20/10/23 TESTING

SDLC

House or Flat or Real time example

Software Development life Cycle.

It is a procedure / process to develope Software Application. It Contains different stages / phases

* Requirement Collection / Gathering :-

↓

Usually Customer Gives requirements in the form of CRS / BRS. It will be Converted into SRS, FRS, FS by BA. BA will hand over the requirements to the Sw Company. This whole process is called as Requirement Collection / Gathering.

CRS - Customer Requirement Specification
BRS - Business Requirement Specification
SRS - Software Requirement Specification
FRS - Functional Requirement Specification
FS - Functional Specification.
BA - Business Analyst.
↳ Acceptance Testing (done by BA, Customer)

* Feasibility Study :-

In this stage we have, BA, Finance, HR, PM, Architech.

In this stage all the team members will gather togethes. Here BA will Explain requirement to everyone. HR will thinks about People and hiring process. Finance team will thinks about Budget, profit and loss. Architech will thinks about design and technology (Java, .net, Python, CSS, JS, AJS, etc...). PM (High responsibility) will take final decision and responsibility. Once everything is finalized

If everything is fine, mutual agreement will be done b/w Sw Company and customer/client. So this is the decision making stage. -> Before date Company should handover project. Else penalty will be provided around 25% of the money.

* Design:-

In Design we have two type

HLD -> High level Design

LLD -> Low level Design

HLD is like blueprint. It tells about external architecture of an application.

LLD will tell about internal architecture of an application.

Both HLD and LLD are documents. It is done by architects and senior developers.

* Coding / Implementation / Development

In this stage, All developers will write source code based on requirements and design. Source code is of 2 types. as front end source code and Back end source code. When we know merge both front end (Web tech. Html, CSS, Javascript) (look wise) and Back end (Java, SQL) (Work wise) that becomes full stack development. This will be done in development server.

* Testing:- -> once src is compiled, it will be converted into UI/GUI Format

once application is ready by developers, it will be installed to testing server. All test Engineers will test an application based on requirements given by client to find bugs/defects.

Testing Engineer will do Testing on UI/GUI.

Static app, Dynamic app

Testing will be done on testing server.

Till all the bugs are fixed by developers this process will be going on.

Once Everything is completed i.e, if application is matching the requirements, Application will be installed to the next server.

* Installation / Deployment :-

Once application is completely ready it has to be made available for end users (public). Moving an application from Testing server to Production Server is called as Installation / Deployment.

Installation will be done by sw Company based on customer request. / Senior developer / Server. Test engg.
Date of Installation will be decided by Customer.

* Maintanance

once application is used by end users / public, its support / service has to be done. That is called Maintanance. (support given to sw application)

Maintanance will be done on production Server. During maintanance, below things will be done

* Defects will be fixed
* Adding, modifing, deleting a feature

Maintanance will be done by sw Company.

Server:
Any computer which has capacity to provide access for lakhs of people called server.

'Free' Support - It is for some duration initially

Paid Support - later it will be paid maintenance

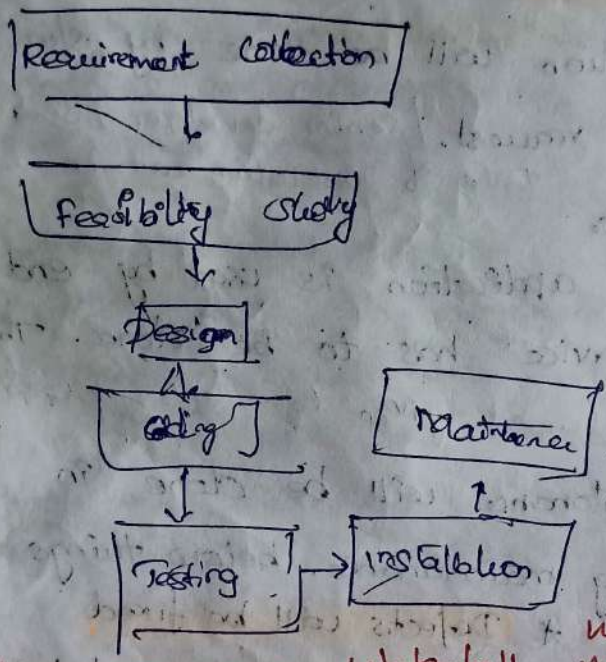
Maintenance is done by Support team / Production Support team.

23/10/23 : Types / Models of SDLC.

Waterfall model
Spiral model

Waterfall Model:-

It is one of the oldest and traditional model of SDLC.



1) When do we go for Waterfall model?

When customer freezes (does not changes) the requirement

For any short term project (↓ than 1 year).

For developing simple application.

Eg: Calc; Notepad, alarm., etc.

2) What are the advantages of Waterfall model?

We can get stable application.

There will be no disturbance to the team members. if there is change in requirements

What are the draw backs of Waterfall model:

Requirement and design are not tested

Testing happens only after Coding.

During initial days of SDLC Process developers were doing Testing (Currently it is done by Test Engineer)

Requirement and design are not tested.

Testing happens only after coding.

If there is any requirement change, lot of rework will happen.

Why requirement Changes & who will change the requirement

Customer will change the requirement due to

① Competition
② To stay in the business
3 To upgrade the technology.

If developers do testing what will happen / why developers should not do testing?

Developers are ~~aims~~ to build the application and not to brake an application.

Developers can't find their own mistakes.

Even though mistake is their, developer will hide it.

Developer might be over Confident in what they do (the tested Source code is Correct. & he will not check it properly).

25/10/23

Spiral model:

Spiral model / Iterative model. It is also one of the traditional model.

When do we go for spiral model?

When customer gives requirement part by part / Stage by Stage (Module by module).

When there is a Lot of depedency b/eo modules. Inc

What are the advantages of spiral model?

Customer can see the application partialy and can get confidence.

Requirement changes can be done.

Compare to waterfall model, rework is less.

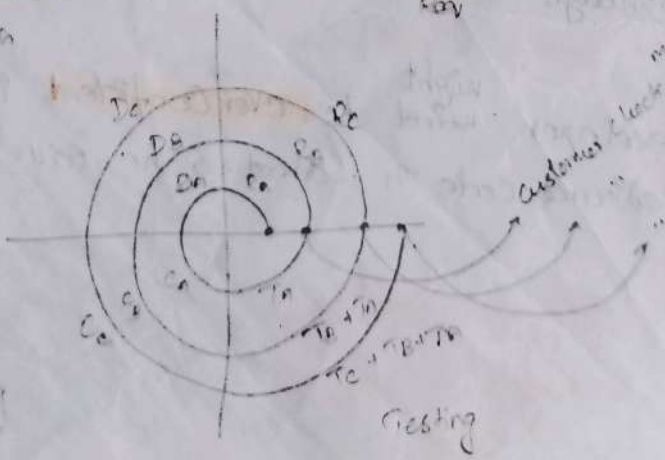
What are the drawbacks of spiral model?

Requirement and design are not tested, testing happens only after coding.

Developers used to do testing before (but currently TE will do testing).

If there is any requirement change, it may delay the project.

Customers can cancel the project, if he is unhappy.



For Example. Log reg as A, Inbox reg as B, Compose reg as C.

Customer will give requirement part by part.

Regression testing is done in spiral model.

In the 1st iteration login module is taken care, if Customer is happy, he will give next part of the requirements like this process continues till end of the project.

Other Eg: Gaming App, Microsoft's windows, elg. project, etc.

26/10/23

V & V model:- (Verification and Validation model).

Verification is also known as QA Quality Assurance

Validation is also known as QC Quality Control.

Both Verification & Validation is done by Test Engineers

Verification is the process of checking "are we building Product right"

Validation is the process of checking "are we building right Product"

When do we go for V & V model?

- When customer need High Quality Products.

For Complex applications.

Eg: Banking, healthcare, Space, Airlines, navy applications, etc.,

For long term projects (more than one year)

Advantages of V & V model:-

Testing is started at Initial stage itself (reviewing documents)

Requirement and design are Tested

Documentation of defects are less

Requirement changes can be done.

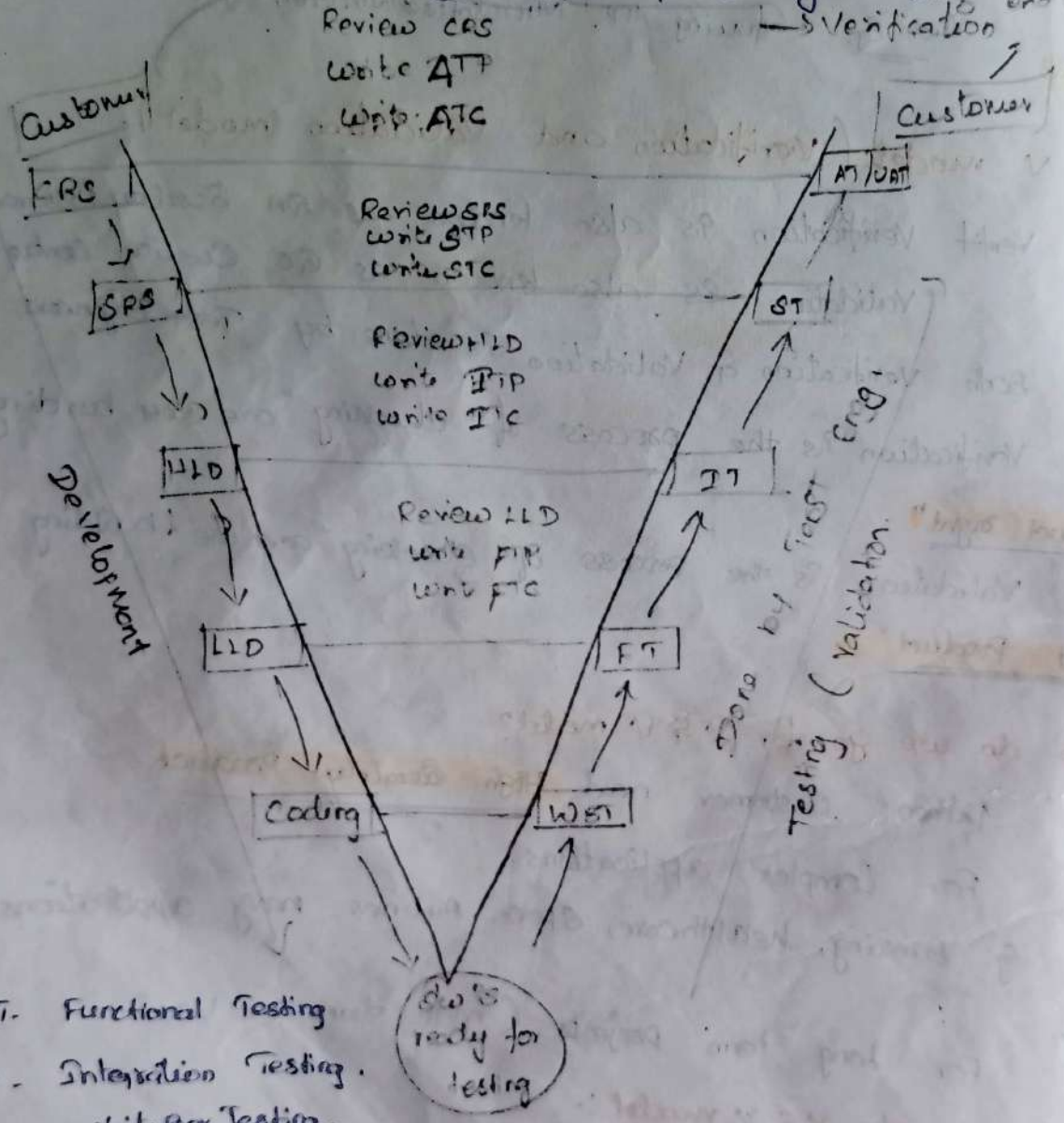
Quality will be high compared to other models

Reworks will be less.

Draw backs of V & V model:-

Documentation works will be more (Test plans, Test cases)

Too much of resources are needed.
(Developers & Test Engineers).



FT. Functional Testing

IT - Integration Testing.

WBT - White Box Testing.

AT - Acceptance Testing.

UAT - User Acceptance Testing.

ST - ~~Sanity testing~~ System Testing.

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 parallely. The left side of the model is done by developers. and the right side of the model is done by Test Engineers.

When the customer gives the requirement of 100 Pages document in the form of CRS. It is converted in the form of SRS by BA. At same time, review of CRS is done by test engineers. If there is any mistake it goes back. if not it will continue the process. The SRS will be reviewed against the CRS to find the defect. Parallely 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 testing is white box testing. This testing is done by the developers, then there will be FT it is done by TE. At the same time the execution of the test cases are also done.

After FT, The IT is done, later ST and then AT is done by the customers, 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.

27/10/23

PROTOTYPE MODEL:

Prototype is a dummy model & it is a non working Application

When do we go for Prototype model?

When customer is not clear about the market requirement.

When Slw Company is new to the domain then they go for prototype model.

When developers are new to the technologies (if they want to do any Experimentation they go for prototype model).

When customer & slw company are new to the business.

Domain

Domain is a categorization of slw Application in to different Industry.

- Eg: BFSI (Banking, Financial services and Insurance)
  - Telecom
  - Healthcare
  - E-commerce
  - Education
  - entertainment
  - Gaming
  - Logistics
  - ~~Govt~~

Advantages of Prototype model:

Initially customer can get to know what he gets at last.

Initially developers itself will also come to know what they should deliver on last day.

Requirement changes can be done initially itself.

Initially Investment is very less.

Drawbacks of prototype model:-

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 rythmn of the company.

28/10/23

Note:

Once I get a job, I may be working in any of these models. (or) I may works with the combination of these models. That is called Hybrid model.

(or)

I may works in the new model which is designed based on these models. i.e, derived model.

(or)

I maynot know in which model I am working in.

My a

Accademic Project Linking to SDLC

My accademic project belongs to Spiral model.

Agile:

Whenever Customes wants application, too fast / with in less time and with high Quality, we go to Agile model.

Software Testing.

Testing the functionality of an application based on Customer requirements.

(or)

The intent of finding bugs or defects in an application is s/w Testing.

(or)

Testing is the process of QA and QC.

Manual Testing

Testing the functionality of an application with out using tool.

Automation Testing:-

Testing the functionality of an appliation by using Tool.

Eg: Selinium, QTP/UFT etc.

Seen Scenarios:-

Testing an application in all possible ways is called scenarios.

TASK:

1) WSF Chair (50 scenarios).

2)
1) Verify durability of a chair
2) Verify the material used
3) Verify the dimension of a chair
4) Verify the dimension of weight
5) Verify the dimension of height
6) Verify no of legs in chair
7) Verify chair backrest option

8) Verify all legs of a chair on plane surface

9) check the chair this is compatible for taking rest

10) check it is capable of for 2 person to sit.

11) check the design of the chair

12) Verify the color of the chair

13) verify the maximum amount of load the chair handles

14). Verify the non breaking tendency of chair

15) Check stress of a chair by dropping it from Particular heigh

16) check the chair under different climate

17). check the type of chair

18) check each leg of chair

19) Check the condition when washed in water.

20) Verify if any cushion is provided with chair

21) Verify the paint type

22) Check if there is suppor for hand

23) Check the usability of a chair

24) Check the chair by standing on it

25) Check the chair has wheel.

26) Verify the shape of the Chair.

27) Verify the size of the Chair.

28) Verify the board of the chair

29) Verify the smell of the Chair

30)

Negative Scenario of Chair :-

1. Verify by kicking the Chair
2. Verify by Jumping on the Chair
3. Verify by throwing the Chair
4. Verify by burning the Chair
5. Verify by standing on the Chair
6.

Write a Scenario for Pen / How will you test a Pen

1) Verify the type of pen (ball pen, Gelpen, Inke pen)
2) Verify the Color of Pen (blue, Grea, Red)
3) Check the Ink flow of pen
4) Check the color of Ink
5) Verify the grip of pen
6) Verify the Shape of Pen (Cylinder, triangle)
7) Verify the fragrance of pen
8) Verify by opening the cap
9) Verify the closing the Cap
10) Verify by holding the pen
11) Verify by the Ink is not erasable
12) Verify the thickness of the nib
13) Verify the height of the pen
14) Verify the material of the pen
15) Verify the refil is fitting the pen
16) Verify the ball rotation in the pen
17) Verify the size of ball in the pen
18) Verify the Smoothness of pen
19) Verify by pouring water in position
20) Verify by Smashing the pen

Scenario:

Testing an application in all possible ways is Scenario. We need requirement to write Scenario.

Scenario is of two types

* Positive Scenario
* Negative Scenario.

Testing the application with valid / Expected data / input is positive scenario.

Testing the application with invalid / unExpected data / input is negative scenario.

If we test with +ve scenario it is +ve Testing

if we test with -ve scenario it is -ve Testing.

If we need to find more bugs, We need to find more Scenarios.

If we miss scenarios, we will miss bugs

For Every scenario, there may be (or) may not be a bug.

Eg: There may be +ve scenario with a bug & there may may by +ve scenario without a bug & there may be negative Scenario with a bug & there may be negative scenario ~~without a bug~~. without a bug.

Eg:

Valid	pattern lock	with some one mobile	→ bug.
valid	pattern lock	with valid mobile	→ No bug (Error)
Invalid	pattern lock	with valid mobile	→ bug
invalid	"	with invalid mobile	→ No bug (Error)

Scenario is not a bug / Bug is not a Scenario. Both are Seperate.

Note:-

Unless interviewer ask, do not divide +ve & -ve Scenario.

If they ask scenario / test case for any object or application Ask interviewer to about the requirement for that object / Application.

Test Cases:-

A Test Case is a document which contain all the possible scenarios & explains step by step

Template is a format used to write a test case.

We have different Templates from Company to Company and project to project.

When do we write test case?

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

There are two stages of Test Cases.

1) Test Case preparation / writing

Under the test case Preparation, we have to fill following sections like Header, Body, Footer Section.

In a body section we will fill Expected Result.

2) Test Case Execution:-

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

Actual Result will be written based on Application

If Expected Result & Actual Result matches, STATUS is PASS

If not, STATUS is FAIL. If it FAIL, then it is a DEFECT / BUG.

D/B Scenario & Test Cases

Scenario	Testcase
(i) It will tell all possible ways we can test in the Application	It is step by step procedure to test the application
(ii) Scenario says "what to test"	Test case says how to test
(iii) Scenario is a high level document	Test case is a low level document
(iv) Scenario don't have navigation steps	Test case will tell where the exact defect is present
(v) Scenario takes less time to write.	Test case take more time to write.
(vi) Scenario will not tell where the exact defect is present.	Test case have navigation step.

7/4/23

Types of Testing:-

White box testing — Developer
Black box testing — Testers
Grey box testing — both

Testing the Source Code of an application line by line is called white box testing.

This is usually done by developers.

Testing the UI / GUI of an application is black box testing

This is usually done by Test Engineers.

Testing the Source Code & UI / GUI of an application together is Grey box testing.

This is usually done by a person who knows about coding and Testing. They can be SDET (S/w Developing ment Engineer in Test).

Microsoft Gave the name

Other Names for White box testing:-

~~Unit~~ Testing
Unit Testing
Glass box Testing
Transparent Testing
Structural Testing
Open ~~Closed~~ box Testing

Other names for black box Testing:-

Functional Testing
Behavioural Testing
Closed box Testing

Other names for Grey box Testing:-

All the names of White box & Black box Combined together will be called as Grey box Testing.

Types of Black Box Testing:-

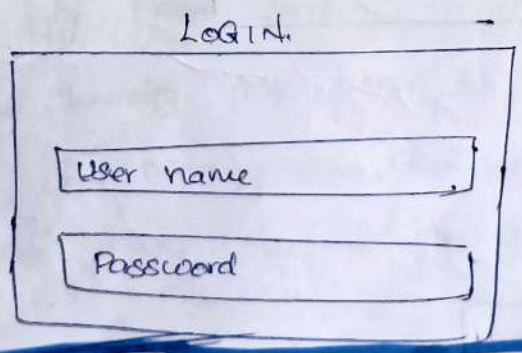
1) Functional Testing / Component Testing / Field Level Testing:-

Testing each and Every Component independently and thoroughly with respect to (W.R.T) Customer requirements is called functional Testing

Eg: Face book Login page.

We perform functional Testing for all Components inside the models of an application.

Eg: (a) Text field / Text Box



Static - Testing the doc related Project is Static to

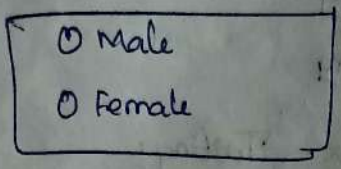
Eg: req doc, Scenario doc, ToC doc, RTM doc etc

Dynamic - Testing the app related to dev ; then it is whenever we do testing on app Dynamic

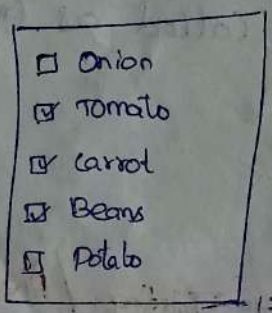
b) Radio Button

when we want to select any one option from multiple option we use Radio Button

Gender



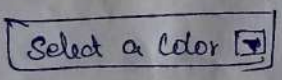
c) CHECK Box



When we want to select both single & multiple option we use Check Box.

d) Dropdown

When we want to select only one option from multiple option we use Radio drop button

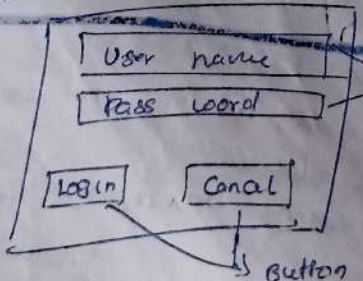


Click ctrl+ Select more option ↓ to select many

e) Link :-

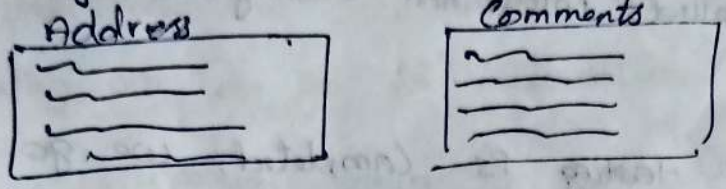
https://www. . . . . .

f) Button :-



g) Text Area :-

It is a bigger box which accepts many texts like Address, Comments etc.



Note:

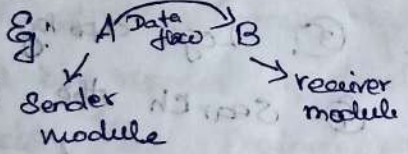
Once developer gives every module, we should do functional Testing. for each module.

After testing each module, we go for integration Testing.

8/4/23

Integration Testing:-

Testing the dataflow blw two / more dependent modules is Integration testing



To do integration Testing, atleast we need two dependent modules.

Eg: I Am sending money to my friend using Gpay. Checking till here is functional testing.

Sent Money should be deducted from Sender (my) account & it should be added in my friends account.

deduction → debit

adding → credit

Checking this comes under integration Testing.

Eg: Signing up in Gmail is functional Testing By using the sign up details am able to do login then it is called. Integration Testing.

Note:

Once integration testing is completed, we go for System Testing.

System Testing:-

Testing the application from end to end Just like a real end user in Testing. Server is System Testing.

Testing server should be similar to production server

Eg: In flipkart application if we ① open the browser ② Enter URL ③ Login app by using User name & Password ④ search the product.
5) Place the order with one payment mode.
6) Logout.

Note:

To do System Testing, we need atleast maximum modules of an application.

After functional testing & integration Testing we will do system Testing.

Maximum defect will be found in functional Testing & integration Testing. If F.T & I.T is not done properly then we get defects in

System Testing.

9/4/23

What are the drawbacks of the manual Testing / Why do we go for Automation Testing?

Manual Testing is time Consuming process. It needs more man power (Human Resources)

Testing is a repetative Job (Due to regression Testing).

Due to this we go for Automation Testing. The tool we used here is Selinium. Some other tools like (Selenium, UFT/QTP etc...)

Retesting & Regretion Testing:

Testing the defect fix / Bug fix of an application is called retesting. / once test Engineer finds the defect, it will be reported to developer, After developer fixes a defects, Test Engineer will test whether it is fixed properly or not. this is retesting.

Testing the impact areas of an application after defect fix / changes done in the application.

changes can be 1) Adding a feature
2) Modifying a feature
3) deleting a feature

Eg: for example I am testing mobile phone, I observed selfi Camera is not working. This is a bug. I will report this to developer. Once developer fixes the bug, I will test the selfi Camera again. This is retesting. After this I will Test all the impact areas related to Selfi Camera. Like Back Camera, Scanner, zoom, flash, video call etc.

Testing all these impact area comes under regression testing.

For example when I am testing the power glass I found Nose Pad is not fixed poopperly. This is a bug. I will report this to developer. Once developer fixes a bug, I will test the nose pad again. This is retesting. ~~After this I will the whole power glass again~~ After this I will Test all the impact area related to Nose pad like frame, Glass, Power of a glass, Scratches in the glass etc. Testing all these impact area comes under regression Testing.

Fixing in the nose
Moving down from nose
itchiness in the ear

Smoke Testing / Confidence Testing / Built Verification Testing.

Testing the basic or critical features of an application before doing the testing like functional Testing, integration Testing & system Testing.

Why ~~do~~ we do Smoke Testing?

To check whether the basic feature is working fine / not.

TO get confidence that the basic feature are working fine

If there is any defect in important feature that can be found earlier and report it to developer so that they will get time to fix the defect fast.

In Smoke Testing we check only the +ve scenario.

When we do smoke Testing?

As soon as the build is given we do Smoke Testing.

For every build we have less time to test the application. That time we do smoke testing first.

Eg: Atom modul.

For Login page of FaceBook module, The smoke testing scenios are

1) verify by opening face book
2) Verify by entering valid user name.
3) Verify by entering valid password & click on login button.
4) Verify by entering valid ph.no & password & click on login button.
5) Verify by clicking forget password & checking whether the page is open.
6). Verify by clicking create new account & clicking page is open.

Smoke testing for remote:

1) Verify by turning on/off
2) verify by raising/decrasing volume.
3) Verify by changing the channel
4) Verify by pressing mute button
5) Verify by pressing recall button
6) Verify by. Clicking on child lock
7) ~~Verify by~~

15/4/23

Note:

Smoke Testing is also called as Health Checks of a slow.

I/B Smoke & Sanity Testing:-

Smoke	Sanity
1) It is testing the basic / critical features of an application	It is testing the new features & Bug fixes of an application
2) It is done on Initial Builds.	It is done on last Builds.
3) It is done on unstable Builds.	It is done on stable build.
4) Here we do testing on Positive Scenario.	Here we do testing on both +ve & -ve Scenario.
5) It is wide & Shallow Testing	It is narrow & deep testing.
	It is also called as ~~sub test~~ Sub Test of regression Testing.

W/y Sanity test is done on new feature?

To Check whether the app works as expected even after adding small changes with bug free.

Diff formal smoke & informal smoke testing

formal smoke	informal smoke
1) It is done with the process	It is done without process.
2) Here we have documentation	Here there is no documentation.
3) There is a proof.	There is no proof

Performance Testing:-

Testing the response time of an application by applying load for an application is known as Performance Testing.

Load: Number of user who uses the application

Response Time: Time taken to get the expected screen based on the user action.

Types of Performance Testing:-

Load Testing:-

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

Eg: If customer RFD say application should be for 1 lakh users, then the designed number of user is 1 lakh.

If we do testing till 1 lakh user, this is called as Load testing.

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.

Eg:

Req :- 100000 users --- response time should be within 2sec.

Stress Testing :- 1,10,000 users --> response time 2 1/3 sec.

Soak Testing :-

Testing the response time of an application by applying load continuously for long duration of time

Eg: We will check the response time for 72 hrs or more hrs by applying load continuously and check response time.

Volume Testing :-

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

Eg) Huge data sharing through ~~share~~ share it, Bluetooth, google drive by uploading more Photos and checking the time taken to transfer the data.

Spike Testing:-

Testing the response time of an application by suddenly increasing the load or suddenly decreasing the load.

Can we do performance testing manually?

Yes, But there are some drawbacks, like:-

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

2) There will be no ~~accu~~ accuracy in the results, we human beings we do not perform action at same time. Because of this, result may vary.

So, we go for automation (Ex: J. meter, Load Runner etc.)

Compatibility Testing:-

Testing the application with different hardware and software platform, is called Compatibility Testing.

Why do we do Compatibility Testing?

1) To ensure that application is working for multiple platform because there might be different types of users.

2) To check whether the application is consistently working in all platform or not.

Types of Compatibility Testing:-

1. Software Compatibility Testing:-

Testing an application (Ex: paytm) in different software platform is also compatibility testing.

Eg: In different O.S like

Windows (w7, w8, w8.1, w10 etc).

Linux (Ubunto, federo etc)

MAC (MACINTOSH) (Tiger, Lion, Leopard, Snow Leopard, mountain lion, Sierra etc).

All above is for PC.

Below is for MOBILE:-

Android (mobile)

Alpha, Beta, Cupcake, donut, eclairs, froyo, ginger bread, honey comb, Icecream, Sandwich, Jellybean, kitkat lollypop, marshmallow, nougat, oreo, pie,

IOS (mobile).

2) Hardware Compatibility Testing:-

Testing an application on different hardware platform is hlw Compatibility testing.

Eg: In different processor like - Intel, AMD
In different mother Board like - Intel, Asus
In different RAM - (Random Access Memory)
In different ROM - (Read only memory)
In different Hard disk - (Sony, Seagate)
In different Graphics Card - nVidia, AMD.

3). Mobile Compatibility Testing:-

When we test an application for different operating system and also for different version like Without, windows, ios and for each brand of mobile

Eg: Samsung, redmi, nokia etc.

For each brand
For different models like A series, M series, Galaxy for SAMSUNG.
Xperia C4, C5, Z5 for Sony.
V15, V1, V19 for VIVO.

4) Browser Compatibility Testing:-

Testing an application in different browsers

Eg:
Chrome
Mozilla firefox
Internet Explorer
Opera
Uc
Netscape Navigator
Duckduckgo
Safari (It is for Apple Computer & for IPHONES).

What kind of bugs / defects we find in Compatibility testing?

1. Look and feel changes (ex: font & colour changes)
2. Object overlapping (Login & 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 / problem
6. Scattered Content (clumsiness)
7. Certain buttons, links and components may work in one browser and may not work in another browser.

Globalization Testing:-

Testing an application which is developed for multiple languages 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 the word displayed. This may happen because of below reason.

1. Machine does not have feeling.
2. It Cannot understand Exact meaning
3. It Cannot understand Exact Grammer

4). It cannot understand Exact Spelling.

All these are reason for wrong translation. So we go for human translator.

If a person is very good with multiple languages local and International, we could perform good globalization testing.

Japanese, Chinese, german, French, Korean Spanish languages are in high demand.

Types of Globalization Testing:-

1. I18N - Internationalization
2. L10N - Localization

Internationalization Testing:-

Testing the application whether it displays the right content at the right place in the right language is called I18N testing.

Eg:

When we are testing an application for English, Hindi, Kannada, Tamil, Telugu etc, sentences should display from left to right (i.e left aligned)

Whereas for Arabic, Urdu, sentences should display from right to left (i.e Right aligned).

Localization Testing:-

Testing the application with respect to the local culture or local standard, to the country or state or region is localization testing.

Eg:

For India	For US
Rs	$
dd/mm/yyyy	mm/dd/yyyy
Pin Code (560097)	Zip Code (CA 12345), (NY 25684), (SJ 58746)

Useability Testing:-

Testing the user friendliness of an application is usability Testing (or)

checking whether application gives to enduser what he needs within the actions is Usability Testing.

Ex:1 Touch screen mobile phones are more user friendly than keypad mobile phones.

Ex:2 In mobile, face lock is more user friendly than, fingerprint

Finger print is more user friendly than pattern

Pattern is more user friendly than pin

pin is more user friendly than password

Ex:3 VPS is more user friendly than GPS

VPS - Virtual positioning System

GPS - Global positioning system.

Ex:4 The application which accepts the current location by itself is more user friendly than where application user should enter current location.

If an application takes current location automatically i.e more user friendly than the application where we should enter the current location

Ex: Zomato, Swiggy, ola, Uber etc.

Who should do usability Testing?

End User is b

Customer / client

Other project team member

If no one is available, last option is test Engineer.

Note:

PT, IT, & ST is very important to start the business

Compatibility Testing; performance, ~~optimization~~ globalization, Usability testing is very important to expand the business.

Reliability Testing:-

Testing the functionality of an application continuously for long duration of time is reliability Testing.

If I test any app continuously for 1 month and 2 month is reliability.

Aesthetic Testing:- (look wise), (front end)

Testing the beauty of an application is aesthetic testing i.e here, we check the colors design, alignment, fontsize, font style of the application.

Accessibility Testing / Adhoc Testing / 508 ACT Testing.

ADA - American Disability ACT.

Testing the Application which is developed for physically challenged people is accessibility Testing.

This testing is mainly done in developed country. Ex If an application has RGB colour, it is a bug as a part of accessibility testing. Because colour Blind people cannot see RGB Colours in an application.

Tools used for Accessibility Testing:-

Infocus
Wave. etc,

Exploratory Testing:-

Testing an application by Exploring it, when there is no requirement.

We do this testing based on common sense, then we write scenarios and we convert to Test Cases.

What is the draw backs when there is no requirements:

There will be misunderstanding blw developer & Testing(dev team).

Some times defect is misunderstood as a feature and feature is defect understood wrongly

Some times defect is misunderstood as a feature and feature is misunderstood as a defect.

How to over come the draw back.

~~Misunderstanding blw dev & TE (No req).~~

By communicating with each other professionally (politely).

Communicate with developers, BA, Customer etc.

Adhoc Testing :- / Monkey Testing / Gorilla Testing.

Testing the application randomly without following any requirement documents. is adhoc testing. Eg: req doc, Scenario doc, T.C doc

Why we do adhoc Testing?

We may get more defects when we don't have much defects by doing Smoke, FT, IT, ST.

This testing help us to Concentrating more on negative scenario

This will help us to cover even invalid datas on the application

Note:

While doing Adhoc testing, our brain should think like below people.

Monkey
Children / kids
Happy / Excited people
Depressed people
Drunk People
Uneducated

by thinking from their angle, we can get more -ve scenarios & find bugs.

Note:

Smoke Testing, Sanity Testing, Adhoc Testing, Exploratory Testing all are situation based Testing

23/4/23

Security Testing:

Testing the data and Resources of an application is accessible only for authorized users and it is highly secure

Eg:

1) Pass word should be in encrypted format
2) Any application with locks (https) is more secure than unlocks (http)

   http - hyper text transfer protocol
   https - hyper text transfer protocol secured.

3) Otp, Captcha, login are more secured than other login.
4) Virtual keyboard are more secured than physical keyboard. Eg: Citi bank.
5) Encrypted data are more secured than decrypted data (otp for password, Atm pin, Verification code
   In some cases Acc Num, Aadhar num, etc).
6) session timeout option is more secure for banking application.
7) Incognito window is more secured than normal browser window.

Types of Security Testing:-

1. Vulnerability Scanning
2. Security Scanning
3. Penetration Testing
4. Risk Assesment.
5. Security Audit
6. Ethical Hacking
7. Posture Assesment.

1. Vulnerability Scanning:-

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

2. Security Scanning:-

Once the 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 risk 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 System 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 measure 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.

Fuzz Testing / fuzzing:-

Testing the Application using Invalid data also checking the vulnerability of an application using automation Testing.

Checking with invalid data is nothing but Adhoc testing.

Checking the vulnerability is nothing but Security Testing.

Combining both and testing using a tool is fuzzing / fuzz Testing.

STLC:

It is a procedure to test the software Application

It has different stages / phases.

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 lead or test manager. Because the plans will be done by experienced people.

Write test case:-

It is a step by step procedure to perform the testing on the application. It is done by the test Engineer. Once we go through the requirement we identify the scenario and then converted into the test case. To write the test case, we need requirement and test case template or tools (qc/alm, JIRA etc).

Prepare RTM:-

Requirement Traceability Matrix.

It is a document which is prepared to check whether every requirement has atleast one test case or not.

To prepare RTM we need both requirement and test case.

Execute test case:-

Once the requirement is given to the test Engineer, he writes test case 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 cases and software Application. This is where exactly the software is been tested, This is most important phase of software testing life cycle. ●

Defect tracking:-

While Executing test case, we will find defect that defect will be released to developer. Test Engineer should track what is happening to that defect. This Process is Defect Tracking.

Test Execution Report:-

This report will tell about total no. of Test cases executed. It will also tell how many test cases are pass and how many test case are fail.

This will be prepared b/w the project also d end of project.

This can be prepared by Test Engineer and also by Test manager / Test lead.

Till this stage customer can involve & he can ask all documents. This is the last stage from customer angel.

Retrospective Meeting:- / Post mortem meeting:-

Once everything is completed, all team members will gather together like, Test Engineer, developer, manager, BA. They all will discuss about good things in the project and the improvement.

This will be considered for future purpose.

DLC / BLC:-

Defect life cycle / Bug life cycle / Status of defects / Bugs.

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, Now the status will be New / open. That defect will be assigned to the concerned developer or development lead (if we don't know the correct developer). Then the status is Assigned.

Then the developer will reproduce the defect. he starts fixing the and if it is Reproducable, it will be accepted. Then, defect in development server and it has to be installed to testing server. Then he should change the status to fixed.

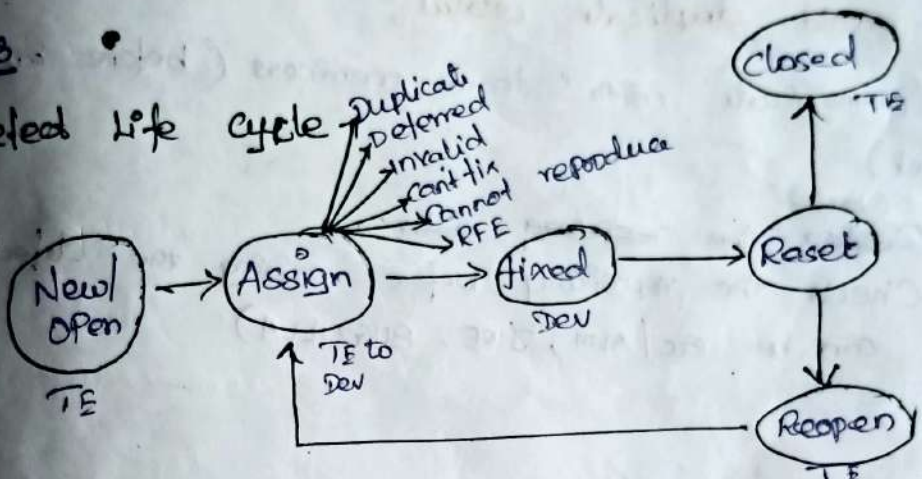
The test Engineer starts restesting the defect in the testing server. If the defect is properly fixed, then the status will be closed. If the defect

is not properly fixed., then again, the defect is Reopen to the developer and assigned back. This process continuous till defect goes to closed status

This Defect life cycle / Bug life cycle.

Status.

Defect life cycle



Apart from Above status, we have other status as well they are

Duplicate
Deferred → less time to fix
Invalid → Not accepting as defect
Can't fix → unable to fix defect
Cannot Reproduce → unable to see defect
RFE. → Not a defect but good suggestion

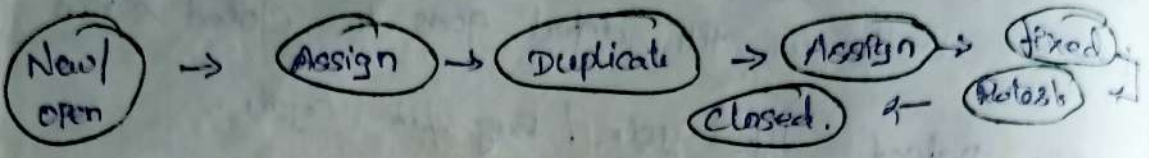
What is Duplicate status ?

Whenever same defect is raised for multiple times, the second defect will be changed as duplicate status by developer.

If it is test Engineer mistake below life cycle will be there.

New/open → assign → duplicate → closed
TE TE to Dev Dev TE

If it is developer mistake below status will happen



How to avoid duplicate status?

Communicate with team members (before raising the defect)

~~Check~~
~~Testing~~ the ~~repository~~ ~~after~~ ,
Check the repository before raising the defect (repository can be QC/ALM, JIRE, BUGZILLA)

Note:

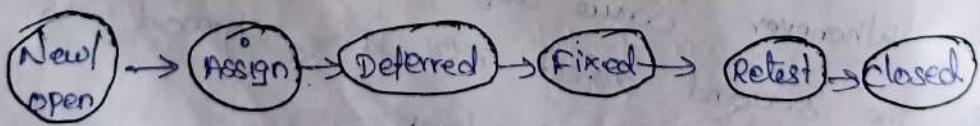
Duplicate is bad status for test Engineers.

What is Defferred status?

Whenever the test Engineer raises the defect, developer has less time to fix all defect.

In this case, developer will postpone the fixing of minor defect. Then he will change the status of that defect to deffered.

Journey of deferred status:



Note:-

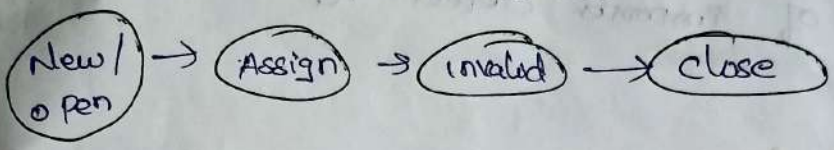
Deferred is good for test Engineers.

What is Invalid or rejected status?

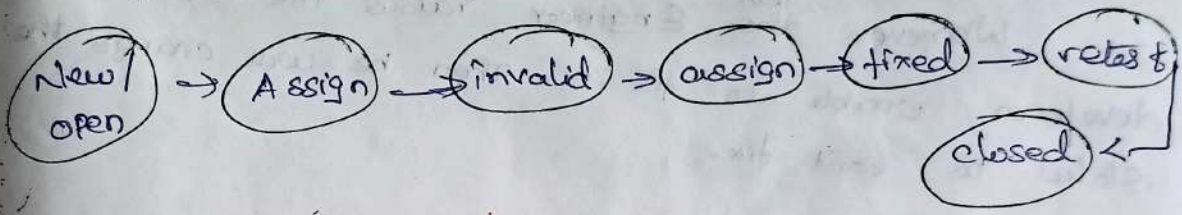
Whenever test Enginoor raises the defects to developer. if developer not accepting as a defect then hese will change the status to Invalid.

Eg of invalid:-

If the developer is correct



If the test Enginoor is correct

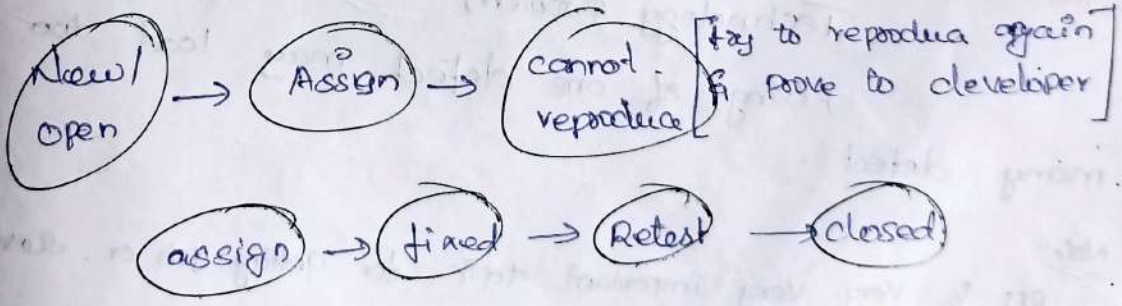


What is Cannot reproduce?

Whenever test Enginoor raises the defect and they also have a proof, but developer is unable to see the defect.

In this case developer change the status to cannot reproduce.

Eg of Cannot reproduce:-



Eg: Reason for Cannot reproduce :-

Because of Inconsisten bug (mobile phone sometimes hang and sometimes not, ?.1 is inconsistence bug).

Due to Installation problem

Because of improper defect report.

Calculas

What is Can't fix?

Wherever test Engineer raises the defect but developer unable to fix it. Then he will change the status to can't fix.

Example of status flow:-

New/ Open → Assign → can't fix → closed

(if it is a valid reason this will be informed to manager or BA/ customer.

Once it is finalized it will be closed)

What are the reason for Can't fix.

Technology problem

Fixing of one defect may lead too many defect.

Note:

RFE is very very important topic for getting job or survival

Analysing the status is very important

We can use tools like QC/ALM, JIRA, BUGZILLA, trieflink etc to report the bugs

Status of defect will be used in defect report.

What is RFE? (Request for Enhancement).

Whenever test Enginer raises the defect developer does not accept it as a defect but ~~output~~ accept it as a good suggestion.

In this case he will change the status to RFE. New/Open → Assign → RFE → Close

Severity and Priority of a defect:-

Severity:-

Severity says how much that defect is impacting Customer business. Used in defect report to raise defect.

we have different types of severity;

1. Blocker / show stopper
2. Critical
3. Major
4. minor
5. Trivial

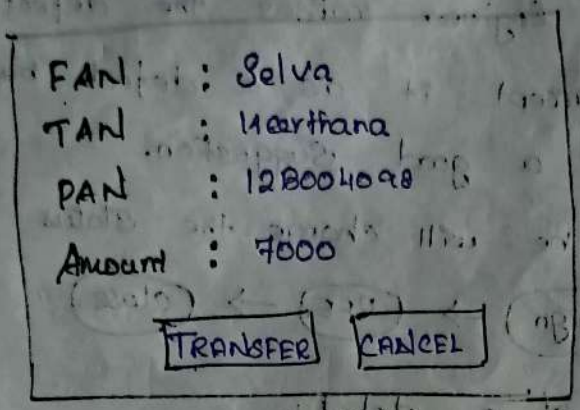
Priority:-

Priority says which defect has to be fixed first or it will tell the importance of fixing the defect.

Types of priority:-

High		Urgent	P1
Medium	(or)	High	(or) P2
Low		Medium	P3
		Low	P4.
			P5

Amount Transfer



Example for Blocker Severity and Urgent priority Bug:-

Transfer Button is not working or functioning in Banking Application

Example for Critical Severity and high priority Bug:-

Transaction History page is not opening

Example for Major Severity and Medium priority Bug:-

Cancel Button is not working.

Example for minor Severity and Low priority Bug:-

In the transfer success message, instead of "successfully" if it is displayed as "sucessfully".

Example of trivial Severity and Low priority Bug:-

Negligible colour change, font change, alignment change comes under this category.

Severity usually will be the same

Priority keeps on changing.

Both Priority & Severity should be set by Test Engg by analysing the defect.

1/10/23

Mini Project :-

Project Name :- OLA

APP URL / Download :- Client server App downloaded in playstore

Testing Environment / Platform :- Production server, Android 13, mobile

Domain Name :- Travel

Module Tested :- Home page, source & destination page, cab selection page, payment page.

Types of Testing :- Functional Testing, Integration, system testing, Exploratory testing, Performance testing (as a team), Compatibility testing (as a team), Smoke testing.

What is tested ?

Scenarios Tested :-

Verify by trying to book a cab. "where to where", "local-to-local" → Vadapalani - Arumbakkam.

Verify by trying to book a cab from city to city, city to city (other state) ... !

Eg: Chennai - Theni, Chennai - Thirupathi.

Verify by booking a cab from South India to North India. (Tamilnadu - Delhi). → city to city (Nagar to Kashmir)

city to island (Chennai - Maldevies.

city to city (country to country) → Bombay - Taiwan.

Defect report:-

It is a report which is prepared by Test Engineer for every defect / Bug.

It should be clearly understandable to developers.

Defect report contains different attributes

Eg: of defect report:

Defect Id: 001

Build No: B13

Test case number:- (TC_____)

Status : Assigned

Severity : major.

Priority : High

Testing Environment : Chrome browser, windows os

Module Name:- order online (spaloon meals)

Reported by : Keerthana.T

Assign to : Concerned developer.

Brief description:- B veg shake is displayed in red color instead of Green

Test DATA: NA

Steps to reproduce / Description

1) Open the Browser (chrome)
2) Enter URL (www.zomato.com)
3) Choose location as Hyderabad
4) click on search for Restaurant field in Home page.

f) Type Supertan Meals in Search bar
g) Select Supertan meals (Banjara Hills)
h) Select order online option
i) Click on shake option
j) Scroll down till you get Beetroot juice.

Expected Result:- Beetroot juice should be displayed in Green colour.

Actual Result:- Beetroot juice is displayed in Red colour

Attachment:- (Using snipping tool.

6/12/23

Difference b/w Defect, Bug, failure, Issue and Error.

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

Bug:-
It is a informal name of Defect
Whenever test Engineer finds a defect, report it to developer and if developer accepts the defect it is termed as bug.

Error:-
It is a mistake in the source code done by developer

Issue:-
Any problem which is faced by customer is called as issue.

Failure:-

When multiple issues are faced by Customers or end user, it will lead to failure.

To avoid failure, we should avoid issues. To avoid issues test Engineer should find defects.

Tools used in IT Industry:-

Functional Testing Tool:-

1) Selinium:- (To automate web app!) (demand)
(Thought Works)

→ It is a tool which is founded by Thought works. It is an open source tool (It is free of cost).

→ Selinium can be used with language like Java, Python, Pearl etc.

→ Currently, Java 'selenium' combination is in high demand.

2) QTP, UFT:- (Powerfull tool)

QTP → Quick Test Professional

UFT → Unified functional Test

It is a licensed tool i.e it is Paid tool. Currently, this licensed is with HP. Before this it was with Mercury.
↓
Hewletts Packards

To use QTP/UFT, we can use language like VBS (Java Visual Basic Scripting, Js (Java Script), etc.,

UFT is a latest version of QTP
Currently this tool is used very less in IT industry.

3) Silk Testing :- Web App → Selinium, QTP / UFT
4) Test Partner Mobile App → Appium
5) Win Runner → Postman
6). etc., API
↳ App Programming Interface

Performance Testing Tool :-

(+) { Load Runner (Licensed tool it is with HP)
J meter
Silk Performer.
QA Load
Neo Load etc;

Test management Tools :-

(%) Qc / ALM → Application Life Cycle Management.
Quality Center

It is also a licensed tool. It is highly used tool currently. The latest version is ALM
The License is with HP.
We can do below activities using this tool Under different folders (modules).

1) Requirement
Add Requirement
Modify "
Delete "

2) Test Plan
write Test cases
Modify :
delete "

3) Test Lab :-

Execute test cases

4) Defect :-

Report / raise / Log the defect.

Track a defect

2) JIRA :- (Best tool for Agile model)

3) Test Link :-

4) OTM ⇒ Oracle Test manager

5) Fire flink. etc..

Defect management tools :-

Defect tracking tool :-

1. Bugzilla (open source) (free tool)
2. Qc / ALM
3. JIRA (Partially ~~Partially~~ license, Different tool for Agile model.
4. Test Link
5. OTM
6. Fireflink.

Note :-

In Bugzilla, we cannot do below activities :-

Add Requirement

Modify Requirement

Delete

Write Test case

Modify test case

Delete Test case

Execute Test case

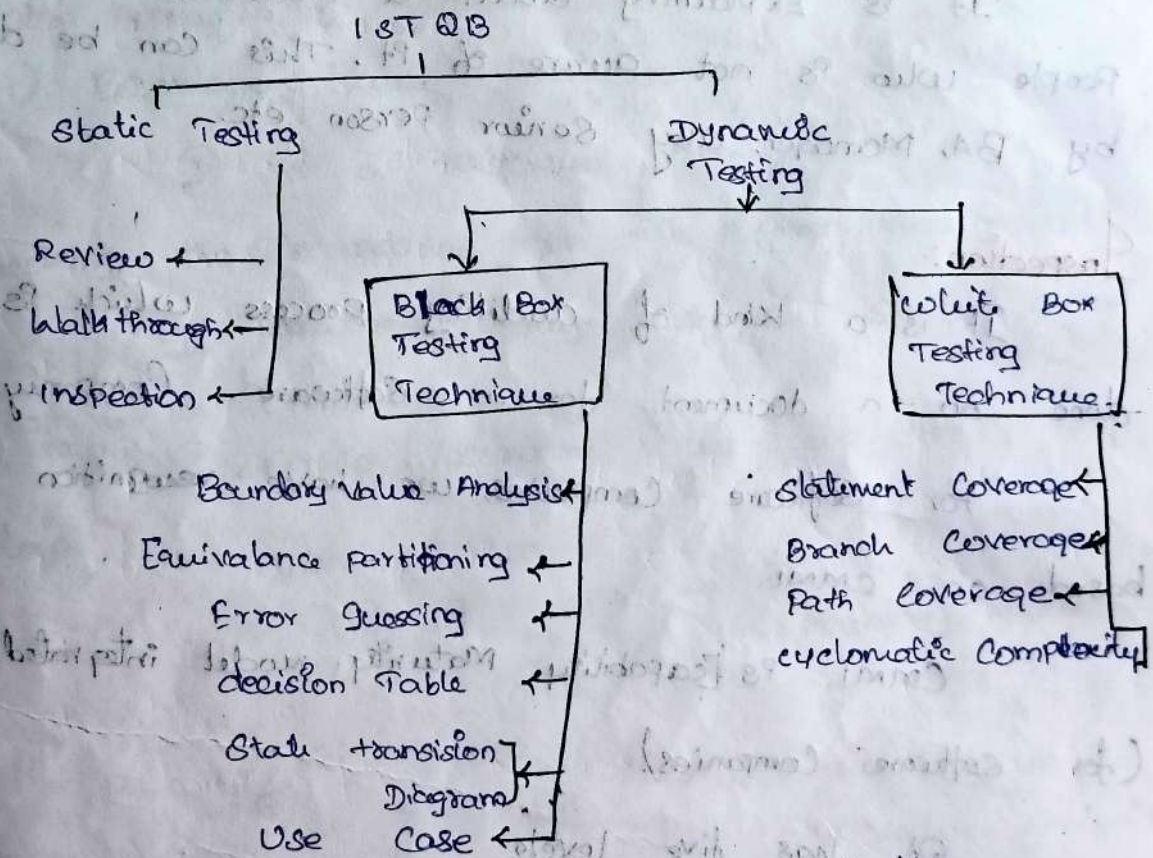
In Bugzilla we can do below activities.

1) Report
   Raise
   log.
2) Track the defect.

8/12/23

1ST QB:-

International Software Testing Qualifications Board.



It is a certificat for Software testing Engineers.

Static Testing:

Review:-

It is finding a mistakes in the document.

We have different types of Review

1. Self review (checking our own documents)
2. Peer review (checking others document who is of same level)
3. Manager review (Manager checking the documents).

Walk through:-

It is Explaining about a document to the people who is not aware of it. This can be done by BA, Manager, any Senior person etc.

Inspection:-

It is a kind of auditing process which is done on a document for a Software Company

For Software Companies, we have recognition based on CMMI

CMMI is [Capability Maturity model integrated] (for Software Companies)

It has five levels

CMMI level 1
CMMI level 2
CMMI level 3
CMMI level 4
CMMI level 5

Highest level is CMMI level 5

CMMI is the recognition given by SEI

SEI is Software Engineering institute.

9/12/23

Dynamic Testing:-

Testing which is done on Application is Dynamic Testing. (Eg: FT, IT, ST etc).

Black Box Testing Techniques

White Box Testing Techniques.

Black Box Testing Techniques:-

1) Boundary Value Analysis:-

BA is a technique where we are checking the boundaries for the given requirement

Eg: Requirement say username should accept blw 6-10 characters for this Boundary Value analysis is like below

6 (+) 10 (+)

5 (-) 7 (+) 9 (+) 11 (-) → 3 Value Boundary Value Analysis

6 (+) 10 (+)

5 (-) 11 (-) → 2 Value Boundary Value Analysis

Eg 2: Requirement says, Phone number should accept 10 digits exactly.

10 (+)

9 (-) 11 (-)

Note:

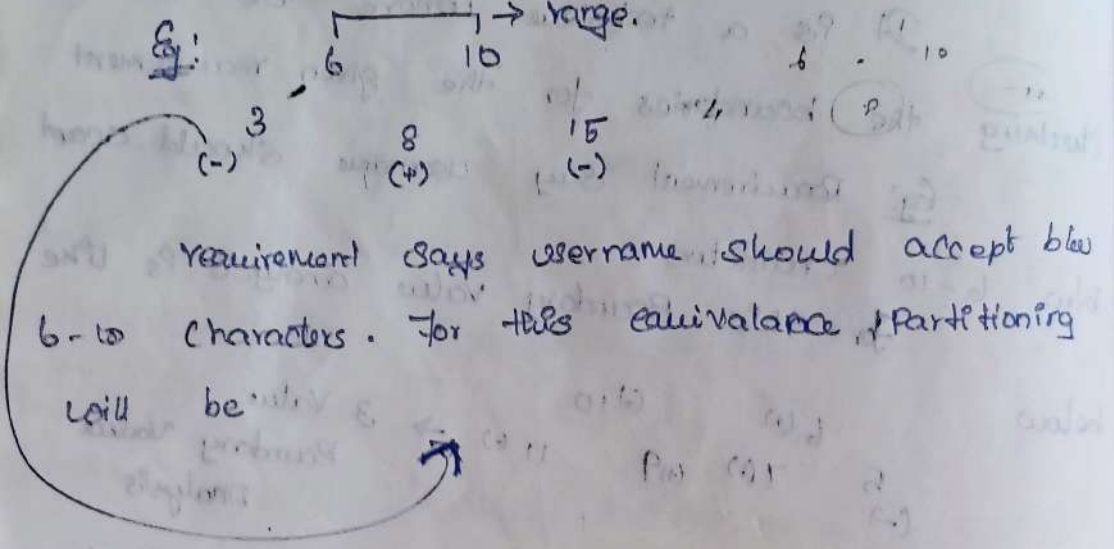
* BVA can be applied for FT
* We can apply this technique for Text field etc. (username, Password, Email Id, phone number) based on requirement.

less Scenario, more Coverage is the Advantage

2) Equivalance Partitioning :- / Equivalance Class

It is a technique which will help us to come up with one positive, & negative scenario for a requirement which is in range.

Eg:



requirement says username should accept b/w 6-10 characters. for this equivalance Partitioning will be

Note:

1) We can apply equivalance Partitioning for requirement which is in range.
2) It can be applied for FT in Text field & Text Area

It can be

3) Error Guessing :-

It is a technique where we come up with scenarios with different types of I/p data.

Eg: Requirement says username should accept only alphabets. minimum 5 to 10 characters.

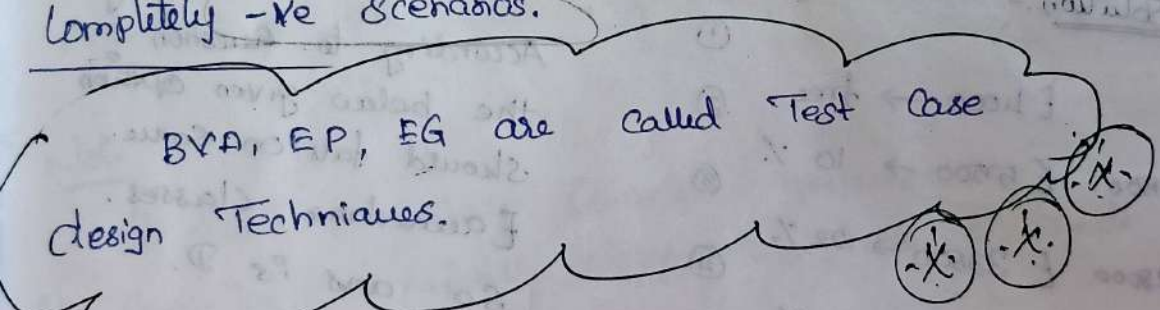
For this Error Guessing scenarios are

1) HARSH@ → S.C
2) HARSHA → Num
3) H@R$#@ → Combo of all
4) Empty Space

Note :-

This is used for F.T and it is completely -ve scenarios.

BVA, EP, EG are called Test case design Techniques.



6 - 10 ✓ ✓

6 - 10 (-) S.C

3 8

11/12/23

Equivalent partitioning.

ISTQB A.P 1

Path

Open the Browser

Enter the URL → tryqa.com

Click on ISTQB. Dumps - mock test

Scroll down

Click on ISTQB foundation level Part-1

Q.P.1

Q.No: 5

Refer Question from above link on path.

Solution:-

Partitioning.

E 4000 → free ①

+1500 E 5500 → 10 % ②

+28000 E 33500 → 22 % ③

>33500 → 40 % ④

According to Question the below given option should fall on same Equivalance classes.

So ans is D.

d) E 5800; E 28000; E 32000

All the 3 value comes under same Class that is 3rd class as answer.

Decision Table:-

It is a technique which helps to analyse the requirement in easy way. in the table format.

Eg:- ISTQB Q.No - 10
Q.P.No: 1

Refer Q.No:10 from Q.P.No:1

Solution:

There are 4 People. 2 belongs to citiBank other 2 belong to Non citi Bank.

A belong to Rule 1
B belong to Rule 4.

According to the table [d] is the answer

Note:-

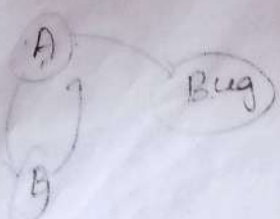
If there are any Complex requirements, it can be explained using this table in simple way.

State Transition diagram:-

12/12/23

It is a technique which helps us to understand the requirement easily.

Eg: ISTQB Q.P.No: 1 Q.No: 14

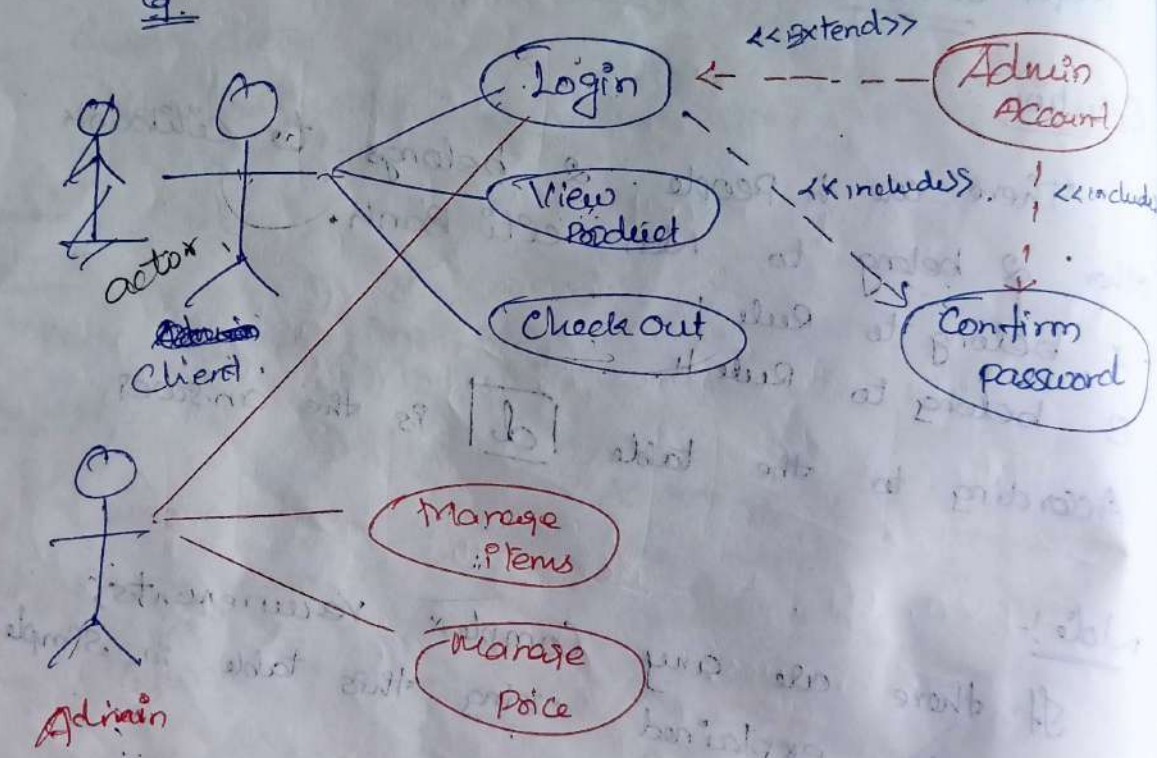


Use Cases:-

It is a technique which is explained in the form of actors and action.

Each actor can perform one action or multiple actions. This will help us to understand the requirement easily.

Eg:



White Box Testing techniques

Statement Coverage:-

It is a technique which covers all the statement in a program.

Branch Coverage

It is a technique which covers all the branches in a program.

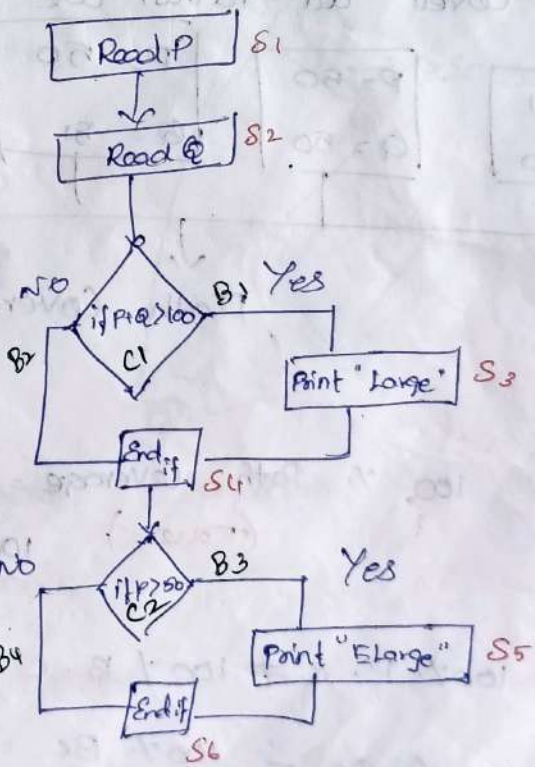
Path Coverage:-

It is a technique which covers all the paths in a program.

Q.P.NO: 1

```
Read P
Read Q
if P+Q>100
Print "Large"
End if
if P>50
Print "PLarge"
End if
```

Q.NO: 27:



In the above flow chart we have 6 statements, 4 branches and 2 conditions, 4 Paths

To cover all statements, we need one Test case,

Test Case

P = 51
Q = 50

→ statement Coverage = 1

To cover all branches, we need 2 test cases

P = 51
Q = 50

P = 50
Q = 50

↓

Branch Coverage = 2

To covell all Paths, we need 4 test cases

P = 51
Q = 50

P = 50
Q = 50

P = 150
Q = 51

P = 51
Q = 49

↓

Path Coverage = 4

Note:-

if 100 % Path Coverage = 100 % Branch Coverage = 100 % statement Coverage (True)

If 100% P.C ≠ 100% B.C ≠ 100% S.C (False)

If 100% SC = 100% BC = 100 % SC (False)

Cyclomatic Complexity :-

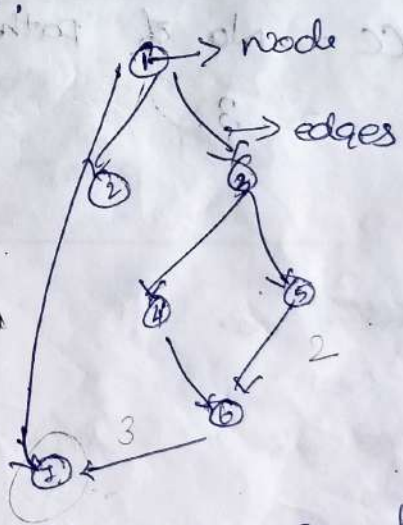
It is a technique which will help us to find Complexity of a source code.

Eg: If Complexity is 1 to 5 it is low Complexity. If it is 6 to 10 it is medium Complexity. If it is more than 10, ... it is high Complexity.

This technique helps to reduce the Complexity of source code.

It is source code is Complex & Testing become complex.

Eg:



In the above diagram we have 8 edges & 7 Nodes

~~Formula: To calculate~~

Formula to calculate Cyclomatic Complexity :-

(1) $CC = (E - N) + 2$

$= (8 - 7) + 2$

$= 1 + 2$

$CC = 3$

② CC = (No of regions) + 1

= 2 + 1

= 3

③ CC = No of Predicate Nodes + 1

= 2 + 1

= 3.

(It is a node which has 2/ more edges)

In node which has 2/ more edges is Predicate Node:

Here 1 & 3 are Predicate Node.

④ CC = No of paths

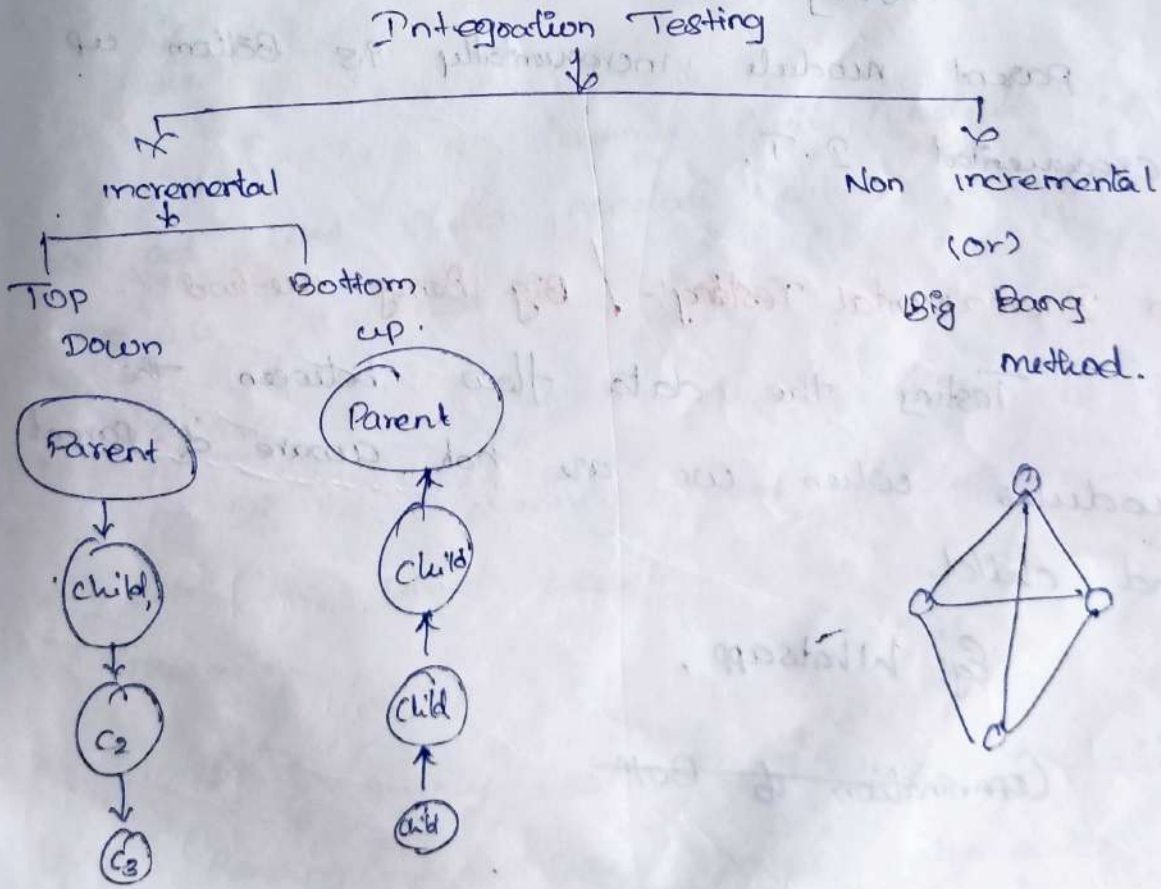
= 3

11/10/23

Differena B/w QA & QC

QA	Qc
Quality Assurana	Quality Control
QA is also called as Verification	Qc is also called as Validation.
QA is process oriented	Qc is product Oriented.
It is a preventive method	It is Lotte Corrective method.

Types of Integration Testing:-



Incremental Testing:-

Testing the data flow from one module to another module incrementally.

Two types

Top Down

Bottom UP

Top Down:-

Testing the data flow from parent module to child module incrementally is Top Down incremental I.T.

Bottom up:-

Testing the data flow from child to parent module incrementally is Bottom up incremental I.T.

Non Incremental Testing:- / Big Bang method

Testing the data flow between the modules when we are not aware of parent and child.

Eg: Whatsapp,

~~Combination of Both~~

Sandwich

Testing the application by applying incremental and non incremental is Sandwich Testing.

Eg: for incremental → ATM, Chalo

Eg for non incremental → Whatsapp, Telegram

Eg for Sandwich → Amazon, Facebook, Instagram

Stubs & drivers:-

When one module is there, other module is not there but we want to do integration testing then we go for stubs & drivers.

Stub

It is a dummy child module which will act like child module when parent module is present

Drivers:

It is a dummy parent module which will acts like parent module when child module is present.

It is mainly used for Complex applications like Banking, health care, Airways, Submarine. Navy, military, Space

Note:

Both Stubs & drivers are only used for testing purpose

Stubs & drivers will not be given to Customers & end users.

15/12/23

D/B Functional & Non functional Testing:-

Functional	Non Functional
1) FT, ST, IT Comes under functional Testing (Functional, System, Integration Testing).	PT, CT, ST, GT, AT, RT, UT, Comes under ~~fun~~ non functional Testing (Performance, Compatibility, Security, Globalization, Aesthetic Reliability & useability)
2) This is to start the business	This is to Expand the Business
3) This will be done first	It will be done after functional Testing.
4). This is mandatory for every Application	This is based on additional requirement.

Types of RTM: (Requirement Traceability Matrix).

Forward Traceability Matrix

Backward " "

Bidirectional " "

Forward Traceability matrix:-

Maping from Base document to derived document

Eg: Maping from Requirement to test case to defect report

Backward Traceability Matrix:-

Maping from derived document to Base document.

Eg: Maping from defect report to test case to Requirement.

Bidirectional Traceability Matrix:-

Maping from derived to Base & Base to derived document (or)

Maping from derived to base & derived to another derived.

Base document -> Requirement

derived document -> Test Case, defect report.

Types of Regression Testing:-

1. Unit Regression Testing
2. Regional Regression Testing
3. full Regression Testing.

Unit Regression Testing:-

In unit regression Testing, we Check the impact areas with in the same unit / model

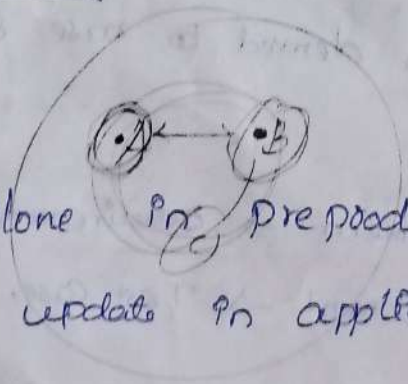
Regional Regression Testing:-

Here we check the impact areas blw multiple modules, i.e called Regional Regression Testing.

Full Regression Testing:-

Here we Check the impact areas for Complete application that is full regression Testing.

Patch:



DI is done in preproduction server. where some update in application can be done.

Build!-

It is a Compile and Compress format of Source Code. It is also different Version of an application.

Build will be ready once developer make any changes in the application.

What does build Contains?

A Build Contains New feature, old feature and bug fixes.

What are the Version of build?

There are 2 Versions as Compress & Archieve.

Compress	Archieve
Format: Zip files Eg xyz.zip	xyz.Jar Java archive xyz.war web .. xyz.tar tape ..
	Jar -> Back End | tape - Both front & back end war -> Front End
Multiple files will be converted to single file	multiple file will be converted to single file.
Size of the file will be ↓	Size of the file will be almost same.

Test Cycle:

What is Test Cycle?

It is a duration / time take to test an application once build is given. For every build there will be one test cycle.

Eg: If we have 30 build, we have 30 test cycle.

What is release?

Release / Go-live / Production Release

Starting from requirement Gathering, developing an application. Testing an application and Releasing it To end users that is for production server is called one release.

Release is also called as GO-LIVE / PRODUCTION RELEASE.

Acceptance Testing / User Acceptance Testing :-

Testing the business scenarios of an application which is done by customer is acceptance Testing.

Why Acceptance Testing will be done?

To get a confidence Before Application is released to production server.

By mistake, s/w company might be given wrong Application to customer.

To avoid this, Acceptance testing will be done by customer.

When to do Acceptance Testing?

After white Box testing and Black Box testing completed throughly and Before production release of an application.

In blw this, Acceptance testing will be done.

Is Acceptance testing mandatory or optional.

It is depends on customer Decision.

Types of acceptance Testing:-

1. Alpha testing.
2. Beta testing.

Alpha	Beta
It is done at organization / blw company (Development place)	It is done at Customer place.
It is done at Controlled Environment	It is done at Controlled Environment.
It is done at virtual Environment	It is done at real Environment.
In alpha both WBT, BBT	Here, Black Box Testing will be done.
It is closed for public	It is open for public.
Alpha is offshore	Beta is done at onsite

1) What are the different approaches of acceptance testing?

1) customers can do acceptance testing in company. (Alpha)

2) customers can do acceptance testing in his place (Beta).

3) BA can do acceptance testing and company (Alpha)

4) Employees of customers can do acceptance testing at customers place (Beta)

5) Test engineers of slw company can do acceptance testing (Alpha)

6)

Hot fix / Incident Management System:

Once Application is realeased to production Server, if end user / customer finds any bugs: It will be reported to Slw Company (P.M). If will be fixed are resort immediatly. and released back to production Server. This whole process is called HOT FIX:-

This will be done without seeing any time (Day / Night time) based on priority.

Incident (Bug)	Priority	Hours given to fix
1	High	10
2	medium	5
3	Urgent	1
4	Low	3
5	Medium	1

Incidents will be Created by customer

Fish Bone Diagram / Ishikawa diagram

Once there is a production issue, it will be resolved. Later all team member will gather together and discuss about Root cause of production issue

Each reason will be document in the form of fish bone diagram.

Each team member ~~with~~ should give reason for production issue.

Note:

Fish Bone diagram will be ~~use~~ done only if there is how fix.

Short term release.

Whenever between 2 planned releases, if there is any ~~unplanned~~ sudden unplanned release, then it is called short term release.

Dtes will be decided by customer.

D/B Test strategy and Test plan.

Test strategy is the approach to do the testing.

We can have multiple approaches while doing testing.

From multiple approaches we will choose one approach and make plan for PT to execute.

Eg: we can travel to home town in different mode of transport (bus, car, train)

Out of these, if we choose bus, we should plan about how to book the bus

Note:-

Always approach comes first, then plan will be done.

20/12/23

Defect Seeding :-

Whenever developer intentionally injects a bugs, without informing to test Engineer, that is defect seeding.

To do this, developer will take approval from manager. It is a process to test the efficiency of Test Engineer. (It is done secretly)

Defect Leakage :-

Whenever Customer / end user finds any bug which is not found by test engineer, that is defect Leakage / Bug leakage.

For a very Good testing team, bug leakage should be 0%.

Defect Masking

Whenever one defect hiding another defect is defect masking.

Masking defect will be hiding masked defect.

Defect Density:

It is a calculation of no. of defect w.r.t to line of code and duration

Lines of code → L.O.C

Thousand Lines of code → K.L.O.C

Eg: $DD = \frac{\text{No. of. Valid defect}}{\text{No. of. LOC.}} \times 100.$

$DD = \frac{\text{No. of. valid defect}}{\text{No of. days (duration).}}$

21/11/23

Test Bed:-

It is nothing but a layer / environment To do testing.

Eg: chrome browser / fire fox browser can be called as test bed.

Android / windows can be called as test bed.

Test Methodology:- / methods of Testing:-

We have different methods of testing like Unit testing, functional testing, Integration Testing, System testing etc. At last we have acceptance Testing.

Recovery Testing!

Test the application by checking whether it is able to recover from crash state.

Eg: open multiple applications make it intentionally hang, restart the system or mobile. Again system / mobile should work fine with all the datas saved.

What is test Coverage? How will you ensure that everything is covered in Testing?

First I go through the requirement thoroughly. I will write Scenario & test cases I will apply test case design techniques like BVA, EP, EG

I will give my testcases for review purpose I prepare RTM. I also make sure statement Coverage, Branch Coverage & Path Coverage is done

By all this things I can say My testing is Covering all possible scenarios.

How will u prove you are a good test Engineer

Ans. Is Test Coverage & (D/B Developer & Test Engineer) Test Engineer from difference.

D/B Developer & Test Engineer :-

Developer	Test Engineer
1) They want to build an application	They want to break (finding Bugs) an application
2) They should be very good in logical thinking	They should be very good in creating (out of Box thinking)
3) They usually think in +ve angle	They usually think in both +ve & -ve angle.
4) They focus on construction of application	They focus on perfection of an application.
5) They should be very strong in programming	They should be very strong in finding bugs.
6) Developers do White Box Testing	Test Engineer should do Black Box Testing

Why testing is important?

Every application is developed for business purpose. If there is any defect, it will be -ve impact for end users. This will spread across multiple end users and no of enduser will be reduced. This becomes loss in customers business.

To avoid this, we should do testing before application goes to end user.

Eg: Am trasfering money to one person but it goes to other person. & business gets lost so we should do testing.

22/12/23

Principles of s/w Testing?

1) Exhaustive testing is not possible :-

Partially we cannot do exhaustive testing since it contains to many scenarios.

So we go for optimized testing (it is not under testing or it is not over testing).

2) Defective Clustering:-

There is a saying that 80% defects will be found in 20% module. This is from Pareto Principle.

Ex if there is a defect in login module, Surrounding login module will have more no. of defects This login area can be called as defect cluster.

3) Pesticide Paradox:-

Same kind of testing / scenarios will not help us in finding new defects

Always we need to update our testing type and scenarios to find new defects

Ex: If we want to get rid of mosquito, we try with different Solutions Like:-

All out / Good night / Hit / Mat / Bat

4) Testing shows presence of defects:

when we do testing, we can find defects

5) Absence of error - fallacy:-

It is possible that sleo which is 99% bug free is still unusable. This can be the case if the system is tested thoroughly for the wrong requirement.

This can be avoided by doing static testing.

6) Early Testing:-

Perform the testing from initial stage of SDLC (ex: review)

That is static testing. If we do this, we can avoid the wrong things which can happen.

7) Testing is context dependent:- (Need Based).

For different domain, oath testing should be in different way. Similar type of testing for all application is a bad practice.

we have to perform based on the type of application

Ex: If Banking, my thought is more on security

if gaming, my thought is more on Adhoc.

Test Plan:-

Test plan is a document that is prepared for future testing activities.

It is prepared by Test lead (or) test project manager / Test manager.

It contains different attributes / sections like

1) objective
2) Scope
3) Schedule and milestone
4) Entry & Exit criteria
5) Defect Tracking
6) Assumptions
7) Risks
8) Contigency plan / mitigation plan / Backup plan
9) Roles and Responsibilities
10) Environment / Platform
11) Deliverable
12) Graphs & matrix.

Objective:-

It tells about the aim/purpose of preparing the test plan.

Scope:-

This will say the limitation.

This section will tell about

a) Feature what to be tested (Gmail).

b) Feature what not to be tested.

(Instagram, whatsapp).

Schedule & milestone:-

It tells which activity to be done first and which activity has to be done next.

It is just like a time table of the project.

Ex:

Req (1 april) → write which test case (20 april) → Execute Test case (10 may) → UAT (30 may) → Go live (5 Jun)

UAT & Go live are mile stone.

Note: If we follow schedule, we can reach milestone.

Exit & Entry Criteria:-

This section will tell about when to enter and when to exit each type of Testing.

Entry criteria for FT

1) WBT should be done.

2) Build has to be installed for Testing server.

Exit Criteria for FT:

Pending defects should be lessor than or equal to

0- Blocker
1 - critical
5 - major
20 - minor.

Entry Criteria for IT:

0 - Blocker
1 - critical
5 - major
20 - minor.

5. **Defect tracking:-**

This section will tell about whenever a defect is found how to track the defect and which is the tool used to track the defect. Also it tells about, what all are the terminologies we are all using while raising the defect.

Eg: ALM tool —
P1 / P2 / P3 (Priority)

* Blocker, critical, major, Minor, Trivial (Security)
* Confirm, In progress, Resolved, Verify (status)

6. **Assumption:-**

This section will tell about what are the assumption we have during this.

Eg: All employees will be there in office daily

7. **Risk:-**

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

Eg: All employees are not in office, Leave / Quit.

8) Contingency / Backup / Mitigation plan:-

This section will tell about how to overcome the issue which occur during the process

Eg: Arrange regular knowledge transfer among the employees when everyone are there in office, before risk occurs.

9. Roles and Responsibilities:-

This section will contain what are the roles and responsibilities

1. Roles and responsibility of Manual test Engineer

Going through the requirement, understanding the requirement, identifying the scenario, write the test case. Prepare RTM, Execute Test case. when the build is given, then find the defect and raise the defect.

2. For automation Test Engineer:-

We should perform above test plan we should write test scripts and execute the test scripts by using the selenium tool, manage test scripts.

10 Environment / platform:-

This section contains which platform we are using for testing purpose

Eg: windows 10, 8, 7. Chrome Browser.

11. Deliverable:-

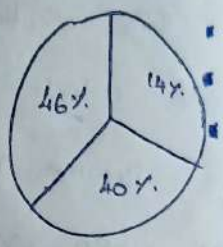
This section will tell what are the documents that should be prepared for the project.

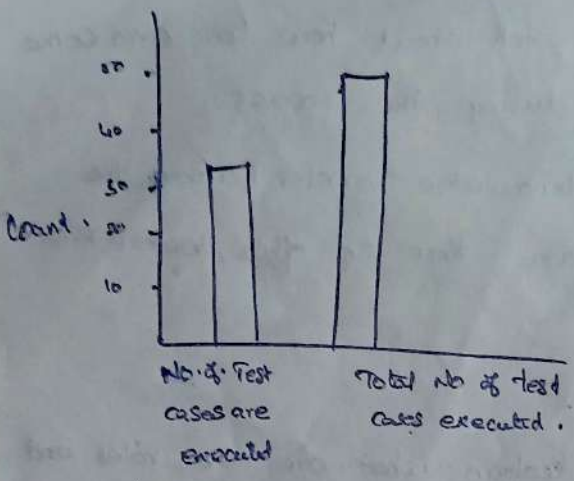
Eg: Test plan, Test case, RTM, Defect report etc.

12. Graphs & Matrices

This section Contain the Graph and Metrics that are prepared for projects.

Test Execution status :-





Snapshot of Manual Test Result :-

Test cycle	Total No of Test case	# of TC Executed	# of Test case Passed	# of TC failed
Cycle 1	88	88	88	0
Cycle 2	126	126	125	1
Cycle 3	174	174	173	1
Total Tcs	388	388	386	2

Entry Criteria for functional Testing :-

WBT has to be done

App has to be installed To testing Server

Smoke testing has to be completed

Exit Criteria for functional Testing :-

Blocker - 0
Critical - 2
Major - 5
Minor - 10
Trivial - 20

Entry for Integration Testing

Blocker - 0
Critical - 2
Major - 5
Minor - 10
Trivial - 20.

Exit criteria for Integration Testing:

Blocker - 0
critical - 0
major - 3
minor - 5
Trivial - 10

Entry criteria for System Testing:-

Blocker - 0
critical - 0
major - 3
minor - 5
Trivial - 10

Exit criteria for system testing:-

Blocker - 0
critical - 0
major - 0
minor - 1
Trivial - 5

Agile Model:

It is one of the peoperfull and the Best model in Industry.

In this model we have different types of Variants / Version

1. Scrum (80 to 90% of industry uses Scrum)
2. XP (Extreme programming)
3. RUP (Rational Unified Process)
4. RAD (Rapid Application development)
5. KANBAN. Process.

Scrum Terminologies:

1. Epic
2. Product Backlog
3. Sprint
4. User story
5. Sprint Planning meeting
6. Sprint Daily meeting
7. Sprint review meeting
8. Post mortem / Retrospective meeting.
9. Sprint Backlog:

Epic :-

It is a complete set of requirement given by the customer

It is also known as product Backlog.

User story / Story cards.

It is a part of requirement given by the customer

Eg: A-z Crequire ment

A (login) - user story,
B (compose) - user story
c (Inbox) - user story.

Sprint

It is a duration or time taken to work on the one or more user story.

It's is a time taken will depend on 2/3/4 weeks. based on the customer decision.

Sprint planning meeting:-

It is a meeting which has to be conducted before the sprint starts

In this meeting Product owner and scrum master will be involved. They will discuss / decide which user

story has to be worked in particular sprint.

They also decide the duration of the sprint

Product owner:

He is the person who has the customer / representative of the customer. He will decide which has the important user story.

Scrum master:

He is the person who has the representative of the software company. He can be project Manager / team lead / test lead. product is depends upon the company design.