we know 80% errors occur in desiegn phase

Is code is tested in static testing ? “No” || the documentation is tested

Software development starts, continues, and ends with documentation

Early documentation → is used to defi ne the software to be built

Later documentation covers → the software training, installation, and operation (user guides)..

Static = not while running

The primary goal of static testing is reduce defect by reducing defects in the

documentation from which the software is developed

**REVIEW:** It is a type of testing done before excution

Review is a process during which a work product or set of work products.to present managers,users,customers or other intrested partiesfor comment

**Walkthrough review**

It is not a formal process

It is led by the authors

Author guide the participants through the document according to his or her thought process

to achieve a common understanding and to gather feedback.

Useful for the people if they are not from the software discipline, who are not used to or

cannot easily understand software development process

**Inspection review**

It is a most formal review type

It is lead by the trained moderators

During inspection the documents are prepared and checked thoroughly by the reviewers before the meeting

**Informal review**

It is an unplanned and undocumented review

**Technical review**

Documented

Defined fault detection process

Includes peers and technical experts

No management participent

Static testing(before excution)

Review

Walkthrough (not Formal)

Insepction (formal finding fault)

Informal review (Unplanned and undocumented )

Technical Review (fault process)

**Dynamic testing techniques**

The process of evaluating a system or component based upon its behaviour during execution

**Black-box testing (functional)**

High level (main testing by tester)=system+UAT

1. system testing
2. Uat testing

**White box testing (non functional)**

Low level (programmer)=unit+integration

1. unit testing
2. Integration testing

Levels of testing

1. unit testing

In unit testing individual component of programm tested.the purpose of this testing is that each module is working properly.if focuses on small unit of software design(done by the developer by using simple input and observing its sample output)

Eg:- In a progarm we are checking if loop method or function is working fine

1. integration testing

In integration testing individual units are combined and tested as group(Developer)

1. top down
2. Bottom up
3. Sandwich
4. Big-bang

Main purpose of integration testing is to check modules are communicating with each other as DFD data flow diagram which is specified in TDD( technical document diagramdaiagram)

**system testing** :in this testing we can test whole application (complete/integrated sofware is tested)done by tester

1. Acceptance testing

A level of software testing in which software is tested for acceptance UAT done at client location where software is actually used

1. alpha testing:- done by tester in company in presence of customer
2. Beta testing :- done by customer to check the software is ok or not(safety requirement)eg (no immediate solution is provided)
3. Installation testing:-providing resources at client location.it is a type of testing that test engineer check deployment process as per user guidelines

Deployment document/user manual= it is provided by project manager

4 .Usability testing:-checking application for user friendliness

5 . Monkey testing:-used for game testing,used for random input

To check the application or system will crash

6 .portability testing:-developed application should support multiple environment

7 .forced error testing:-to check valid error message will display

8 .exploratory testing:- when test engineer does not have idea of functional testing then he is learning through exploring application.

9 .End to end testing :- we can check all internal component for successful response

Internal component like client network server database etc are working fine means testing internal component

1. Security testing:to check the security of the application
2. Reliability testing:-to develop application to support longer duration (stability)
3. Audit :- an independent evolution of software
4. Inspection:-it is a formal evolution of the software
5. Concurrency testing:-multi user testing
6. Debugging:-excuting programm line by line

**SDLC(SOFTWARE DEVELOPMENT LIFE CYCLE)**

IT IS A PROCESS USED BY SOFTWARE COMPANY TO DEVELOP DESIGN,TEST SOFTWARE

1. requirements analysis (client needs)
2. Design (blue print)
3. Coding or development
4. Testing
5. Mainatinence

**Some of the most popular sdlc models**

1. **WATERFALL MODEL**
2. **V SHAPED MODEL**
3. **INCREMENTAL LIFE CYCLE MODEL**
4. **SPIRAL MODEL**

**WATERFALL MODEL**Water fall model is a old traditional model.it will done by step by step like development to testing to deploy

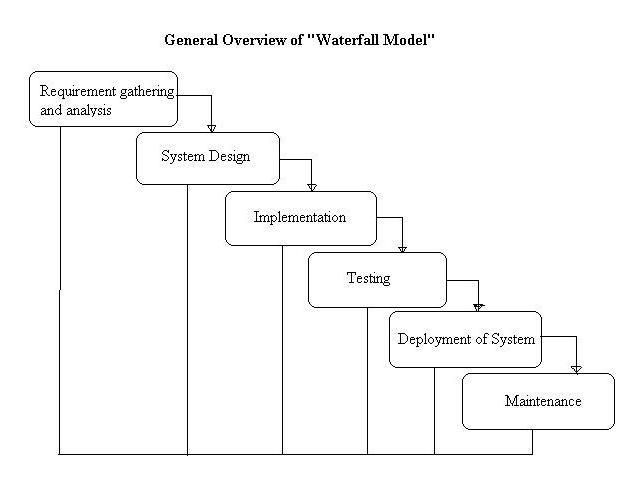
In waterfall model the product quality is good because it is step by step in each step there will be separate document.

Waterfall model not able to accept any reqirements in between project so bugs will be low.

Initial investment is less because after development it need tester

No changes in the middel

Testing will start after coding or development



**2.V SHAPED MODEL**

Verification and validation

Verification:- done before development (to check we are doing correct or not )

We have to verify the document because software is not ready

Verification = before s/w = static

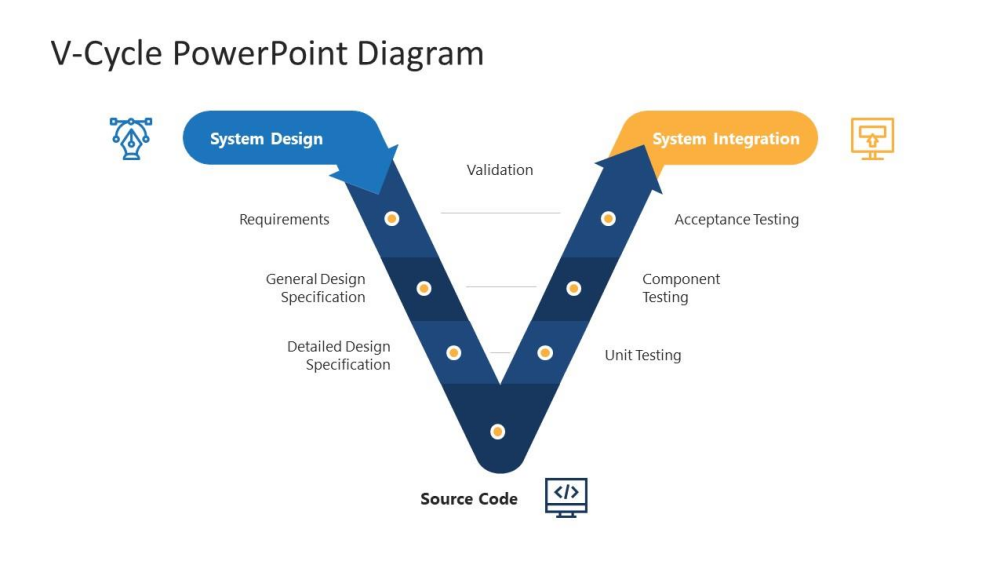
Review-walk through-inspection

Static testing=verifying the document

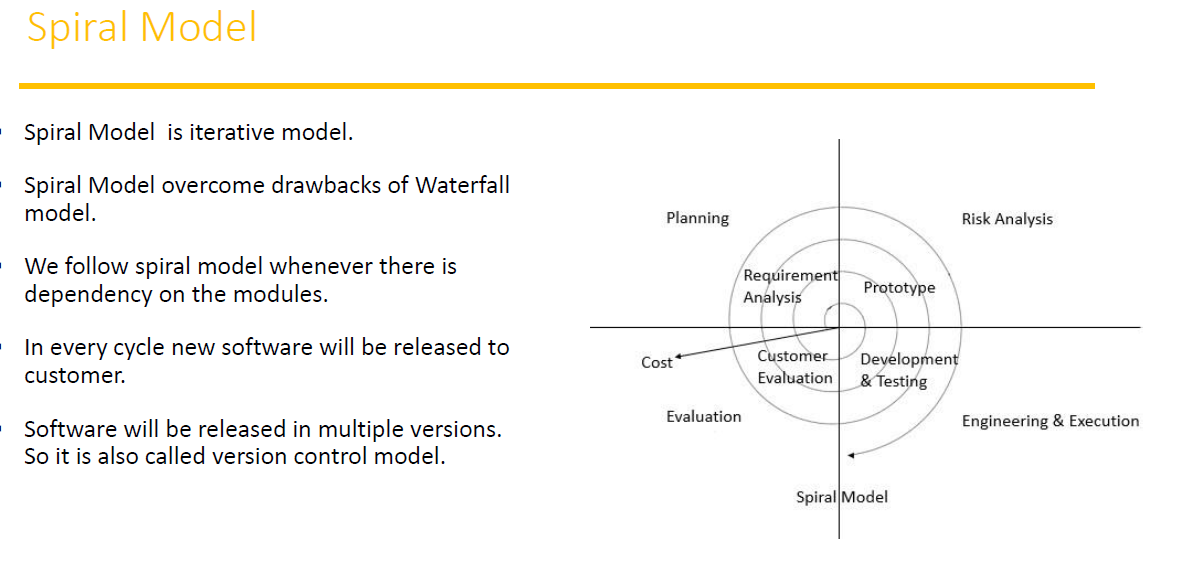
Validation **→** Actually testing software → done after software ready → done Right ?

Validation = after s/w = dynamic--when the product is ready u have check it is ok or not

Dynamic testing=unit-integration-system-user acceptance



**Spiral model**



Spiral model is iterative model

Spiral model over comes the drawbacks of water fall model

In every cycle new software will released to customers

We follow spiral model whenever there is dependency on the modules

Software will release in multiple versions .so it is called as version control model

**Advantages of spiral model**

Testing is done in every cycle before going to the next cycle

Customer will get to use the software for every module

requirement changes are allowed after every cycle Before going to the next model

**Disadventages of spiral model**

Requirement changes are not allowed in between the cycle

Every cycle of spiral model looks like waterfall model

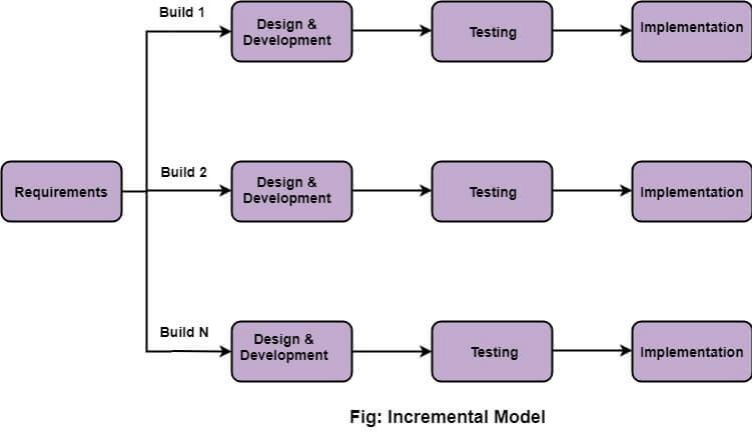
There is no testing is required in developer mode

**Incremental model**

Requirements are divided into multiple modules each module goes through SDLC phase

Analysis-- design-- coding-- testing-- maintenance-- requirement

Module1+module2+---------------module n



When use incremental mode

The project has a lengthy schedule

The cost of or budget of project is high

There has to be good plan

**QA | QC | QE**

**QA:-Highest possible quality**

Quality assurance| process related| high level management

The process designed by QA| responsible for highest quality

**QC:-quality control**

QA is process oriented| QC is product oriented(that work on project of actual tester)

QA:- responsible for preventing defect(involves all phases)

QC:- responsible for finding the defect (involves only testing)

**QE:- Quality engineer**

QE:- responsible for write the code for testing (automation engineer)

**System testing(actual testing)**

GUI Testing | Usability Testing | Functional Testing | non functional

1.**Testing GUI application** , user interface testing

such as menus , check boxes , icon, images

not functional , just look and feel

check size and position of element

image quality, spelling check , alignment

Fonts are understanding or not

**2. Usability Testing** :

check the easiness of application

helping messages are display if user confuse

check user friendly application or not ?

3**.Functional testing**

Check behaviour of the application

Check database testing

Error handling display error message

Calculation and manupulation

Eg. 5+5 = 15 ( user requirement ) => we follow this

5+5 = 10 ( math calculation)

check text box disable or enable as user requirement

Check database operation DML table, column , records etc



Checking database operation

Black nox testing+white box testing =grey box testing

1. **Non functional testing:**

Once functional testing done then software work user requirement then do non functional testing

Performance testing :-

Load testing:- gradually increases the load

Stress testing:- suddenly increases the load (eg online filling form)

Volume handle :- how mauch data will handle

Security of software

Recovery application

Compatibility testing (work with all platform)

End to end testing:-

Eg : login → add customer

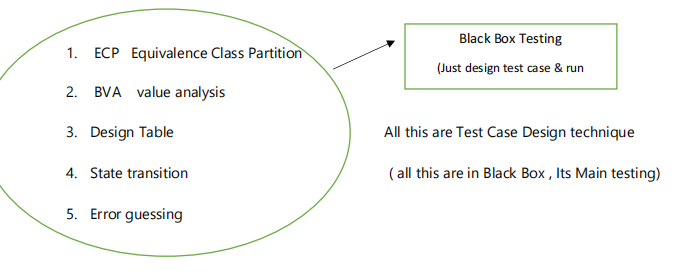
→ delete and edit customer

→ logout

Testing all function i.e. add delete edit and logout

**Test Case Design Technique:**

It helps better design and reduce the number of test cases to be excuted



**Boundary Value Anaclasis (BVA)**

18-60---allowed to sign in

Min—18---Valid

Max—60--Valid

Middle Value---50,45,25,20---Valid

Min-1=17---In Valid

Max+1=61---In Valid

**Equivalence class partitioning**

1-100—Allow

Min Max Middle

1-10 10-20 20-30 30-40 40-50 50-60 60-70 70-80 80-90 90-100

5 14 25 33 45 47 63 76 89 95

**Design table**

this technique is used if we have more conditions and based on condition we have to perform the action

Eg:- transfer money from account to account

Condition:-

Account number has to be there

Otp matched

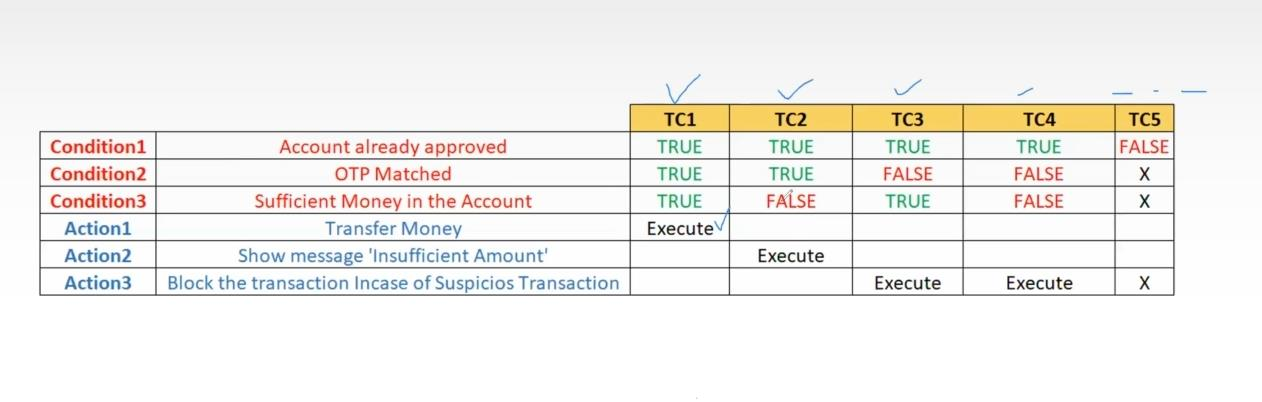
Sufficient money in the account

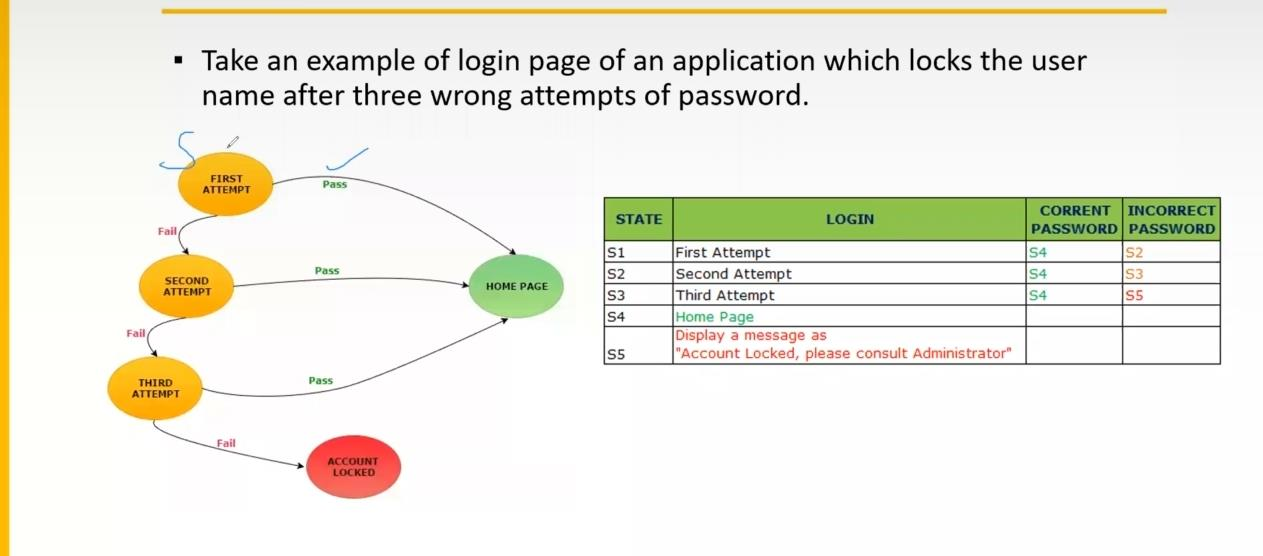
Action :-

Transfer the account balance

Show insufficient money in the account

Block account if any suspecious activity

  **state transition technique**



The case is if 3 attempts are happen in the application

If password is correct allow

Id password is wrong don’t allow

If password is typed wrong for 3 attempts block the account and return to homepage

**Error Guessing Technique :**

No any specific technique

This test based on tester skills

Eg:- submit form empty and gausess ce error

**Test case Scenario**

Simply the name of the tester skill an what to test

**Test case**

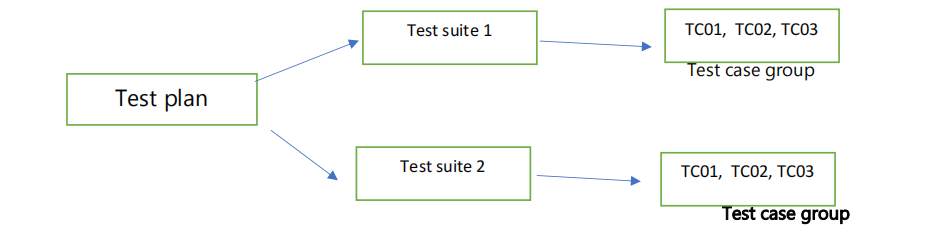
→ how to test i.e. step

Group of steps that is to be execuated to check functionality

Eg. Test scenario = check functionality of login button

Test case = TC01 , TC02 , TC03 , … etc

**Test suite ---**group of test case



**Test case document**

Test case Id

Test case Title

Description

Precondition

Priority

Request id

Steps/ Action

Excepted result

Actual result

Test Data

**Requirement Tractability matrix : (RTM)**

Trace how many test cases are executed or covered

In simple keep track of test cases