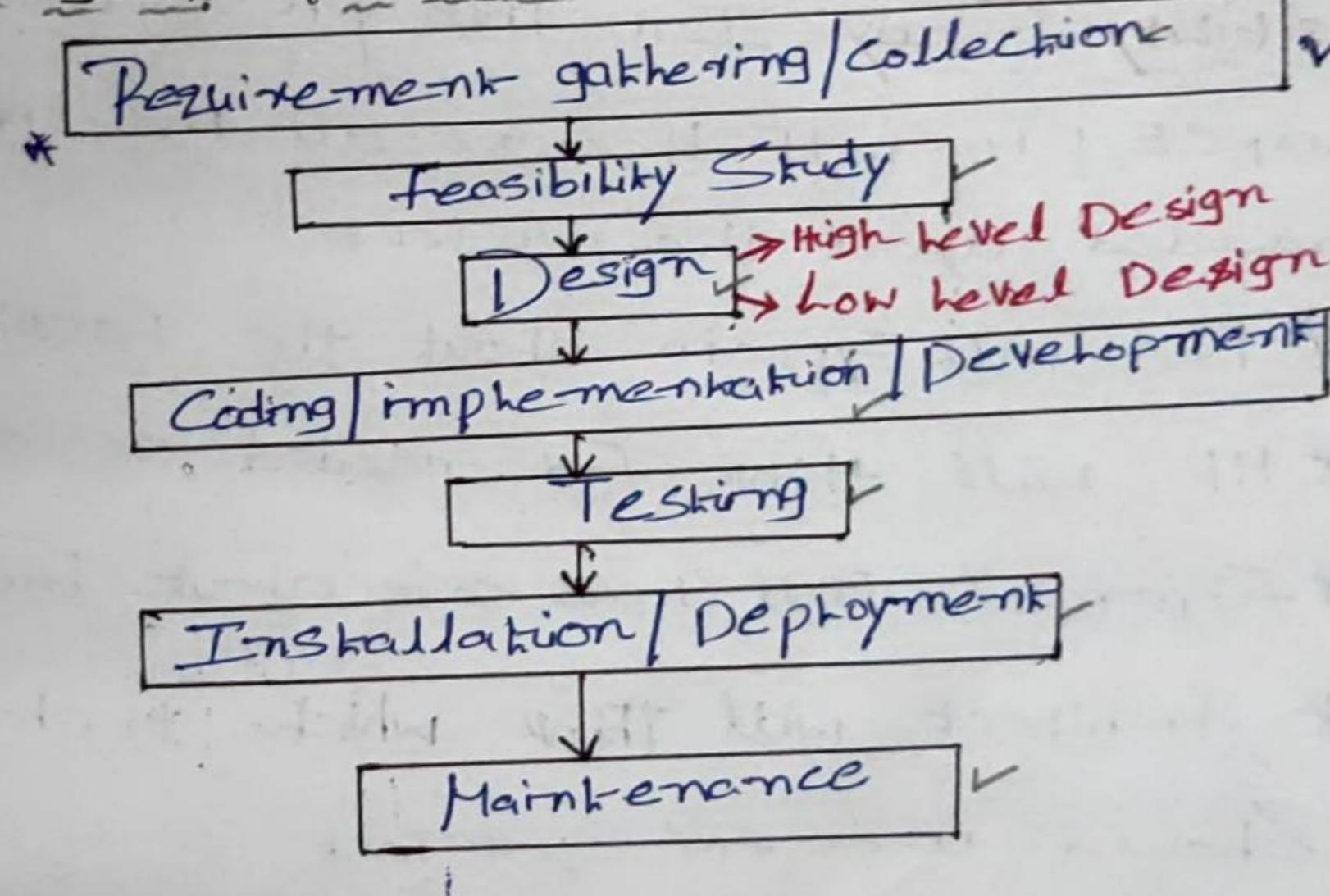


MANUAL TESTING

SDLC (Software Development Life Cycle)

- * It is a procedure to develop the Software Application
- * It consists of different Stages or Phase
- * CRS: Customer requirement Specification
- * BRS: Business requirement Specification
- * SRS: System / Software Requirement Specification
- * BA: Business Analyst
- * HR: Human Resource
- * HLD: High level design
- * LLD: Low level design
- * TE: Test Engineer
- * PM: Project Manager



Requirement Gathering / Collection: 1. Usually Customer Client Gives a requirement in the form of CRS/BRS

2. And it is converted into SPS by BA
3. BA is a Business Analyst.
4. BA acts like bridge b/w Customer and Company

BA Should be Good in

1. Communication Skills
2. Domain Expert
3. Convincing Skills
4. Technical Skills
5. Analyzing Skills

* BA collects all the Requirement from the Customer and Gives to Software Company This is Requirement Gathering / collection

Feasibility Study: In This phase BA, HR, ARCHITECT, FINANCE, PM will discuss all the information that is needed about the project

- * BA will Explain about the Requirement
- * HR will think for needed resources
- * Financier will take care about budget
- * Architect will think which technology we should use for project
- * PM will take the final decision and gathers all the data information to proceed for the next stage (or) phase.

Design: Once everything is fine, we go for design

- * In design phase, we have HLD and LLD.
- * Usually design phase is done by Architect or Senior developers

* High level Design: It is just likeable print

* It Shows the External Architecture of the Application

Low level Design: It Shows the internal Architecture of the Application

- * Both HLD and LLD are the documents

Coding: Once the design is done, developers starts writing the "source code" for Application by looking at design and requirement

- * All developer will involve in writing the source code. as (SDET developer)

* Developers will write front end and back end source code

* This is done in development server. All the developer will be involved in this

Testing: Once Coding is done, developers will give

Application to test engineers. All test engineers will start testing the Application (UI) and find defects

- * If any defects are found Software Application
- * It again goes back to the developer.

- * Developers will Fix it and again given back to Test engineers. like this process continues till Application meets / matches the Requirement
- * In testing Server, Testing will be done
- * Once Everything is Completed, we will go for next phase called Installation/ Deployment
- Installation/ Deployment: Making the Software Application Available For end users is called Installation. i.e Moving the Software From testing Server to production Server (End user will use)
- * It is done by the Company once testing is completed
- * Customer/ Client Should Approve For installation
- * It can be done by Installation Team/ Testing Team/ Development Team
- Maintenance: Once Software provided to Customer (or) End users. If they face any problem, a support has to be provided. that is maintenance
- * Initially it will be free, later it will be paid.
- * Free service will be provided based on the Agreement b/w Customer and Company
- * During maintenance, defect fix will be handled and changes will be taken care

- * Changes can be adding, modifying, Deleting a Feature
- * Maintenance always happens once Application is in production Server.
- * It will done development team / maintenance team (technical team). This is decided by customer and manager

SDLC Models:

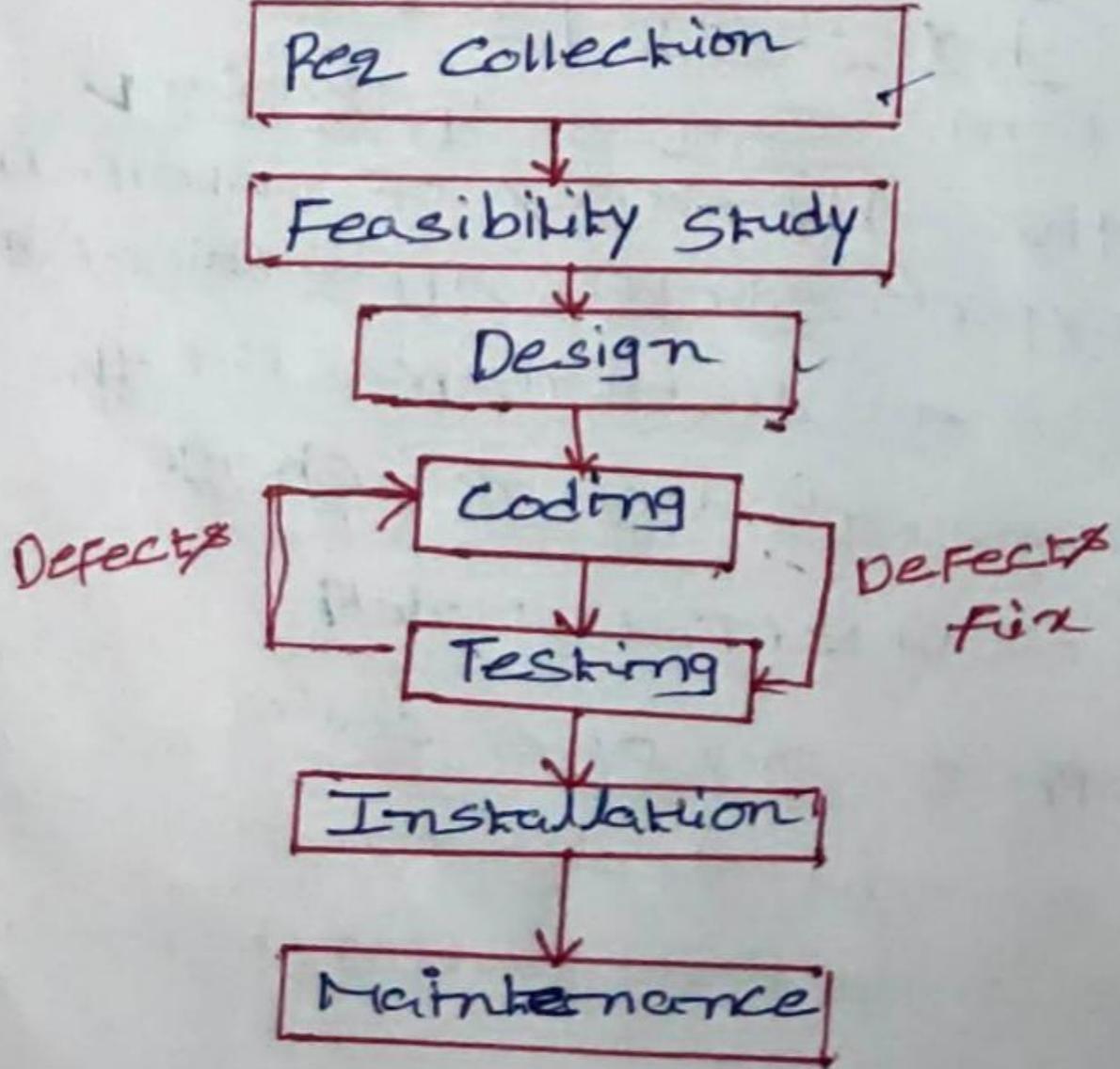
- * Waterfall Model ✓
- * Spiral Model ✓
- * V and V Model ✓
- * Prototype ✓
- * Agile ✓✓

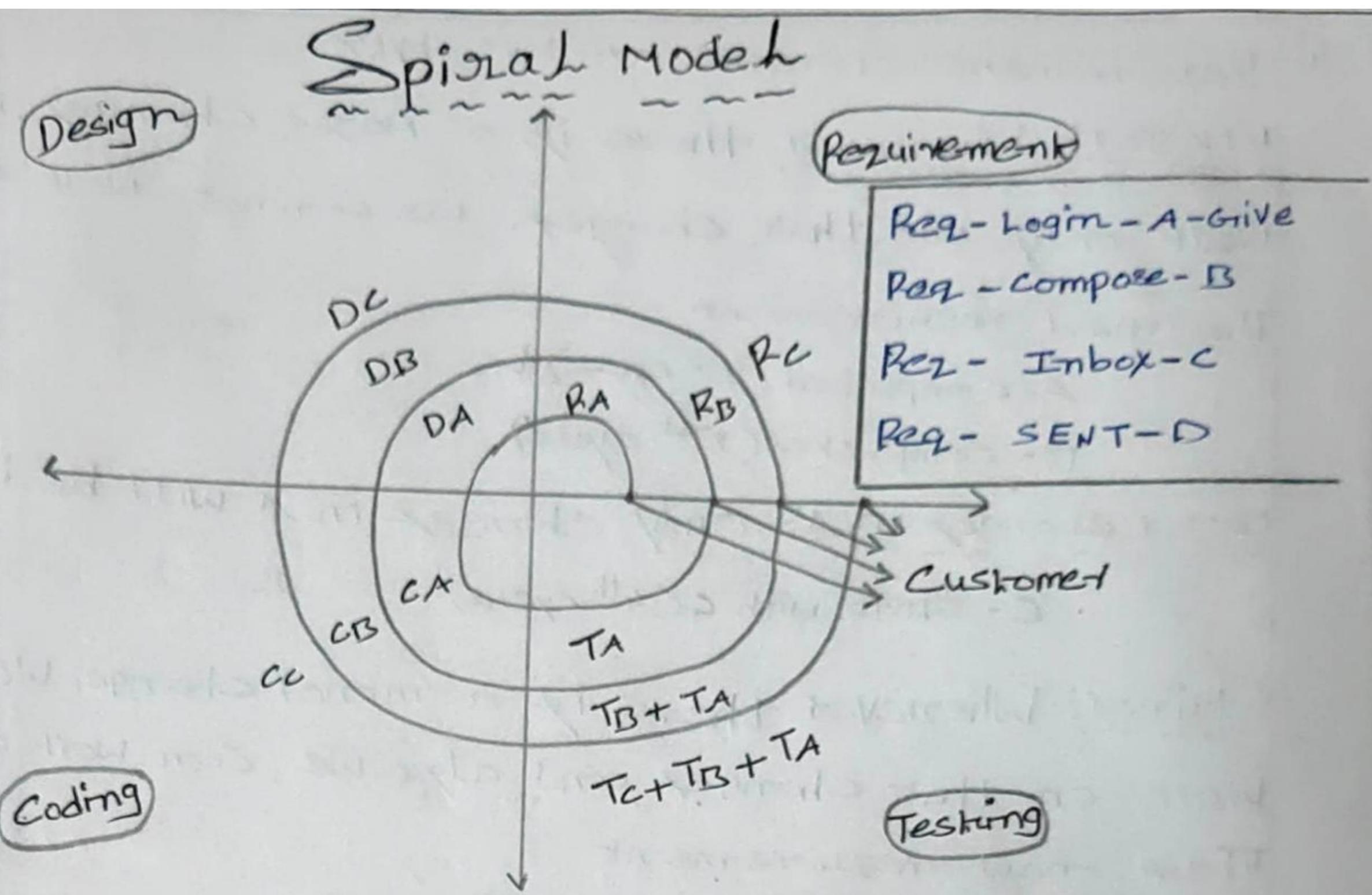
Waterfall Model:

- 1) It is one of the oldest traditional Model
- 2) When we go for Waterfall Model?
 1. When the 'Customer' (Freezes) (Does not change) the requirement
 2. For any Short-term project
 3. For developing Simple Applications
- 2) What are the Advantages of Waterfall Model?
 1. We can expect Stable Application (If requirement does not change)
 2. There will be no disturbance for the team members if the requirement does not change
- 3) Drawbacks of Waterfall Model?
 1. Testing happens only after coding
 2. Requirement and design are not tested
 3. Developers used to do testing before (Currently it is done by test engineers)
 4. If requirement changes, it leads to lot of 'REWORK'

- 4) Why developers are not involved in testing?
- 1.) Developers always focus how to build the software
Not to break the Software
 - 2.) They always be over confident what they do.
 - 3.) They does not like to find their own mistake
 - 4.) Since it is not easy to find their own mistake
Even though mistake is there, they will hide it
- 5) Why Requirement changes?
- ~~Ans.~~ Customer will change the requirement because of below reasons
- * Due to competition ✓ ✓
 - * To upgrade Technology ✓ ✓
 - * To Stay in Business ✓ ✓

Waterfall Model





* When do we go for Spiral Model?

- 1) When the customer gives requirement part by part
 - 2) When there is a lot of dependency b/w the modules
- 2) What are the advantages of Spiral Model?

- 1) Customer can see the Application partially and get Confidence
 - 2) Requirement changes can be done ✓
 - 3) Compare to Waterfall model, rework is less ✓
- 3) What are the drawbacks of the Spiral Model?
- 1) Requirement and design are not tested
 - 2) Testing happens only after coding ✓
 - 3) Developers used to do testing (but now TE is doing testing)
 - 4) If there is any requirement changes it may delay the project

4) Requirements changes are two types.

1. Major: Whenever there is a major change. We work only one that changes. We cannot work on

The new requirement

A - Completed (1st cycle) ✓

B - Completed (2nd cycle)

Major changes in A: Only changes in A will be done
C - Start with C (4th cycle)

2. Minor: Whenever there is a minor change, we can work on that changes and also we can work on

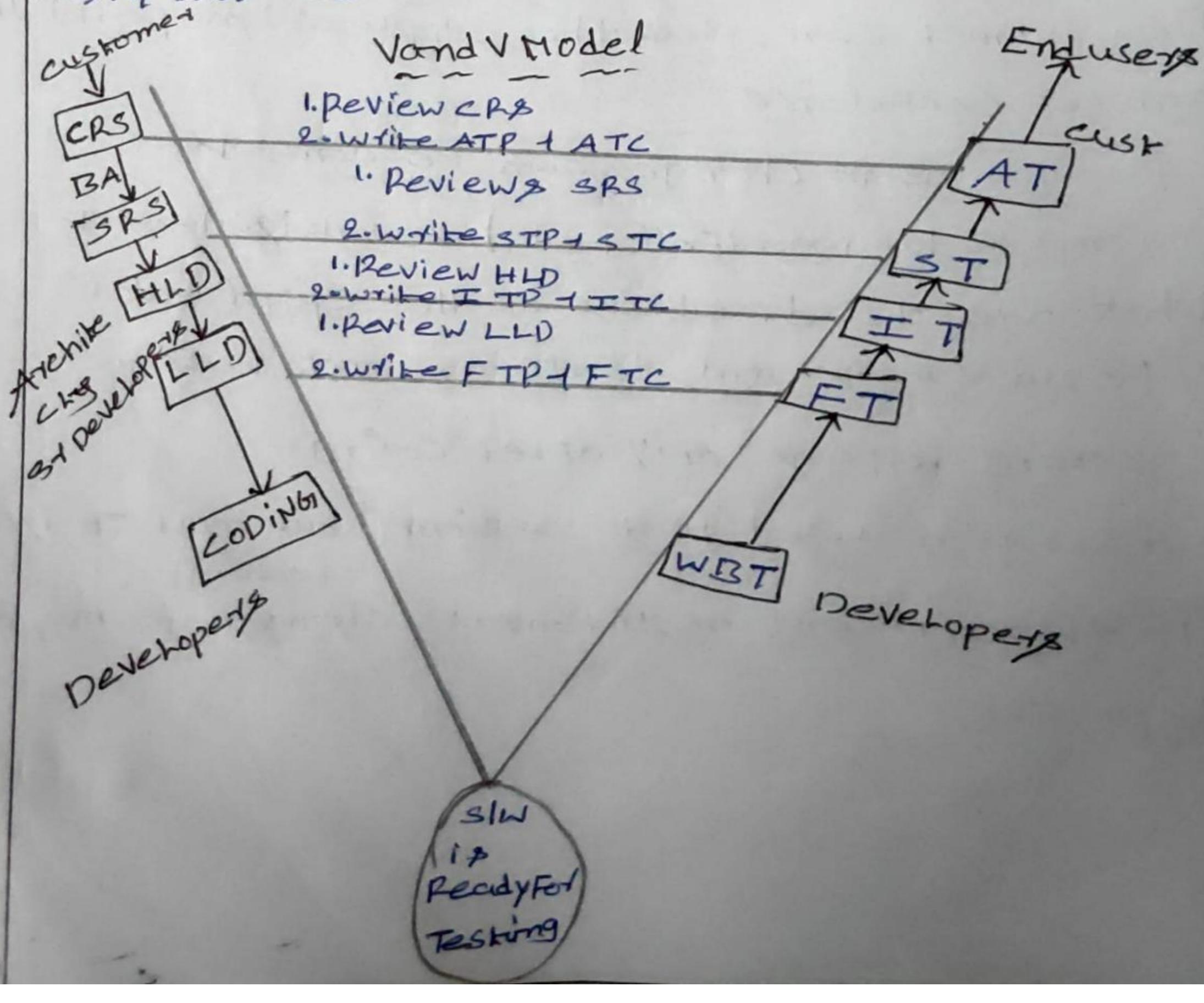
The new requirement

A - Completed (1st cycle)

B - Completed (2nd cycle)

Minor changes in A/C: Will work on minor changes

in A + will also work on C Module (3rd cycle)



V and V Model

1. V and V Means Verification and Validation
2. Verification is a process of checking "are we building product right"
3. Validation is a process of checking "are we building right product"
4. V and V is done only by test engineer
5. Verification is a process
6. Validation is a process

When do we go for V and V model?

- 1) When the customer needs high quality product
 - 2) For Complex Application
 - 3) For long term project (more than 1 year)
- Eg:- Banking, insurance, health care Application
Airlines Applications, many Applications etc etc

Advantage of V and V model:

- 1) Testing is started at initial stage V
- 2) Requirement and design are tested
- 3) The downward flow of defects are less
- 4) Requirement change can be done
- 5) Quality will be high compared to other model
- 6) Rework will be less

Drawbacks of V and V Model:

- 1) Documentation work will be more
- 2) Too much of resource are needed

Explain about V and V model:-

- * V and V Means Verification and Validation Model
- * It is one of the best Model in SDLC
- * In This Model the development and testing are done parallelly.
- * The left side Model is done by developers and Right Side of The Model is done by the test Engineers.
- * In the Middle Verification is done by test Engineers.
- * When The customers give the requirement of 100 Pgs Documents in the Form of CRF.
- * It is Converted in the Form of SRS by BA.
- * At the Same time, Review of CRF is done by test Engineers.
- * If there is any mistakes it goes back, if not it will continue the process.
- * The SRS will be reviewed against the CRF to find the defects.
- * parallelly They prepare the test plan and test case
- * Once the documentation of development process is done.
- * with the design and coding, The Software is Ready for testing.

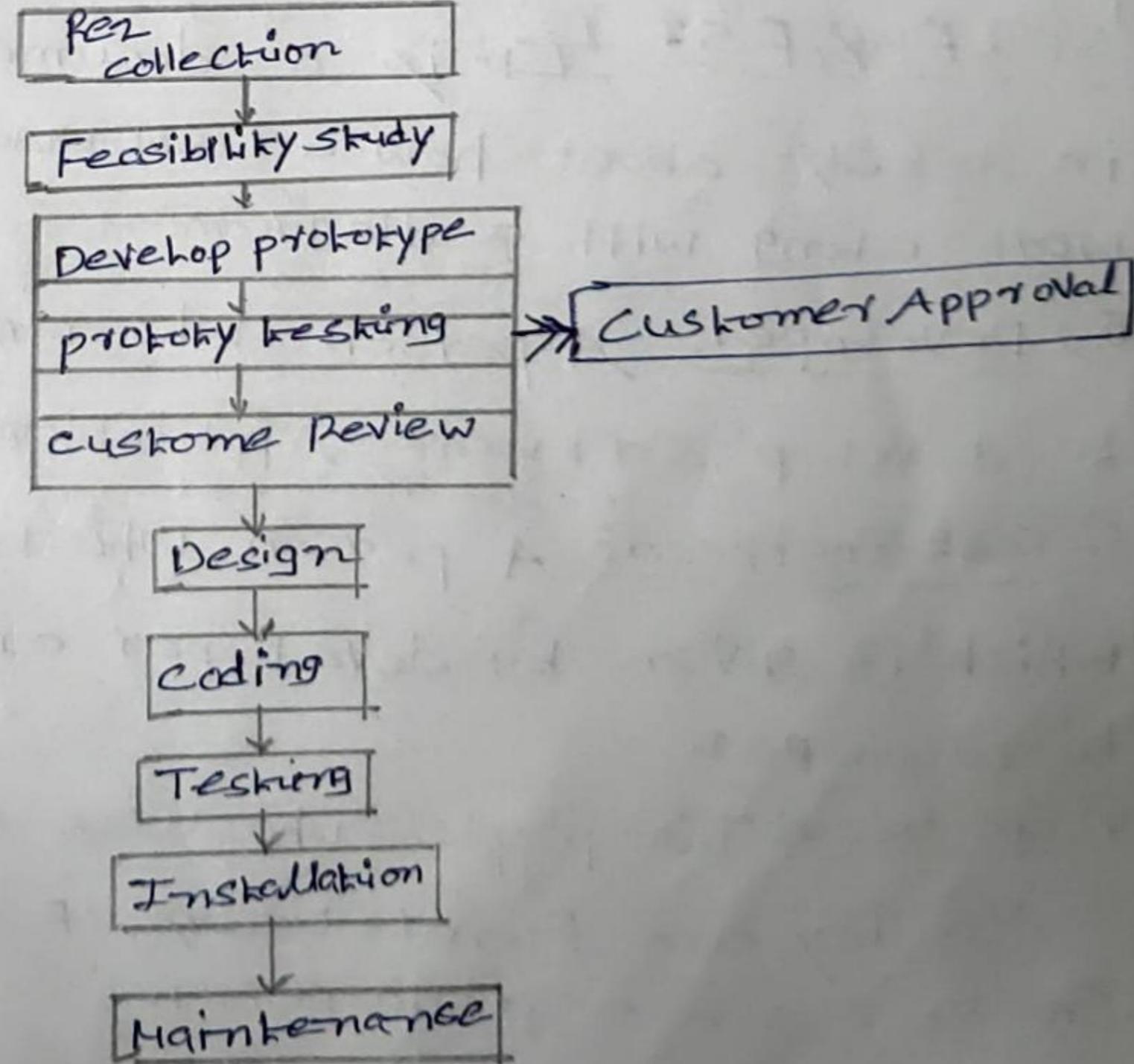
- * First ~~is~~ testing is White box testing. This ^{soo} testing is done by the developers.
 - * Then there will be FT is done by the test Engineers
 - * At the Same the Time the execution of the test Cases are also done
 - * After Functional Testing, The Integration testing is done. later System testing and then acceptance Testing is done by the customer.
 - * Then it is released to end users.
 - * The review of the document is Verification and it is based on QA
 - * The testing of Application is Validation and it is based on QC
- Prototyp Model
- * prototype is a dummy model that is non-working Application
 - When we go for Prototype Model:
 - * When the customer is not clear about requirement
 - * When the software Company is new to the Domain. Then they will go for prototype Model
 - * When the developers are new to the technologies. It is a experimentation.
 - * When the customer and software Company are new to the business

Advantages of prototype:

- * Initially customer can get to know what he gets on
- * Initially itself developers will also come to know what they should deliver on last day.
- * Requirement changes can be done initially itself
- * Initial investment is very less

Drawback of prototype:

- * There will be a delay in the actual start of the real project
- * Investment is done on non-working product
- * Too many changes can disturb the rhythm of the company



Terminologies used in Software industry

- 1.) Req: Requirement
- 2) Customer/Client: A person who interacts with customer to understand gives the requirements to the Software company for his business needs
- 3) Business Analyst: A person who interacts with customer to understand what exactly they need.
BA is a bridge between company and customer.
BA converts CRS/BRS into SRS/FRS/FS
- 4) CRS: Customer requirement specification
BRS: Business requirement specification
→ It is a document given by customer explaining his business needs
SRS,FRS,FS: It is a document which explains in details about how a software should look and work along with a diagram
- 5) Developer: A person who write a source code to develop a Software Application based on Req
- 6) Test Engineer: A person who test an Application which is given by developer and find defects based on Req
- 7) End user: The people who uses the Application at last. They are Final users of an Application
End user can a public person

* Defect: It is a deviation in the application with respect to requirement. It is formal name

* Bug: it's an informal name of defect

Whenever a defect is found by a test engineer it should be raised to the developers.

Once the developers accept the defect, it is termed as

Bug

Error: Mistake in the source code is known as error which is done by developer.

Issue: problem faced by the customer or the end users

Failure: Multiple issues will lead to failure when the end users face lot of issues that will lead to the failure

QA and QC

QA: is quality assurance

It is a process oriented

QC: is a quality control

It is a product oriented

QA	QC
It is a procedure that Focus on providing assurance that quality requested will be achieved.	It is a procedure that Focus on fulfilling the quality requested.
QA aims to prevent the defect.	QC aims to identify the defect.
It is a method to manage quality verification.	It is a method to verify the quality validation.
It does not involve executing the programs.	It always involves executing the programs.
It is preventive technique.	It is a corrective technique.

Domain: It is categorization of the Software Application into various fields.

Each Software Company will work for various domains.

Finance:

It is a banking software used by the banks.

Server Environment: A Server is a computer with high configuration.

- * It is like a Super Computer.

- * It accepts the Request from user, identify the user and sends the response to the user.

- * All these will happen Through Internet and user is recognized from IP address.

- * We have usually 3 types of Servers.
 1. Development Server (D.P) ✓
 2. Testing Server (QA) Server (TE) ✓
 3. Production Server (end user) ✓

Ex: www.dev.facebook.com ✓
 www.QA.facebook.com ✓
 www.facebook.com ✓

* We have 3 different Servers because developers, Testing engineers, end users should use the app and work independently.

* When Developers are working for 1 Feature. That Feature is given to testing team, during that time developers can start working for another Feature.

 1. Everybody can work independently
 2. Dependency will not be there
 3. If any problems occur it can be handled separately
 4. Disturbance will not be there for end users

* BFSI: Banking, Financial services, Insurance (Infosys, TCS, Capgemini)

* Health care: Legato, Emids, GE

* Education: Byju's, Toppr etc

* Entertainment/ Gaming: Sony, EA Sports

* Social media: FB, Instagram, WhatsApp, etc

* E-commerce: Amazon, Flipkart, Myntra

Software testing:

- 1.) Testing The functionality of An Application with respect to given requirement
- 2.) Checking The application with the intent of finding the defect ^{Internet}
- 3.) Checking the behaviour of An Application to see whether it meets customer requirement or not
- 4.) Testing The process of QA and QC

Manual testing:

Testing The Software without using any tool And Ensure its working fine

Automation testing:

Testing The Software by using the tool to ensure it is working fine

Example: (The tool can be Selenium)

TV Remote: If we operate by using the remote it is a Automation testing

* If we do without remote it is manual testing
* Why testing important: ✓

Every Software is developed for business purpose if testing is not done, end users may find the defects while using the application

This will spread negative impact in the market and number of users who uses Application will be reduced. This will become loss for the investor on business. To avoid all those things SW testing has to be done before it is released to end user. Difference b/w Developers and Test Engineers!

Developer	Test Engineer
1) Build a Software	1) Break a Software (Finding defects)
2) He Should be good In logic	2) He Should be good In Creative, out of box thinking
3) Positive Thinking	3) positive + negative thinking
4) Aim For Construction	4) Aim For perfection ↑ Debugging
5) Single thinking	5) Multi Thinking
6) Narrow mind	6) Broad Mind

SCENARIOS

- 1.) Testing the Application in all the possible ways of scenarios.
- 2.) It can be +ve Scenarios (or) -ve Scenarios
- 3.) Testing the Application with valid or expected Data is called +ve Scenarios. It is also called positive testing
- 4.) Testing the Application with Invalid or unexpected Data is called -ve Scenarios. It is also called negative testing
- 5.) To identify the defect we should find the Scenarios First

- ⑥ Scenario is not a defect or defect is not a Scenario
- ⑦ +ve scenario and -ve scenarios are not a defect
- ⑧ Few Cases we may get confused, whether it is a +ve scenario or -ve scenario. It is also in this case we can judge whether it is a defect or not a defect based on Req

- * We can get "positive scenario with a defect"
- * We can get "positive scenario without a defect"
- * We can get "negative scenario with a defect"
- * We can get "negative scenario without a defect"

Note:-

Unless Interviewer asks, do not divide positive and negative scenarios

Drawbacks of Manual testing:

- 1.) It is a time-consuming process
- 2.) It is a repetitive in nature (Because of Regression testing)
- 3.) Turn Around time is more in manual testing.
So we go for Automation testing
- 4.) Need more manpower

Retesting and Regression testing:

- Testing the impacted areas of an Application:
- 1.) After the defect fixed
 - 2.) After changes done in Application.
Changes can be adding, modifying and deleting a feature

Retesting: Testing the defect fixed which is done on an application is called retesting.

Why Regression testing has to be done?

When defect is fixed or new feature is added, old/existing features might stop working.

To ensure existing features are still working fine, we have to do Regression testing.

Since Regression is repeatedly done, many company will go for Automation.

We can also do Retesting and Regression testing manually.

Example: Test Application - TE

Find defect - (Front camera not working) - TE

Raise defect / log defect / Report defect - Test Engineer

Developer will check that defect and Accept defect.

Fix the defect (Modify the source code of front camera) - Developer

Test Engineer - Test the Front Camera again -

Retesting

Impact Area testing - Regression testing

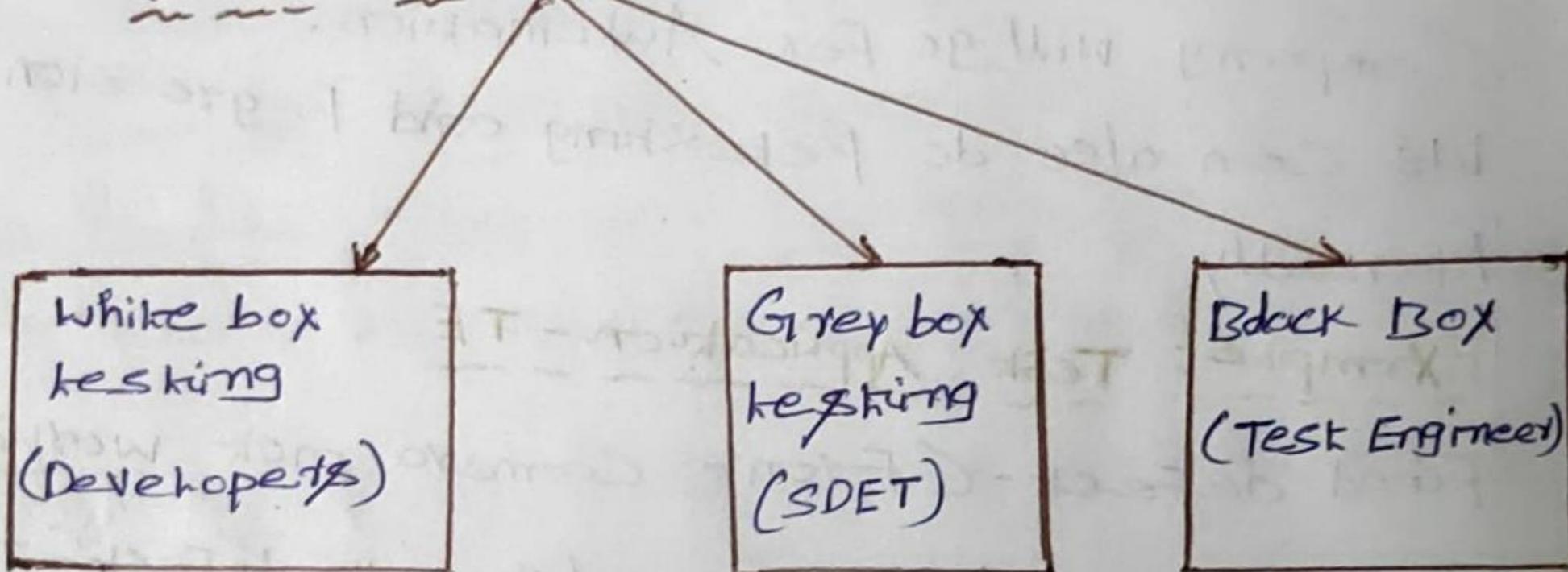
Impact Areas for this defect can be:

Back camera	Settings / filters
Flash	Quality
Gallery	Pixels
Zoom in	Scanning the QR
Zoom out	WhatsApp camera
Video	Instagram camera

Example:-

- * Valid un and Valid pwd and click login - Defect (raise)
Fix-ReTest
* Valid un and Invalid pwd and " " - NO Defect - Regression
* Invalid un and Invalid pwd " " "
* Valid un and Blank pwd and " " "
* Blank un and Valid pwd " " "
* Blank un and Blank pwd "

Types of testing



Unit testing

Glass box testing

Structural testing

Transparent testing

Open box testing

Functional testing

Behavioral testing

Closed box testing

White box testing: Testing the Source Code (or) Internal Structure of an Application is called White box testing

It is usually done by developer

When Situation demands, even test Engineer

Can do this IF we know coding

Black box testing:

Testing the user Interface (UI) or (GUI) of an Application is known as black box testing. It is done by test engineers.

Here we don't worry about internal structure of an application

Grey box testing:

Testing the source code and user interface of an application parallelly together is called grey box testing

It is done by a person who is aware of both testing and source code of an application means he can be a developer, or test engineer. OR he can be SDET (Software Development Engineer in TEST)

Functional testing / Field level testing [Component Testing each and every component independently and thoroughly with respect to requirement is called functional testing]

We perform functional testing for all components like:-

* Text Field

* Labels

* Radio button

* Check box

* Dropdown

* Button

* Link

* Text area

1) Text Field:

Login screen App

Code playon

Two rectangular boxes representing text input fields. The top box is labeled "user name" and the bottom box is labeled "password". Arrows point from the labels to their respective boxes.

2) Radio button:

GENDER

Male Female

3) Check_box:

Meal option

Additions

- Pickles
- Tomato
- Lettuce
- Cheese

4) Drop down:

Select a color

A rectangular box with a dropdown arrow icon at the bottom right. Inside the box, the text "Select a color" is written next to a small checkmark icon.

Select a color

Select a color

- Blue
- Green
- Purple
- Red

5) Button:

Member login

username

password

login
Forgot password

6) Link:

M Compose mail

New message

womehouse@breitkase.com

Subject

7) Text Area:

Send your comments to the author

Submit

- 1) When Developer gives one Module, we start with Functional testing
- 2) Before This, we have to be ready with all Scenarios of all the Modules
- 3) If Developer gives multiple Modules, we Should do Functional testing for all Modules Separately
- 4) Here we Should check all positive and Negative Scenarios for each and Every component and find defects

Ex:- Separately check every Module
Login module ✓ Compose module ✓ Inbox module

INTEGRATION TESTING:

Testing the "Data flow between two (or) more dependent modules" is called integration testing

1) Whenever we are testing one module and If we are sending data to other module, Integration will happen here

2) Checking whether sending data and Receiving Data are same or not is Integration testing

Example: If I send an Email from "compose" Module and checking in "Send" module, it is

Integration testing.

* If the data is not displaying same, then it is defect

* If the data is displaying same

Then it is not a defect

Note:-

- * To do integration testing, Atleast we Need 2 Dependant Modules.
- * After functional testing, we perform Integration testing

System testing:-

Testing the "Functionality" of an Application end-to-end "End to End" Just like a real user is called System testing

It is done on a testing server, which is similar to production server.

System testing Scenarios:-

Ex 1: Sign up → Login → compose → sent items → logout.

Ex 2: Login → inbox → read Email → reply → sent items → logout

Ex 3: Login → settings → reset password → save new password → logout → login (new password) → logout.

Note:-

* We need almost all modules to perform system testing

* After functional testing and integration testing we do system testing.

* Here we get confidence to release application to customer

NOTE: FT, IT and ST are mandatory testing for all applications in the world.

Test Case:

A test case is a document which contain all the possible scenarios and Explains Step by Step. Template is a format used to write a test cases.

When do we write test case?

When the developers are developing the Application, we will write a test case.

There are two stages?

* Test case preparation / writing:

Under the test case preparation, we have to fill following sections like

Header, (Body, Footer) Sections

In body section, we will fill Actual Result.

Test case Execution:

After Application is given by developer, we will fill Actual Result, Status, Comments.

Difference b/w Scenario and Test Cases.

Scenario	Test Cases
* It will tell all the possible ways we can test the Application	* It is step by step procedure to test the Application
* Scenarios say "what to test"	* Test cases says "how to test"
* Scenarios "HLD Document"	* Test case is HLD document
* Scenarios doesn't have a navigation steps	* Test cases have a navigation steps
* Scenarios take less time to write	* Test cases take more time to write

- * Scenarios doesn't say where the Exact defect is present
- * Test cases will tell where the Exact defect is present

NOTE:

- 1) Under functional test cases, each and every Step is functional Scenario
- 2) When we write Integration test case, multiple Steps will become one integration Scenario
- 3) When we write System test cases, all Steps will become one System testing Scenario

Tools used in Industries

Functional testing tools / Black box testing tools

- * Selenium: (Can be used with Core Java, Python)
 - 1) It is a tool which is in demand and it is an open source / free tool. The demand combination is that Selenium is used by Core Java.
 - * Thought works is the first company to come with the Selenium
 - * Selenium is a free tool is an open source
 - 2) QTP/UFT (Quick test professional, unified function test) can be used with 'VBS' (Visual basic Scripting, Java script)
- QTP/UFT is a paid Tool and licensed Tool. It is with HP
- | | | | |
|----------------------|------------|-----------------|---|
| <u>ext TV remote</u> | <u>APP</u> | <u>selenium</u> | <u>battery</u> (placell, nippo, eveready) |
| | | | <u>language</u> (Java, Python, perl) |

- * (Hewlett packard) Before it was with Mercury
- * QTP is older version, UFT is latest version
- * QTP / UFT it is powerful tool and it is a paid and licensed tool.
- * The best combination is that QTP is used by Vbs
- * Selenium is in demand because it is free tool
- * QTP / UFT is powerful tool
- * QTP license cost is around 3 to 5 lakhs

3.) winrunner

ii) Silk test

5) Test partner

Performance testing tools:

Testing An Application by Applying load and check Response Time and Stability

* 1. Load runner ✓

* 2. J meter ✓

3. QA load ✓

4. Silk performer ✓

5. Neo load ✓

* Load runner is a licensed tool, It is with HP

Performance testing (or) scalability testing:

Performance testing of an application

* Testing the response time of an application is known as performance testing

Load: Number of users who user the application
Response time: Time taken to get the expected screen based on the user actions

Types of performance testing:

Load testing:

* Testing the response time of an application by applying the load which is less than or equal to designed number of users (It will be given by customer)

Example: If the customer want Application for 1 lakh users then the designed number of users is 1 lakh

Stress testing:

* Testing the response time of an application by applying the load which is greater than designed no of users that can be around 10-20% greater than the load.

Ex: Per 100000 users ... Response time should be within 2 sec

Stress testing 1,10,000 users ... Response time 2 sec

Volume testing:

* Testing the response time of an application by transferring huge volume of data through Application

Ex: Huge data sharing through share it, Bluetooth, google drive by uploading more photos and checking the time taken to transfer that data

Soak testing: (or) Endurance

* Testing the response time of an Application by applying load continuously for some duration of time

Example: If 2 hrs more users are using application continuously and we will check the response time for 72 hrs. Apply load for 3 days continuously.

① Can we do performance testing manually?

Ans: Yes, but there are some drawbacks like

1.) Too much cost will be involved because of multiple human resource and multiple devices. A huge place for gathering everybody

2.) There will be no accuracy in the result. We human beings we do not perform all the actions at same time. Because of this, result may vary so we go for Automation

Ex(J METER, LOAD RUNNER)

Test management tools:

1.) QC / ALM (quality center / Application life cycle management)

→ It is a licensed tool. It is with HP, it is paid tool

2.) TIRAC (best tools for Agile Model)

3.) OTH (Oracle test manager)

4.) Testlink

→ Using the above tools we can manage the testing activities like below

Ex: QC/ALM. Below are the modules of QC/ALM

Req → Add requirement / Edit requirement / Delete requirement

Test plan → Write test case

Test lab → Execute test case

Defect → report/Raise / log defect. Track a defect

What are the test management activities?

1) Going through the requirement given by the customer and adding to the test management tool

2) Write test cases by identifying scenarios

3) Execute test cases and write results

4) Find defect and report a defect. Track a defect.

Defect (or) Bug Tracking tools:

Bugzilla: It is an open source and free

QC/ALM: (Quality Center / Application life cycle management)

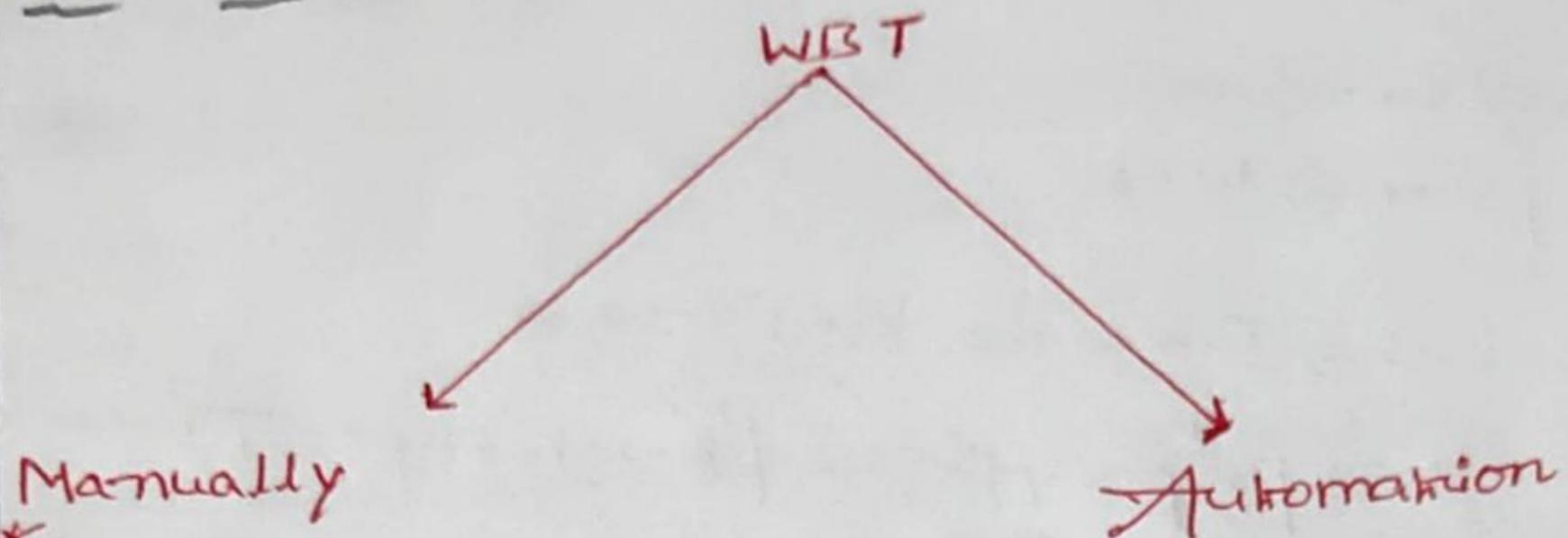
→ It is a licensed tool. It is with HP

JIRA: partially free

OTM: Oracle test manager

* Test link

* In Bugzilla we cannot add the requirement
 write test case and execute test case
 * we can only report the defect and track
 the defect

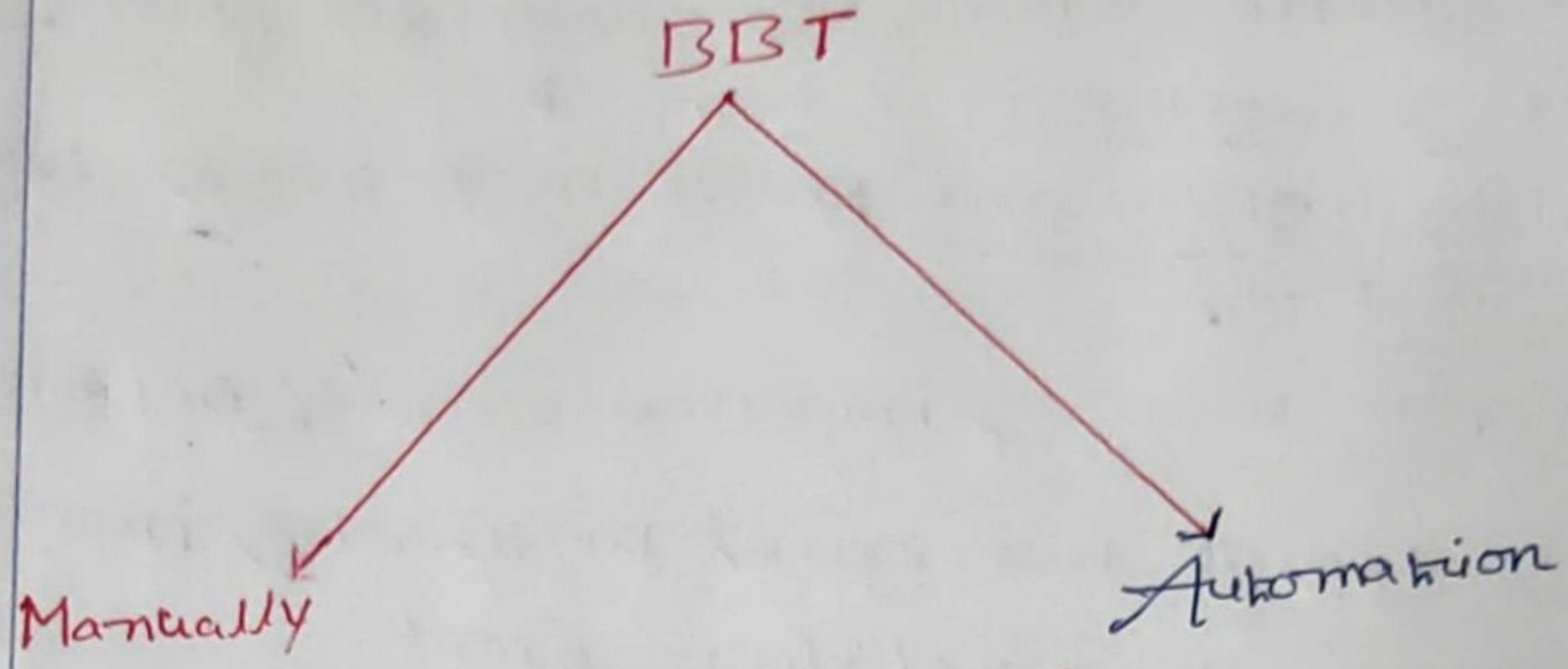


(1) How to do WBT manually?

- 1) Write a program to develop an Application based on Requirements
- 2) Test each and every LOC (line of code) without a tool
- 3) Check for Syntax, logics etc
- 4) Run the developed program and Analyze the Results

(2) How to do WBT through Automation?

- 1) Write a program to develop an Application based on Requirements
- 2) Write a testing program to check the developed program using a tool
 (Ex: JUNIT, NUNIT)
- 3) Run the testing program which will run developed program in a tool (Junit, NUnit)
- 4) Analyze the results



② How to do BBT manually?

- 1) Go through Requirement
- 2) Identify Scenarios
- 3) Write Test Case
- 4) Execute Test case once Application is ready
- 5) Find defects

How to do BBT Automation?

- * Go through requirement
- * Identify Scenarios
- * Write test case
- * Write a program (Test script) to test the Application using a tool
Ex: (Selenium, QTP/UFT)
- * Run the test script in a tool (Ex: Selenium, QTP/UFT)
- * Find defects

Note:

We can write test cases using excel sheet or tools like QC/ALM, JIRA etc
→ Always use the tool if available in the company

If not available, then use template given by manager or Create own template

* we can Import and Export test cases between Excel and tools

Excel (export test) → imported to tool (QC/ALM)

QC/ALM export test cases to excel → import test cases from QC/ALM to Excel

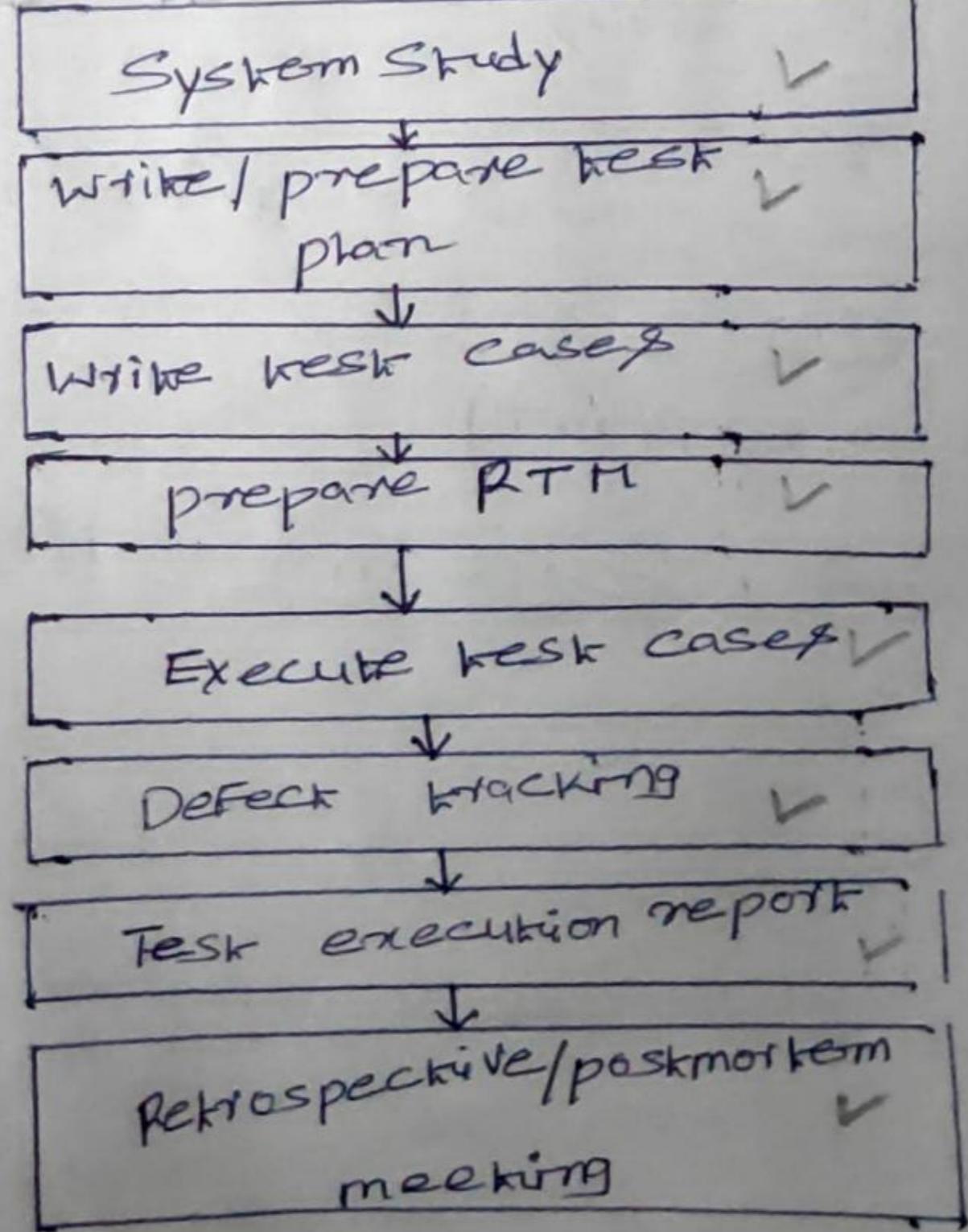
QC/ALM - Excel

Export (Sending) Import (receiving)

Software testing life cycle (STLC)

* It is a procedure to test the software application

* It has different stages or phases like



System Study: It is going through the requirement given by the customer and understand how the system works

Write test plan: It is a document which is prepared for future testing activities

→ It is done by the test head or test manager. Because the plans will be done by experienced people

Write test cases: It is a procedure to perform the testing on the application. It is done by test engineer, once we go through requirement, we identify the scenarios and then converted into the test case. To write the test case we need requirement and test case template or tools (QC/ALM, JIRA)

Prepare RTM: (Requirement traceability matrix)

It is a document which is prepared whether every requirement has at least one test case or not. To prepare RTM, we need both requirement and test cases.

Execute test case: Once the requirement is given to the test engineer, he writes test cases for the application. After the developer gives the developed application then the test cases are executed and it is compared with expected result and actual result.

→ If the Expected result and actual result are same, Then the status is pass, If the Expected result and Actual result are not same status will be fail. This is called execute test cases. So, to execute test cases we need test case and Software Application. This is where exactly the Software is been tested. This is the most important phase of Software testing life cycle.

Defect tracking: While executing the test case we may come across the defects.

→ These defects are raised / logged / reported to the developer.

→ Once we raise, we should also check what is happening to that defect. This is called defect tracking.

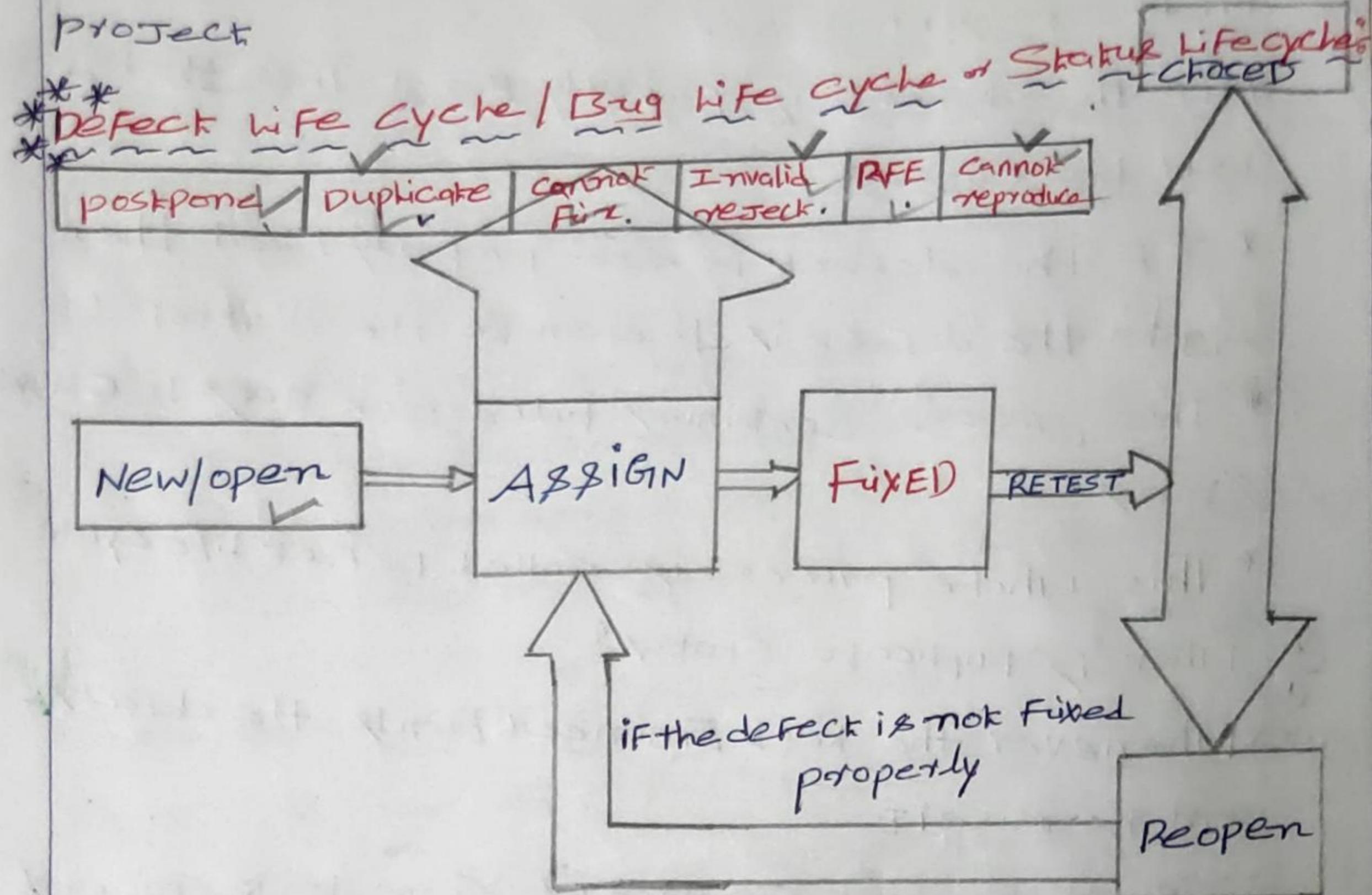
→ It is done by using tool called Bugzilla or ALM JIRA

Test Execution report: Once we execute the test cases, we should prepare one document called test execution report. This document helps about how many test cases are pass and how many test cases are fail.

→ Till here customer involvement will be there according to the customer. This is the end of the project.

Retropective meeting / postmortem meeting: In this meeting, all the team members will gather together like PM, TH, DL, Dev, BA, TM and they discuss about the good and improvements needed once the project is done.

This will be very useful for next release (or) next project



When we are executing the test cases if the expected result and actual result are not same we come across the defect.

- * Then the defect is raised to the developer, the status will be New/open.
- * The defect will be assigned to the concerned developer or development head, then the status is Assigned.
- * Then developer will reproduce the defect and if it is Reproducible, it will be Accepted.
- * Then he starts fixing the defect in the development server and it has to be installed to testing server.
- * Then he should change the status Fixed.

- * Then test Engineer Starts retesting the defect in the testing server
- * If the defect is properly fixed, Then the status will be closed.
- * If the defect is not properly fixed, then Again, the defect is reopen to the developer.
- * This process continues till defect goes to closed status

* This whole process is called Defect Life Cycle

(*) What is Duplicate Status?

Ans) Whenever the test Engineer finds the defects and raise it

If Another test Engineer or Some test Engineer find the same defect and raise it unknowingly mistakenly, Then status of that second defect will be changed as DUPLICATE by Developers

To Avoid This Duplicate Status:

A) Check whether the same defect is already raised or not in the Repository

Ex: Bugzilla, Oct ALM, JIRA before raising any defect

b) Communicate with team members before raising a defect, if you are raising for common module defect

Note: Duplicate is a bad status for Test Engineer

② * What is Can't reproduce / unable to reproduce status:
When the test engineer finds the defect, but developer cannot be able to reproduce that defect. So he will change the status of "Cannot be reproduced".

Ex:- Mobile hang.

→ Reasons for Can't be reproducer

- * Installation problem (Development, Test server)
- * Improper defect report (Chrome, Firefox)
- * Inconsistent bug (A bug which occurs sometimes and does not occur sometimes)

→ New/open → Assign → Cannot reproduce → Fixed → Closed

③ Invalid / Reject / Not a Defect Status:

Whenever the test engineer raises the defect, but developer will not accept the defect as valid and he changes the status to Invalid / Reject.

Reasons for Invalid:

Due to the misunderstanding of the requirement

1. If test engineer's Misunderstand the requirement
New/open → Assign → Invalid → Closed

2. If developer Misunderstand the requirement
New/open → Assign → Invalid → Assign → Fixed → Closed

Note: INVALID is Bad for TE ✓

④ Can't be Fixed / Can't Fix Status:
Whenever the developer is unable to fix the defect. That is raised by the test engineer. Then he changes the status as "Can't be Fixed". And it is a valid defect, but he can't fix the defect.
Ex: Whenever Technology does not support to fix that defect, then developer will change status to "Can't be Fixed".
→ Finally what to do for this defect will be decided by manager / customer
Ex: e.g - Response time < 2 secs - req.
5 secs - defect

⑤ Deferred / Postpone Status:
Whenever test engineer raises the defect, the developer accepts the defect. The developer will not fix the defect immediately. He will fix it later, then the status of those defects will be done as deferred / postpone by developer.
Ex: Whenever there is a major defect and minor defect, developer has time to fix the major defect, in this case the developer will fix the major defect and for minor defect he will make status as deferred or postpone.
→ New / open → Assign → Deferred → Fixed → Closed
(do retesting before closing)

⑥ Request For Enhancement (RFE):

Whenever the test engineer raises the defect which is not given in the requirement so the developer will take it as suggestion. In this case he will change the status as RFE.

Ex: Gmail APP

→ Inbox - 1000 emails - delete all emails - (select all if not there) if this is raised as a defect, developer may make it as RFE

This should be taken care by BA / Manager for

Further decision

They can speak with customer, if customer wants that feature, it is taken as a new REQ.

→ New / open → Assign → RFE → Closed (Add a comment before closing)

Severity: Severity will tell how much that defect is affecting the customer business.

Types of Severity

1. Blocked / Showstopper ✓

2. Critical ✓

3. Major ✓

4. Minor ✓

5. Trivial (This defect is negligible)

→ Severity is used in Defect Report while raising Defect

Priority: Priority says which defect has to be fixed first by the developer

→ For every defect we have to set priority

→ Different types of priority

1. High

2. Medium

3. Low

(OR)

1. Urgent

2. High

3. Medium

4. Low

(OR)

1. P₁

2. P₂

3. P₃

* Who will set severity and priority?

Test Engineer will set severity and priority:

Developer (or) Manager's can discuss about this if they want to change it. They have to provide proper justification.

Priority is very much important for the developer to decide which defect has to be fixed first.

→ If priority is not there, developer may fix the easy defects first and they leave all the important defects.

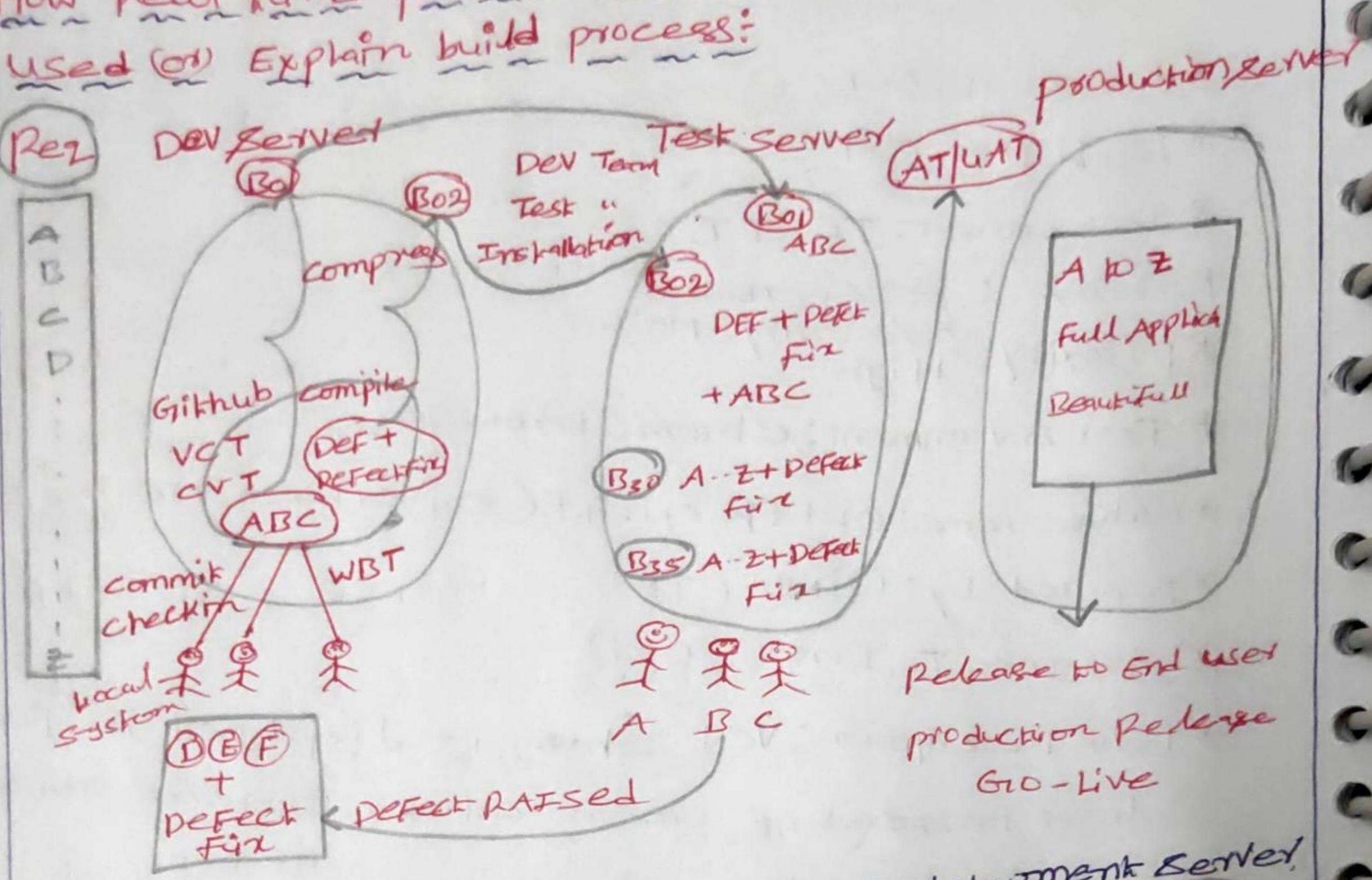
→ As a project point of view, the important part of defects has to be fixed earlier. To manage these things, we have severity and priority for each and every defect.

→ Priority can change once other defects are fixed and closed.

Defect report (i) Bugzilla report (ii) Jira report

- * Defect ID: 001
- * Build NO: B01
- * Test case NO: TC- IT- 05
- * Status : Assigned
- * Priority: *Severity: Major
High
- * Test Environment: Chrome browser
- * Module name: ORDER ONLINE (Superbeam Meals)
- * Reported by: DINGA (TE)
- * Assignee To: Dingi (Dev)
- * Brief Description: Veg Shake is displayed in Red colour instead of green colour (Beetroot juice)
- * Test Data: NA (NOT APPLICABLE)
- * Steps to reproduce / Description:
 1. Open the Browser (Chrome)
 2. Enter the URL (www.zomato.com)
 3. Click on 'Search For Restaurant' Field in Home page
 4. Type 'Superbeam Meals' in Search bar
 5. Select 'Superbeam Meals' (Any Location)
 6. Select 'Order online' option
 7. Click on 'Shakes' option
 8. Scroll down till we get 'Beetroot juice'
 9. Expected result: 'Beetroot juice' should be displayed in Green color
 10. Actual result: 'Beetroot juice' is displayed in Red color

How Real time project works (Q1) How are the servers used (Q1) Explain build process:



* Usually we have 3 servers like Development Server, Testing/QA Server, production Server.

* Customer gives the requirement then the developer starts writing the source code for developing the application in the local system. This source code will be stored in the repository (Ex: github, VCT, CWT).

* Then white box testing will be done by developers.

* The code will be compiled and compressed.

Then we will be getting a file called Build.

* All this will be happened in development server.

* The Build has to be installed from development server to testing server.

- * It will be done by the installation team (or) testing team (or) development team, depending upon project to project.
- * Once build is installed to testing server, Testing team will perform different type of testing depending upon the content of the build.
 Ex: For new Features, FT, IT, will be done for old Feature regression, For defect Fixes retesting will be done
- * While doing the testing, we find the defects that will be raised to the developer.
- * Development team will reproduce the defect and they will fix it in the local System and it is saved in the repository again.
- * Same process will be repeated for every build, until we get the final Stable build
- * once the testing is completed, we will give the application to the customer for Acceptance testing.
- * Customer will do acceptance testing in the testing server or Customer will have their own Server called user acceptance testing server(UAT)
- * once everything is fine Application is released to End users that is for production server. This is called production release (or) GO-LIVE

* What is Release?

Ans: Starting from gathering the requirement, developing the application, testing the application finally releasing it to the end user is called one release.

That is called production release or GOLIVE OF an Application

* What is a build?

Ans: The compiled and compressed format of source code is called build.

Build can also be called as different versions of an application.

A build contains different formats like Compress and Archive.

Compress Format:

- * • zip
- * Multiple files can be made as single file
- * Size of the file will be almost same

Archive Format:

- * • Jar (Java archive), war (web archive)
 - * Multiple files are made as a single file
 - * Size of the file will be almost same
- Note: Maximum in a company, build will be in archive format

* What does build contain?

Ans:

A Build contains

- * New Features
- * Old Features
- * Defect Fixes

It depends upon the contents which developer will add inside a build, and it varies from build to build

Note: Initial build is known as unstable build and

Final build is known as Stable build

* What is test Cycle?

Build is given by developer 1 aprial

> It is effort or time spent to test the application

Once the build is given. The duration of each test cycle can be days or weeks or months, depending upon the build and Application size

Retesting and regression testing with respect to

Build and release:

A Single Application can have Single Release

(or) Multiple release

* A release can have multiple builds

* The final stable build will be released to production server

* The first build of first release, we cannot do retesting and regression testing. (Because defect fix is not there and old features are not there)

* For all other remaining builds, we can do Re-testing and Regression testing based on the situations.

Smoke testing (or) Build Verification testing (or)

Confidence testing:

It is testing the basic/critical Features of An Application before we do thorough testing like functional, integration, System testing

* Why we do Smoke testing?

Ans: * Test engineer will get Confidence on basic Features is working fine or not.

* If we find the defect in the basic features only, that Create a good impact on the testing team

* If there are any blockers or Critical defect found in the basic features, developer will get More time to fix the defect

* When we do Smoke testing?

Ans: * Once the build is given by the developer, We Start doing Smoke testing first

* For every build given by the developer we Start with Smoke testing First we assure Basic feature are working fine

* When time is less for testing team to do the testing, we will do Smoke testing and if customer is fine, we will release to customer

* Difference b/w Formal and Informal Smoke testing

Formal Smoke testing

- * We have documentation (Test cases)
- * We have a proof.
- * It has a procedure

Informal Smoke testing

- * There is no documentation
- * There is no proof
- * There is proper procedure

Smoke testing

- * It is testing the basic Critical Features only
- * It is Shallow and wide testing
- * It is done on the Unstable builds
- * only positive testing is done
- * It is done on initial Builds

Sanity testing

- * It is testing new Features and bug Fixes
- * It is narrow and deep testing
- * It is done on the Stable builds
- * Both +, and - testing are Done
- * It is done on last builds
- * It is the subset of regression testing

Note: In Smoke testing:

1. We check only positive Scenarios
2. We check only basic / Critical Features
3. From FTG / ITG, STC, take only positive Scenarios from all basic Modules and create a new test case document called Smoke test case
4. Since we do Smoke testing for every Module, we can go for Automation
5. Both Smoke and Sanity are partial testing

Adhoc testing (a) Monkey testing (b) Gorilla testing

Testing the Application randomly without following any formal documents like requirement test case

* Why we do Adhoc testing?

Ans: An Application can be used by different set of people or end users.

like:

1. children ✓
2. A person with happy mode ✓
3. A person with depressed mode ✓
4. A drunken person ✓
5. Imitator ✓
6. Mad ✓

When we are not able to get the defect through our regular testing like FT, IT, ST we can think about adhoc testing so that we can get more defects.

* This is the best testing for gaming Application. Because while playing games usually children will click where ever they want without their knowledge.

* This testing is mandatory for gaming Application. For other Application we will do testing based on Customer request or based on Situation.

* When we are not getting much defects with Functional integration and System testing think about adhoc testing we will definitely get defects.

* ^{In}All the defects we found through adhoc testing need not to be fixed by the developer unless it is crashing the Application.

Ex:

1. login to gmail from chrome, login to gmail from Firefox. logout from Firefox, continue using chrome
2. login to gmail from chrome, login to gmail from firefox. change password from Firefox. Continue using chrome
3. Instead of left click, do right click for buttons
4. In notepad, press space bar continuously for 3 to 4 hrs
5. press all buttons at the same time (Login and cancel)

- Q) What are the things we expect in gaming application?
- * Visuals Should be excellent (graphics)
 - * Audio Should be Very Good
 - * Synchronize of Audio and Video
 - * Control options
 - * Opponents (Multi player)
 - * Challenge (level1, level2)
 - * Scores
 - * Response (performance)
 - * Entertainment

Note:

- * In adhoc testing, maximum we do negative Scenarios.
- * After Smoke testing, FT, IT, ST. we think about Adhoc testing
- * Game testing will be done through manual testing usually. Because graphical content cannot be automated.

Exploratory testing:

Whenever Application is given, but there is no requirement for it, then we go for Exploratory testing

- * Here we find Scenarios based on our understanding Convert them into test cases

* What are the drawbacks of Exploratory testing

- * It will lead to lot of miss understanding
- * It may lead to fight between developer and Test Engineer
- * We may misunderstand a defect as a feature
(a) a feature as a defect
- * Because of no clarity, we may waste lot of time

* How to overcome the misunderstanding

- * Communicate closely and politely with all team members. (developers, BA, Managers etc)

Note: leave Ego and let's go and work along with team members. This for my survival

↳ purpose

Compatibility testing:

Testing the Application with different hardware and Software platforms is called Compatibility testing.

* Why do we do Compatibility testing?

To ensure that Application is working for multiple platforms because there might be different types of users

↳ goal

* To check whether the Application is consistently working in all platforms or not

↳ problem

* Types of Compatibility testing:

Software compatibility testing:

Testing An Application in different software platforms is S/w Compatibility testing

Ex: In different OS like

Windows (W7, W8, W8.1, W10 etc)

Linux (Ubuntu, Fedora etc)

MAC (MACINTOSH) (Tiger, Lion, Leopard)

All above in for PC

Below is for mobile:

Android (Mobile)

Alpha, beta, cupcake, donut, eclair, Froyo, honeycomb

Icecream Sandwich, KitKat

iOS (Mobile)

* Hardware Compatibility testing:

Ex: In different processor like - Intel, AMD

* In different Mother Board like - Intel, ASUS

* In different RAM

* In different ROM

* In different Hard disk - bits, byte, Kb, Mb, Tb, Zb

* In different Graphics card - nvidia

Mobile Compatibility testing:

* When we test An Application for different Operating System and also for each different Versions like KitKat windows, iOS and for each brand of mobile

(Ex: Samsung, redmi, nokia etc) for each brand
for different Models like A series, M series
Galaxy SAMSUNG, Xperia XA, C5, Z5 for SONY
V15, V17, V19 for VIVO

* Browser Compatibility testing:

Testing an Application in different browsers

Ex: Chrome, Mozilla Firefox, Internet Explorer, Opera, Netscape Navigator, DuckDuckGo, Safari

* What kind of bugs we find in Compatibility testing:

1. Look and Feel changes (Font and Colour changes)
2. Object overlapping (Login and cancel button sitting on one another)
3. Certain images will not display in certain browser
4. Scroll bar issues (Horizontal & Vertical scroll bar may work in one browser and may not work in other browser)
5. Alignment issues / problems
6. Scattered Content (Chaosness)
7. Certain buttons, links and components may work in one browser and may not work in another browser

Types of Application:

Generally, we have three types of Application

1. Standalone Application
2. Client Server Application
3. Web Application

Standalone Application: This kind of Application can be installed, accessed and used without any dependency on server or internet.

Ex: notepad, calculator, word, calendar

This is the fastest Applications with respect to response time

Client Server Application: In this case we have Software in two categories

1. Client Software
2. Server Software

- * Client Software will be installed by the end users usually from play store.
- * Server Software will be available at the Company location
- * To use Client Software to interacting with each other, we need Server software. To connect to the server software, we need internet.
- * In Client Server Application Static elements will be already stored in the mobile only dynamic elements will be accessed through the internet.
- * This is the fastest compared to the Web Application and slower compared to Standalone Application.

Ex: WhatsApp, Facebook, Gmail, reddit app

Web Application:

In this we access the application through the browser. Here static and dynamic element will load together.

Ex: We can access Gmail or Facebook any other application by entering the URL into the browser

- * We need internet Connection Compulsory
- * Here the browser will act as Client

Globalization testing:

- * Testing on Application which is developed for Multiple language is Globalization testing
- * When the language is changed, the translated Content may not be proper because a machine translator could not understand the exact Meaning of word displayed. This may happen because of below reasons
 1. Machine does not have feeling
 2. It cannot understand exact meaning
 3. It cannot understand exact grammar
 4. " " " " " Spelling

* All these are reasons for wrong translation
So we go for human translator

* If a person is very good with multiple language local and international, we could perform good globalization testing

Ex: Japanese, Chinese, German, French, Korean.

Ex:

Yahoo

Welcome to yahoo - English baavi daa to yahoo - Defect

I love you - English

penu premu nimmu - kelenku — defect

Types of globalization testing:

1. I18N: Internationalization

2. L10N: Localization

Internationalization testing:

Testing the Application whether it displays the Right contents at the right place in the right language is called I18N testing

Ex: When we are testing An Application For English, Hindi, Kannada, Telugu, etc. Sentences Should display from left to right (i.e left aligned)

* whereas as for arabic, urdu, sentences Should display from right to left (i.e right aligned)

Localization:

Testing the Application with respect to the local culture or local standard to that Country or State or region is localization testing

Ex:-

for INDIA

Rs

DD/MM/YYYY

pin code

560097

For US

\$

MM/DD/YYYY

zip code

(CA 12345), (NY 25634), (SJ 58746)

Reliability testing:

Testing the functionality of an application continuously for long duration of time. This can be done by single user. It is to check the consistency of an application.

Ex: Mobile should function properly with one user for long time

Money - mummy - no ✓

Daddy - yes ✓

Daddy - yes - ICICI ✓

Mom - no - SBI ✓

Usability testing:

Testing the Application to check the user friendliness of it is usability testing.

OR

Checking whether Application is easy to use and it gives what we want within less actions.

* Testing the Application, whether it is user friendly or not.

Who should do usability testing?

- 1) End users is the best ✓
- 2) Customer or Client 2nd best ✓
- 3) Other project team members ✓
- 4) Nobody is there, the test engineer has to do usability testing

Ex:-

Touch screen - more user friendly
Keypad

Recovery testing

Testing the Application whether it is able to recover from crash state or not

Ex:-

* Intentionally use multiple Applications at the same time continuously and make the Application CRASH (not responding)

* later chose the Application and open it again and see whether all previous data is still available or not. This is recovery testing

* If all data are available and Application is working normally after crash, then it is not a defect.

* If any data is lost or Application is not working properly after crash, then it is a defect

Aesthetic testing:

↳ UI/UX Design

Testing the beauty of An Application is called aesthetic testing

Ex:-

Colour combinations, Font style, Font size, images

Security testing:

It is to check whether all the DATA and Resources of An Application is accessible only for Authorized users and it is highly secured

Types of security testing:

1) Vulnerability scanning:

It is identifying all the risk areas of an Application with respect to security

2) Security Scanning:

Once risk areas are identified, we have to secure them with additional safety measures that is security scanning

3.) Penetration testing:

Once the security is implemented on the risk areas, check again whether still any risk or data leakage is happening, this is penetration testing

4) Risk Assessment:

Once the security risks are identified, categorize them into High, Medium, Low and take precautions as needed

5) Security Auditing:

This is an internal inspection of application and operating systems for security flaws. An Audit can also be done by line by line inspection of code

6) Ethical hacking: it is a process of attacking a system by an internal employee to see whether security measures which is taken is really worth or not.

7) Posture Assessment: This combines security scanning, ethical hacking and risk assessments to show an overall security posture of an organization.

Ex: 1. https: hyper text transfer protocol secured. This is more secured.

2. OTP, CAPTCHA: login are more secured compared to other logins.

3) Virtual Keyboard: are more secured than physical keyboard.

4) Incognito: is more secured compared to normal browser window.

These are more secured compared other application.

Security testing Scenarios:

1.) From Gmail Inbox page, logout from Application. Click on back button, if Inbox page appears, that is a defect.

2.) Being in Inbox page, copy the URL of Inbox page. Logout from Application, paste the URL and see which page appears. If inbox page appears, that is a defect.

Fuzz testing or Fuzzing: It is a testing where we enter or invalid or unexpected data randomly to check how System behaves and handles it.

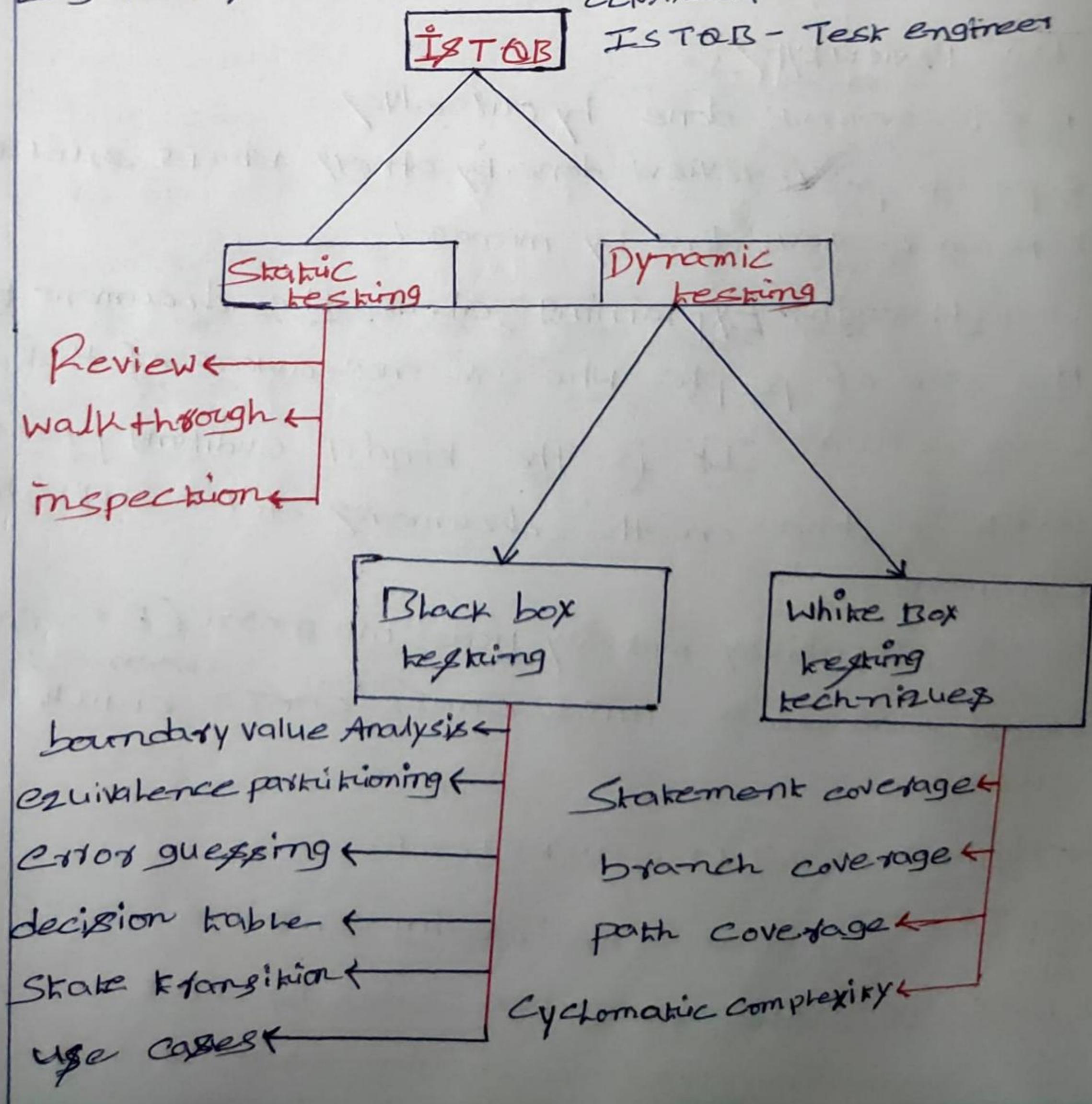
(This is a sort of Adhoc testing)

* It is also to check the Vulnerability of the System. (This is sort of Security testing)

ISTQB

International Software testing qualifications board

* It is a certification for Software testing Engineers.



Static testing:

Testing which is performed on the "document" is called Static testing

* There is no execution here. Checking requirement, Test case, etc documents is Static testing

Types of Static testing:

Review:

finding the mistakes in the document is called review
Mistake can be something which is missing or extra or wrong

In review type:

- 1) Self-review: done by ourselves
- 2) peer to peer: review done by others who is equal to us
- 3) Manager review: done by manager

Walk-through: Explaining about the document to the set of people who are not aware of that

Inspection: It is the kind of auditing process which is done on the documents for a software company

CMMI: Capability maturity Model integrated, (For software companies)

It has five levels CMMI1, CMMI2, CMMI3, CMMI4, CMMI5.

* Highest level is CMMI5 level

SEI: Software Engineering Institute

CMMI is the recognition given by SEI

ISO: International organization for Standardization
it is for Hardware, IBM, HP

Dynamic testing:

Testing which is done on the Application is called dynamic testing, that is one source code or user interface of An Application.

- * Here in this testing, execution ~~takes~~ place
- * Under Dynamic testing we have Black box testing techniques and white box testing techniques

Boundary value analysis: It is a technique where we can apply boundaries for some components in a web page.

* This can be applied for functional testing. That is for the Text Field which is having the requirement

Ex: We can apply Boundary value analysis for Username, password, mobile, email address.

Ex:- P2: Text Field with 6-20 characters

6	20					
X	Y					
x-1	x	x+1	y-1	y	y+1	(3 value BVA)
5	6	7	19	20	21	
-	+	+	-	+	+	
5	6		20	21		(2 value BVA)
-	+		+	-		

Ex2: Amount Text Field with 100-500

99 100 101 499 500 501 (3)

99 100 500 501 (2)

Ex3: 50 to 100 Text Field

49 50 51 99 100 101 (3 value)

49 50 100 101 (2 value)

49.9 50.0 50.1 99.9 100 100.1 (Decimal value)

49.99 50 50.01 99.99 100 100.01 (Decimal value)

Ex4: Req: phone numbered field Should accept 10 digits

9 10 11
- + -

Equivalence partitioning: It is the technique where we come up with given range of requirement. Here it contains one valid and two invalid data.

* This can be applied for Functional testing. That is for the text fields which is having the requirement in range.

Ex: Req 5-10-20 Text Field should accept between 6 to 10

2 8 20
Invalid v In

Ex2: Req: 5-10-20

2 7 15 30 - Equivalence partitioning

2 7 15 30
- + + -

Error Guessing: It is the technique where we come with all the possible scenarios where we should get error message.

- * Here we have more of negative scenarios
- * What is not covered in BVA and Equivalence partitioning

Ex: P2: Text Field Should accept between 6 to 20 characters (Alphabets only). it is mandatory

1 2 3 4 5 6

@#%&^!+

abcde@

Blank (Empty).

Note: 1) Once I get a Job, I need to apply BVA, EP1 Error guessing while writing functional test cases and while doing functional testing whenever ever possible

2.) These 3 techniques are called as Testcase Design techniques

Path for ISTQB question papers:-

Go to 'kyza.com' → Select ISTQB dumps - Mock tests' → scroll down you will get multiple question paper

(5) In a system designed to work out the tax to be paid: An employee has £4000 of salary tax free. The next £1500 is taxed at 10%. The next £28000 is taxed at 22%. Any further amount is taxed at 40%. Which of these groups of numbers would fall into the same equivalence class?

- a) £4800; £14000; £2800
- b) £5200; £5500; £8800
- c) £22001; £32000; £35000
- d) £5800; £28000; £32000. Answer

partition 1
4000
partition 2
5500
partition 3 5900, 28000, 32000
33500
partition 4

Decision table: It is the table which helps upto decide how many number of test cases are needed for a given set of requirement. Any complicated requirement can be converted into decision table for say understanding.

Ex: ISTQB question paper 1 (Qno 10)

State transition diagram: It is a technique which is used to come up with the flow of an application in a easy manner. This can be used in system testing and integration testing.

For atm machine this diagram helps a lot.

Use cases: use case testing is the functional Black box testing technique that helps us to identify the scenarios that whole System can be tested.

→ Here we have a diagram representing in the form OF ACTORS And ACTIONS

→ Each Actor can have different Actions

→ Actor is a user And Actions can be modules where the access is given

Characteristics of use case testing:

→ Use case can capture interaction between Actors and System

→ Actors represent user and interaction is that each user take part into

Test Strategy: Test Strategy is the Approach to test the Software Application \rightarrow $\text{QAT} \rightarrow \text{SQA}$
 \rightarrow Based on the Situation, one Approach Should be Selected.

Test Plan:

Based on that Selected Approach, we plan how to take it forward

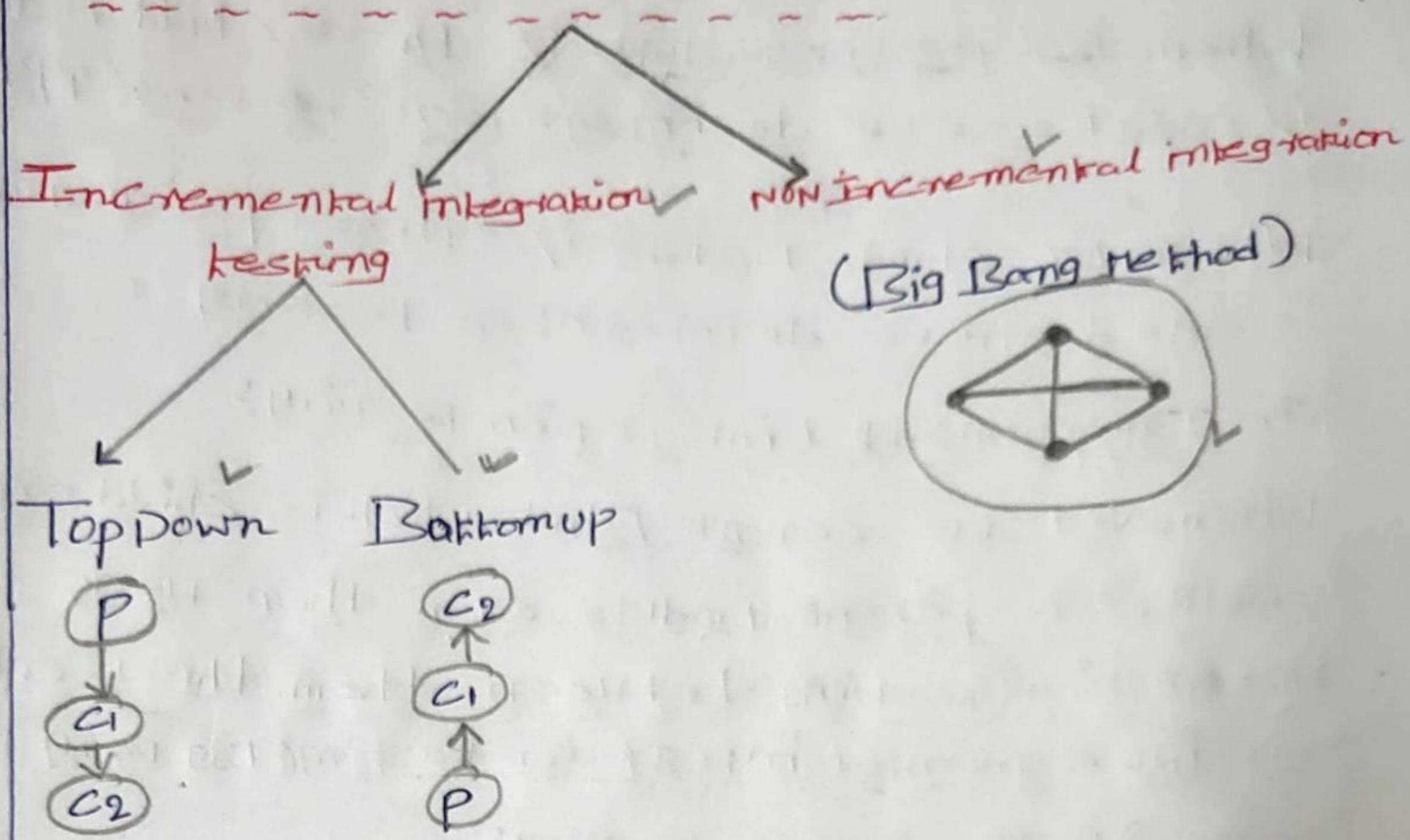
Difference b/w functional and non functional testing?

Functional testing	Non Functional testing
Functional testing	Performance testing
Integration testing	Compatibility testing
System testing	Globalization testing
Checking for single user and single platform and single language focus on normal people	Security testing Accessibility testing Usability testing Reliability testing Checking for multiple users 1. Multiple platforms 2. Multiple languages

Integration testing: Testing the data flow b/w two or more dependent modules is called integration testing

Ex: In gmail, if we send a mail, the send mail should be displayed in sent items
 \rightarrow Checking whether it there in sent items or not is a integration testing

Types of Integration testing:



Incremental EX:

~~~~~

ATM: Insert card > pin > language > Account type > enter amount > withdraw

Flipkart: Search > cart > buy > payment

### NON Incremental:

Gmail: Compose >  
Inbox >  
Setting >  
Message >

### Incremental EX:

Gmail: Signup > login

Incremental integration testing:

Whenever parent module is there, child module is added and we do integration testing (TOP DOWN)  
(OR)

Whenever child module is there, parent module is added and we do integration testing (Bottom up)

Non Incremental integration testing:

Whenever we can't find which is child and which is a parent module, even then we do integration testing between them, this is called non incremental integration testing (Big Bang)

Sandwich Integration testing:

It is combination of Incremental and Non-Incremental testing

Ex: Facebook: Signup > Login (Incremental)

Facebook: notification (Non Incremental)

Stubs: parent module is there But child is NOT there

→ Then I will use Stubs as Dummy child module

Drivers: parent module is NOT there But child is there

→ Then I will use Drivers as Dummy parent module

Stubs and drivers (Both are dummy modules)

Stub: Acts like a child module when child module is not present

Driver: Acts like a parent module when parent module is not present

Developers will create stubs and drivers when Real module takes lot of time to develop

Ex: For Complex Application

Hospital, Airways, Navy, Submarine, Space Application takes lot of time to develop

→ Dummy module will be used <sup>only</sup> for Testing purpose

It is not given to public or customer

→ Once real module is ready, Stubs and Drivers will be removed and Real module will be given for testing

Types of Regression testing:

1. Unit Regression testing ✓

2. Regional Regression testing ✓

3. Full Regression testing ✓

Unit Regression testing: Whenever a new feature is added for existing module and the impact is only for that particular module, we can also call it as unit.

→ When we do regression testing for that unit, we do not do regression testing for any other unit that is called unit regression testing

Ex: Flipkart: electronic - discount charged unit

### Regional regression testing:

Whenever there is a change in one module and that module is having dependency with other few modules not all the modules, then we do regression testing for all the dependent modules. We call them as regional regression testing.

Ex2: Gmail: login, compose, inbox, sent, draft

### Full regression testing:

Whenever there is any change in the application and because of that change if we have an impact on the whole application then we do complete testing again for whole application.

→ This is called full regression testing.

Ex3: Flipkart:

Electronics - discount changed - unit

" " - Groceries - new offer - regional

All products - new offers - full

Note: \* Usually in initial builds (not first build) - we perform unit regression testing

\* In the middle of the project, we perform (Regional) regression testing

\* At the end of the project, before giving application to customer, we perform full regression testing

## RTM: (Requirement Traceability Matrix):

It is a document which is prepared to check that Every requirement has at least one test case or not.

### Types of RTM:

| Ex: Business requirement | Test scenario   | Test case                                           | Defects         |
|--------------------------|-----------------|-----------------------------------------------------|-----------------|
| BR <sub>1</sub>          | TS <sub>1</sub> | TC <sub>1</sub> , TC <sub>2</sub>                   | DO <sub>1</sub> |
| BR <sub>2</sub>          | TS <sub>2</sub> | TC <sub>2</sub> , TC <sub>3</sub>                   | DO <sub>2</sub> |
| BR <sub>3</sub>          | TS <sub>3</sub> | TC <sub>1</sub> , TC <sub>2</sub> , TC <sub>3</sub> | DO <sub>3</sub> |
|                          |                 | TC <sub>1</sub>                                     | Nil             |
|                          |                 | TC <sub>2</sub>                                     |                 |
|                          |                 | TC <sub>3</sub>                                     |                 |

### Forward Traceability Matrix:

Mapping from BASE document to DERIVED document

Ex: Req → Scenario → Test case

### Backward Traceability Matrix:

Mapping from DERIVED Document to BASE document

Ex: Test cases → Scenario → Req

### Bi-Directional Traceability Matrix:

Mapping from DERIVED document to BASE document

And mapping from DERIVED document to Another DERIVED document

Ex: Scenario → Req

### Types of WDT:

1. Statement testing
2. Branch / Condition testing
3. path testing
4. loop testing

## Types of BBT:

- \* Functional testing
- \* Integration
- \* System
- \* Smoke
- \* Adhoc
- \* Exploratory
- \* performance
- \* Compatibility
- \* Globalization
- \* Accessibility
- \* Usability
- \* Security

## Types of functional testing:

FT, IT, ST

## Types of non-functional testing:

Performance, compatibility, globalization, usability  
Security, accessibility, reliability

## Defect seeding:

Injecting the defect in the Application by developer to check the efficiency of test engineer if called defect seeding  
→ This is done based on the request of manager.  
→ If Test Engineer is not working properly or performance, then defect seeding will be done secretly

### Defect masking:

Whenever one defect is hiding another defect that is defect masking

TE: Gets Build 01

Login-A - Defect raised (Masking defect)

ComposeB

(Masked " ")

Login defect is hiding Compose defect in build 01

### Build 02:

Login - A - Retesting

Compose-B - Defect

Defect leakage: The defect which is not found by the Testing team but it is found by the customer (end users) it is called defect leakage.

\* For a good testing team defect leakage percentage

Should be "0"

### Defect Triage Meeting / call:

Manager arranges a meeting with all the development team and testing team and discuss about the existing defects which are not fixed because the defect age should not be increased and defect status should be moved for further level until the defect status is closed

\* This meeting is known as defect triage meeting. It is the main part of Agile meeting

APART from closed STATUS, ALL ARE EXISTING

### DEFECTS:

(New, assign, fixed, reopen, cannot reproduce, can't fix, deferred, invalid, off, duplicate)

- \* Main intention of this is to close the defects as early as possible.
- \* If defect is raised on 10th May and today is 19th May, age of this defect is 9 days

### Defect Density:

Defect density = Number of defects / Total size of the project (LOC/Duration) <sup>(Valid)</sup>

#### Ex1:

$$DD = 80 / 5000$$

$$DD = 80 / 100 \text{ (Days)}$$

$$DD = 100 / 2000 \text{ (LOC)}$$

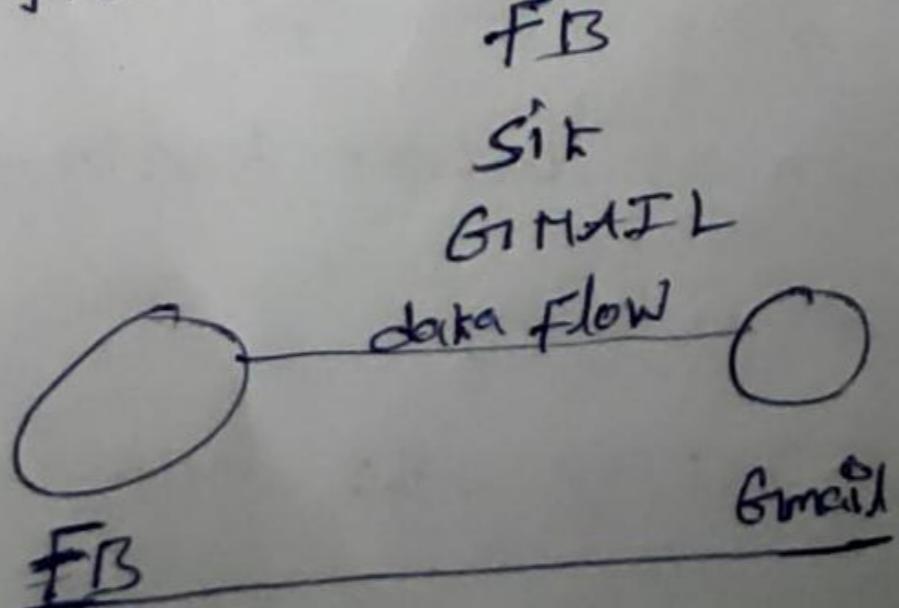
$$DD = 100 / 6 \text{ months (Duration)}$$

Number of defects means the defect which are considered as valid defect not all the reported defect

Latent Defect: It is a defect which is found once application is released to production server and that defect is there in production server for long time.

### SIT: (System Integration Testing):

Testing the data flow between 2 or more systems



Test plan: Test plan is a document which is prepared for future testing activities.

- \* It is usually prepared by test head or manager.
- \* Test engineer can also involve - if there is any support needed for test head or manager.

Test plan contains different sections or attributes:

Objectives: This section tells about the purpose of preparing the test plan.

Scope: This will say the limitations.

- 1.) Features what to be tested (Ex: GMAIL)
- 2.) Features what not to be tested (FB, INSTAGRAM)

Schedule and Milestone: This section will tell which activity has to done first and which activity has to done next.

\* It is just like a time table of the project.

Ex: Req - WTC - ETC AT (Milestone) - PR (Milestone)

Exit

Req (1 April) > Write Test cases (20 April) > Execute Test cases (10 May) > UAT (30th May) > Go LIVE (5 June)

Entry and Exit Criteria: This section tells about when to enter and when to exit each type of testing.

Entry Criteria for FT:

WBT Should be done

Build has to be installed to testing server

Exit Criteria For IT: Pending defects Should be lessed or equal to:

0 Blocked ✓

1 Critical ✓

10 Major ✓

20 Minor ✓

Entry Criteria For IT:

0 Blocked ✓

1 Critical ✓

20 Major ✓

20 Minor ✓

Exit Criteria For IT: Pending defects Should be lessed than or equal to:

0 Blocked ✓

0 Critical ✓

5 Major ✓

10 Minor ✓

Entry Criteria For ST:

0 Blocked ✓

0 Critical ✓

5 Major ✓

10 Minor ✓

Exit Criteria For ST: Pending defects Should be lessed than or equal to

0 Blocked ✓

0 Critical ✓

0 Major ✓

5 Minor ✓

Defect tracking: This section will tell about whenever a defect is found, how to track the defect and which is the tool we are using to track the defect. And what are the terminals given we are using to raise the defect.

Ex:

ALM - tool

P<sub>1</sub> or P<sub>2</sub> or P<sub>3</sub> - priority

Blocked, critical, major, minor, trivial - severity

Open, confirm, in progress, resolved, verify - status

Assumptions: This section will tell about what are the assumptions we have during this project.

Ex: All employees will be there in the office daily

Risk: This section will tell about what are the possible risk happen during the project.

Ex: All employees are not in office. leave/quit  
Contingency plan or mitigation plan or backup plan

This section will tell about how to overcome the risk which occurs during the process.

Ex: Arrange regular KT among the employees when everyone are there in office, before risk occurs.

Q A

Roles and responsibilities: This section will contain what are the roles and responsibilities for team members.

A. Ex: Roles and responsibilities for test engineer:  
(Manual testing)  
Going through the requirement and understanding the requirement and identifying the scenarios And writing the test cases, prepare RTM, and execute test cases when the build is given. Find the defect and raise the defects

b) For Automation test engineer: we should perform above task plus we should write test scripts and execute test scripts by using the selenium tool. Manage test script

Environment: This section contains which platform we are using for testing purpose

Ex: windows 10, 8, & chrome browser

Deliverables: This section will tell what are the documents that should be prepared for the project.

Ex: Test plan, Test cases, RTM, Defect reports etc.

Graphs and Metrices: This section contains the graphs and metrices that are prepared for project

Acceptance testing: It is a testing which is mainly done to check the business scenarios of the application which is done by the customer.

Ex: For Courier business - placing a Courier, shipping Tracking, delivery date, weight, amount, agent, status return option, O.P. address is tested.

But Cancel scenario is forgotten by Test engineer

\* This is tested by customer and he may find defect.

\* If customer is checking in company it is Alpha

\* If he is checking in his place, it is Beta

Why acceptance testing is done?

\* This is mainly done to get a "confidence for customer before he releases the product to the end users"

\* Because of business pressures. Software

Company might be releasing the application to the customer with some defects. To ensure that is not there, customer will do acceptance testing

\* If the product is released to the end user without checking the business scenarios, it will affect the customer business. To avoid this.

Acceptance testing has to be done.

\* We may forget some of the business scenarios to test and those scenarios would be tested by the customer

### Note:

If the customer finds the defect during the Acceptance Testing. That is bad for test engineers. So before giving application to customers, we should think all those business Scenarios and find defects.

Difference between Alpha and Beta testing

| Alpha testing                                                                                | Beta testing                                                                                    |
|----------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------|
| Alpha testing is performed by tester who are usually An internal employee of An organization | Beta testing is performed by the Client or end users who are not an employee of An organization |
| Alpha testing is performed at the developer site (S/w company)                               | Beta testing is performed at the Client location or end user of product                         |
| Alpha testing involves Both black box and White Box testing techniques                       | Beta testing typically uses Black box testing Technique                                         |
| Alpha testing requires lab environment or testing environment                                | Beta testing is real time environment                                                           |
| Virtual environment                                                                          | Real time environment (production)                                                              |
| Done at offshor (Company)                                                                    | Done at onsite (customer place)                                                                 |
| Alpha testing is done under controlled environment                                           | Beta testing is done under uncontrolled environment                                             |
| It is closed for public                                                                      | It is open for public                                                                           |

Hyd (Company) - dev and test.... offshor

Client (pure) ... customer place .... onsite

Approaches of acceptance testing (or) Who can do AT?

Approach No1: BA (or) IT Engineers of customer will do Acceptance testing at customer place. Beta

Approach No2: Employees or test engineers of customer will do Acceptance testing at customer place. Beta

3: Test engineer of Software Company will do Acceptance testing at customer place. Beta

Approach No4: TE (or) BA of Software company will do Acceptance testing under the control of customer at developer place. Alpha

✓ Hot Fix / incident management System:

When the Software Company release the product to customer / end user. If the end user / customer finds any defects in the software Customer will raise an incident to Software Company and Software Company Should fix the defect as early possible and gives back to the customer. This situation is called hotfix

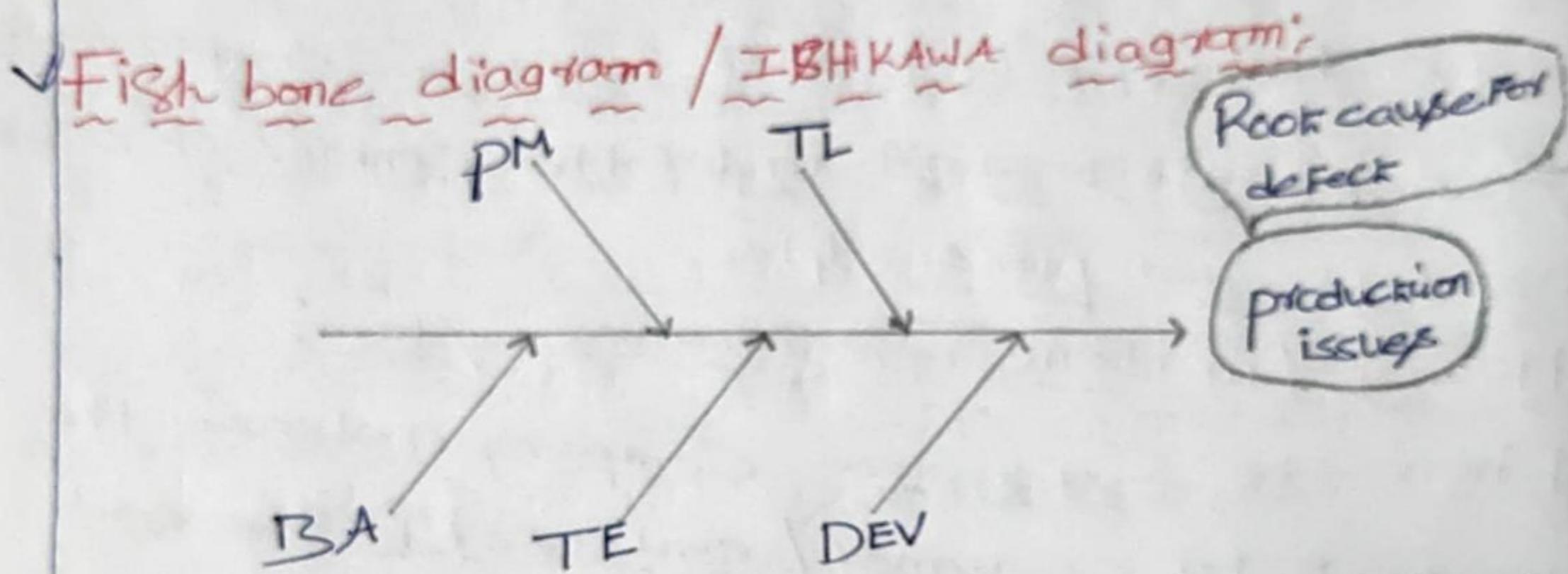
IF Any problem is find by the customer or End user in the production server. It will be raised as the incident to the software company. Every incident is raised in the form of ticket and every ticket will have a priority

The commitment between Software Company and customer is called Service level agreement.

(SLA)

Ex:-

| Priority of ticket | Time given to fix |
|--------------------|-------------------|
| P <sub>1</sub>     | -3 hrs            |
| P <sub>2</sub>     | -10 hrs           |
| P <sub>3</sub>     | -1 day            |
| P <sub>4</sub>     | -3 days           |
| P <sub>5</sub>     | -10 days          |



- \* Whenever there is a production issue, All the Team members will gather together, and they Discuss about reason for production issues That is documented in the form of Fish bone diagram. The main purpose of this is to find the root cause of the production issue.
- \* It is done after HOT FIX. Otherwise it is not done.
- \* It is mainly for future purpose. because of this Employees will work more effectively and production defect (issues) can be avoided.

## Short Term Release / Interim Release:

Between two planned releases of an application, sometimes customer gives an unplanned release which is done in short duration. It is called short term release.

AAA - planned - April

AAA - unplanned - April

BAA - planned - May

## AGILE MODEL - Types of Variants:

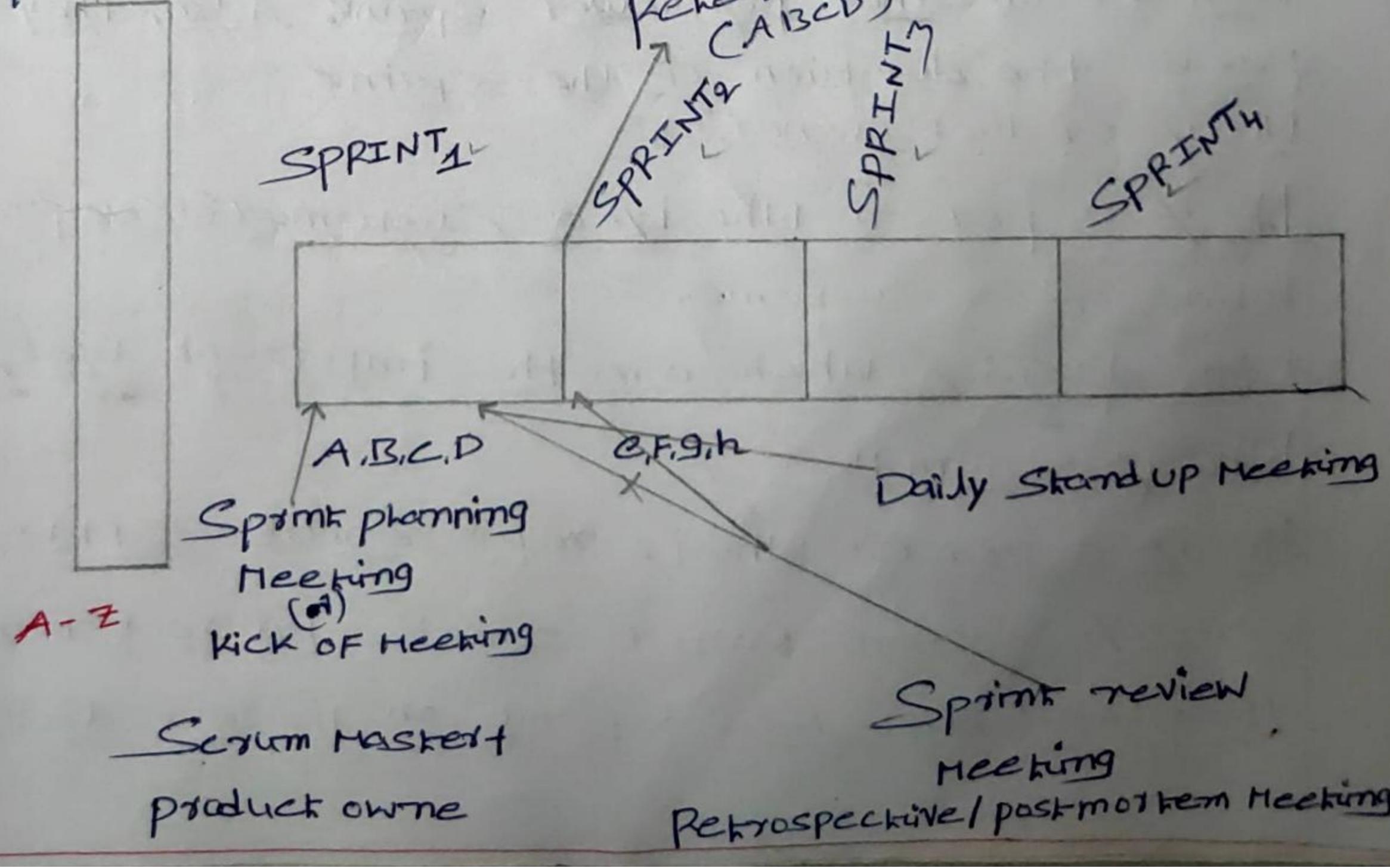
SCRUM, XPC (Extreme Programming), RUP (Rational Unified process), RAD (Rapid Application development), Kanban process etc

SCRUM - 80% 90% Industry uses this ✓

### AGILE (SCRUM):

PO (EPIC)

(product Backlog)



EPIC: It is complete set of REQ Given by customer

A for

→ EPIC is also called product Backlog

→ user Story / Story card - It is a part of the REQ

A - user story ✓

B - " "

C - " "

D - " "

Sprint: Sprint is the duration / time taken

To work on 1 or more user stories / Story Card

\* Each Sprint can be either 2/3/4 weeks. It depends upon Customer Decision

Sprint Planning Meeting:

It is a meeting which is conducted before sprint

Starts

\* In This Meeting product owner and Scrum master will be involved

\* They will decide which user story has to be worked on that particular sprint. Also, they will decide the duration of the sprint

Who is product owner?

He is a person who is a customer or representative of a customer.

\* He decides which are the important user stories

Who is Scrum Master?

He is a person who is representing software company. Scrum Master can be project manager / Dev head / Test head depends on project decision

## Daily Stand up Meeting:

- All the team members will gather together like PM, dev head, test head, test engineer, developer, BA, Customer can also involve in this meeting
- \* Here, each employee should give work update every day
  - \* If the meeting is in the morning, what was done yesterday and what will be done today. Should be updated
  - \* If the meeting is in the evening, what is done today and what will be done tomorrow should be updated
  - \* This is very much helpful for a better communication between team members

Sprint Review Meeting: In this meeting, Scrum Master will check whether all the user stories are completely developed and tested and is it ready to release to customer or not

This happens at the end of Sprint

Sprint Backlog: It is a user story which is not completed in that particular sprint. That is carried forward to next sprint

Ex: If a user story is not completed in Sprint 1, It is a backlog which is taken from Sprint 1 to Sprint 2

Product Backlog: It is the user stories which is taken for a new sprint

Ex: e,f,g,h are from product backlog for sprint 2

In the same way, till all the user stories are completed, multiple sprints will be going on And full application will be ready

### Advantages:

- \* From this model, work is getting completed very fast.
- \* It is the most successive model currently.
- \* We can work more within less time.
- \* Communication is very good between company and customer.
- \* We can handle any pressure.
- \* It is best model for existing project.

How To Explain About Project with respect to Model, build, release?

- \* Chat — vandv — release 1 (6 months)  
30 builds
- \* Status — Agile model — release 2 (4 weeks)  
Sprint 1 — 6 builds
- \* Call — Agile model — release 3 (4 weeks)  
Sprint 2 — 5 builds
- \* WEB — Agile model — release 4 (4 weeks)  
Sprint 3 — 4 builds
- \* Video call — Agile model — release 5 (2 weeks)  
Sprint 4 — 3 builds
- \* Payments — Agile model — release 6 (2w)  
Sprint 5 — 7 builds