

Manual Testing



SDLC

{Software Development Life cycle}

- Process to develop a SW.

- The types of models:

1) waterfall

2) spiral model

3) v model

4) prototype model

5) hybrid model

6) Deviated model

7) Agile Model.

1) waterfall model:

[Requirement Analysis]



[Feasibility Analysis]

Design → HLD

↓ → DLD

[Coding]



[Testing]

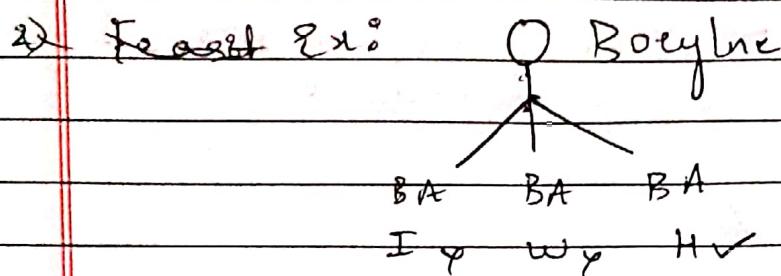


[Installation]



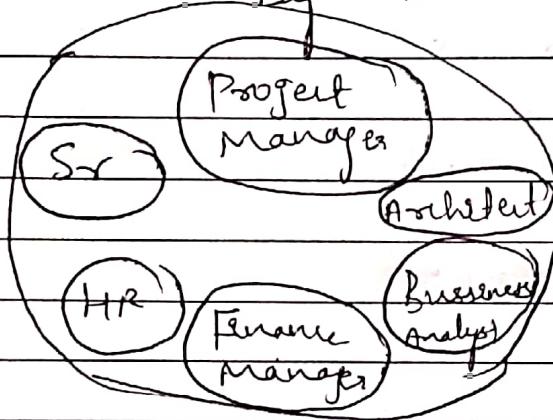
[Maintenance]

- Business Analyst will perform RA.



- 2) Feasibility Analysis:

It is done by ~~Architect~~ project manager

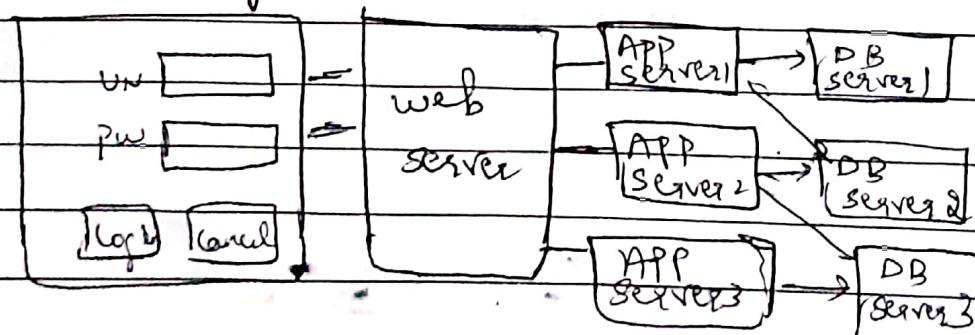


- Business Analyst will convert CRs (Customer Requirement specification) to Software requirement specification

- BA is also known as Subject matter Expert

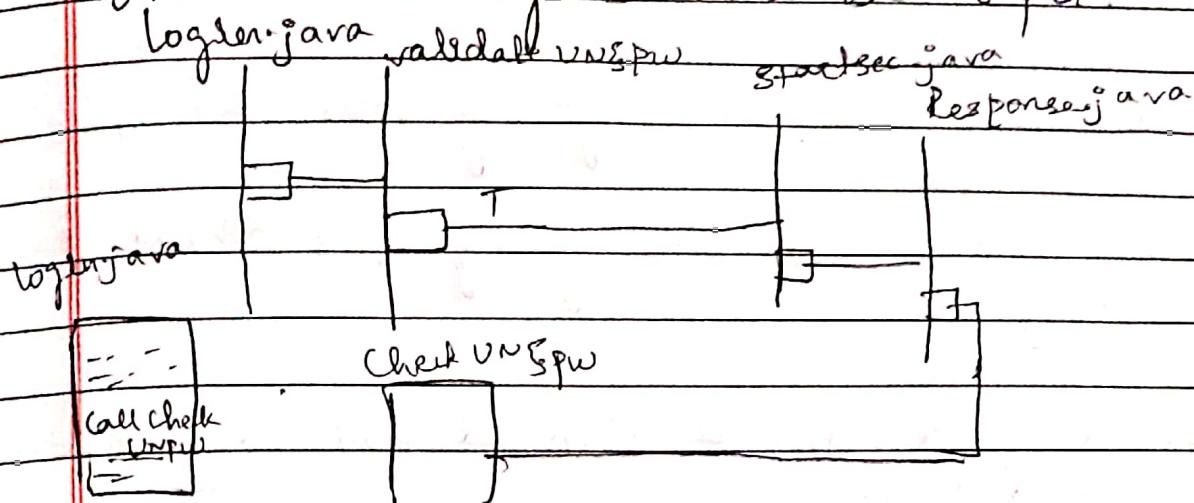
- After the Feasibility Analysis, the freezing of requirement is done.

- 3) Design: i) HLD (High level Design): It is done by Architect.

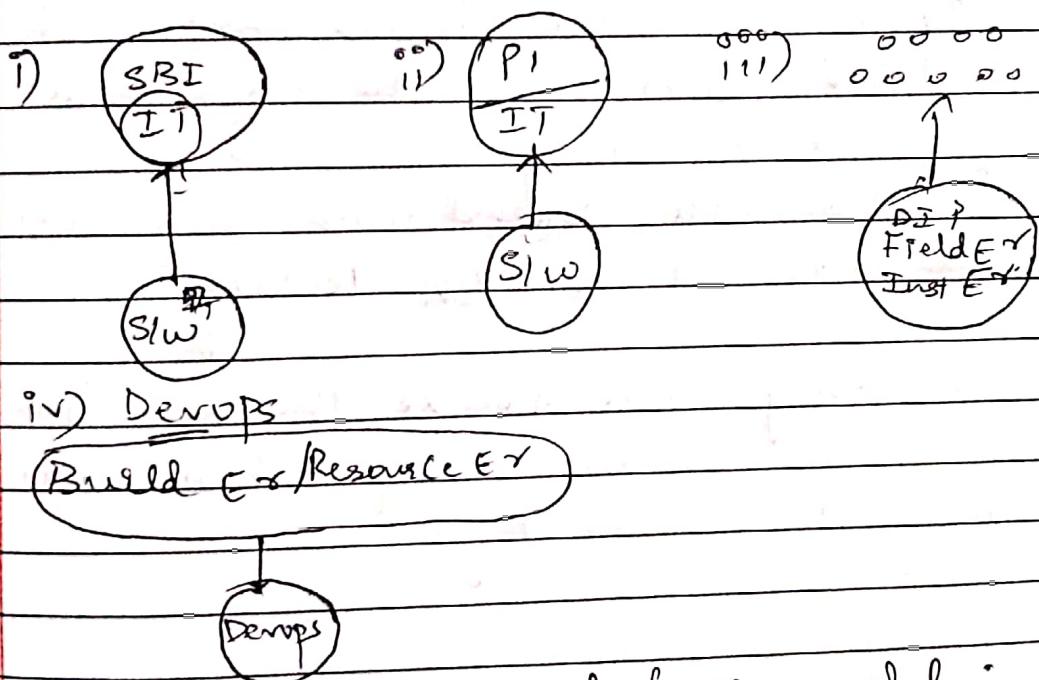


II) DdD (domain Level Design)

It is done by Senior Developers.



Installation



Advantages of waterfall model :

- 1) Initial investment is less
- 2) Simple to understand & easy to adapt

Drawbacks

- 1) Testing is a small phase which is done only after Coding.
- 2) Re-work is done.
- 3) Total Investment is more.



1) Developers are only involved in Testing

Applications

When we are going for Simple Project

2) When we are going for Short term Project.

Q. Why developer should not be involved in Testing

→ 1) They never see product from user's point of view.

2) They can't see build broken (Bugs)

3) They might hide the bugs even after finding the bugs.

4) They will spend more time on development than QA Testing

Q. Why they will freeze the requirement
↳ When you allow changes, then will or may introduce bugs.

Grey Box Testing

Combination of White Box and Black Box is called Grey Box Testing.

① White Box Testing :

Open, Close, Transposed or unit Testing
Testing each and every line of code.

② Manual Testing

③ Automation Testing

Unit Test Cases:

To test code, we provide I/P and check for O/P.

Documentation consists of I/P & O/P of Scenario.

If ($a == 0$)

else

$y = b * 8$

`SOPC("Green");`

else

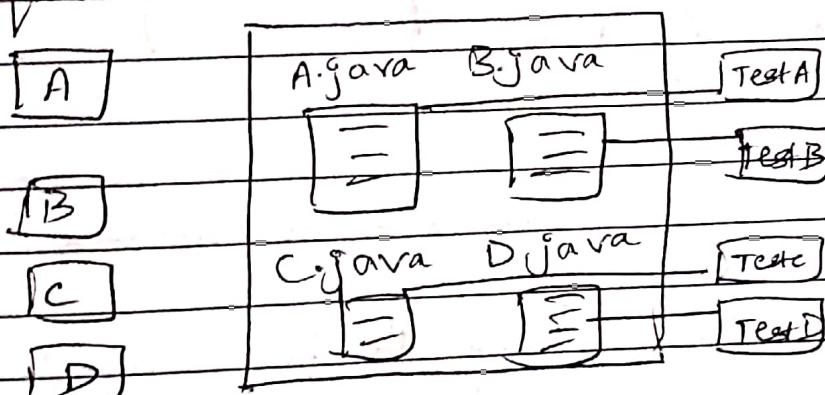
`SOPC("Red");`

}

else

`SOPC("Black");`

Requirement



Types of white Box Testing :

① Path Testing

② Condition Testing

③ Loop Testing

④ White Box Testing Memory point of view

⑤ White Box Testing Performance point of view

⑥ Path Testing : Testing path of independent function till the end is called Path Testing



main()



2) Condition Testing: Testing the logical condition that is true or false is called Condition Testing

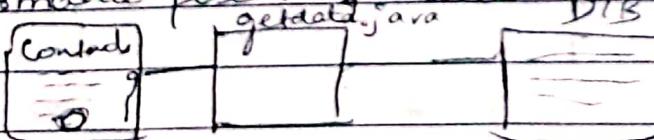
3) Loop Testing: Testing for the repetition or loop is called Loop Testing

4) White Box Testing Memory point of view we will test the code from memory point of view by reducing code size
Ex: `int a, b;`

`b=10;`
`System.out.println(b);`

use Inbuilt function to increase the working or reduce memory.

5) White Box Testing performance point of view checking the code or application from performance point of view

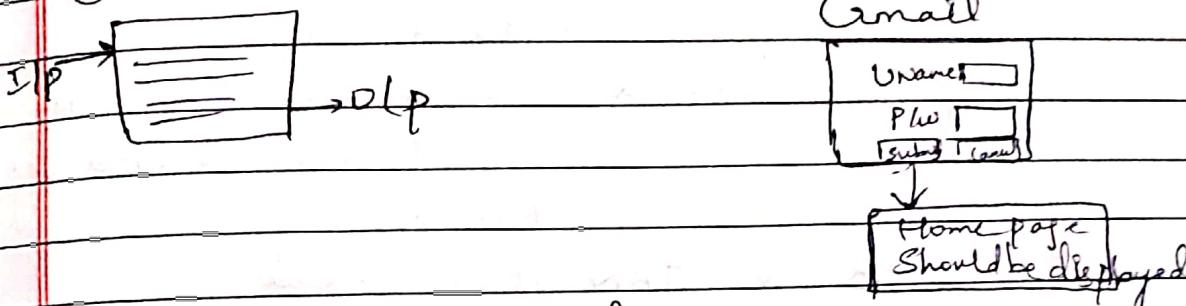


Difference b/w White Box and Black Box
white Box Black Box

- 1) Testing each & every testing function or line of code behavior of application
- 2) Developers are involved \Rightarrow Test engineers are involved in Black Box Testing.



- 3) Source code is Visible 3) Source code isn't visible
 - 4) we should know the 4) we need not know the programming language programming language.
 - 5) we should know the 5) we need not know the internal structure of internal structure of Code.



SDTE Software Development Test Engineers

Defn: Testing the functionality or behavior of the application is called Black Box Testing.

- ① Functionality Testing
 - ② Integration Testing
 - ③ System Testing
 - ④ Acceptance Testing (or UAT)
 - ⑤ Adhoc Testing
 - ⑥ Smoke Testing
 - ⑦ Compatibility Testing
 - ⑧ Usability Testing
 - ⑨ Regression Testing
 - ⑩ Exploratory Testing
 - ⑪ Performance Testing
 - ⑫ Accessibility Testing → Internalization Testing
 - ⑬ Globalization Testing → Localization Testing
 - ⑭ Recovery Testing
 - ⑮ Reliability Testing

City Bank Online - SRS

(CBO)

(Software Requirement Specification)

3.0 Amount Transfer Page

3.0.1 TAN Text Field

3.0.1.1 Should accept 10 digit valid A/c nos

3.0.1.2 Should accept A/c nos which are created by Manager

3.0.2 TAN Text Field

3.0.2.1 Should accept 10 digit valid A/c nos

3.0.2.2 Should accept A/c nos which are created by manager.

3.0.3 Amount Text Field

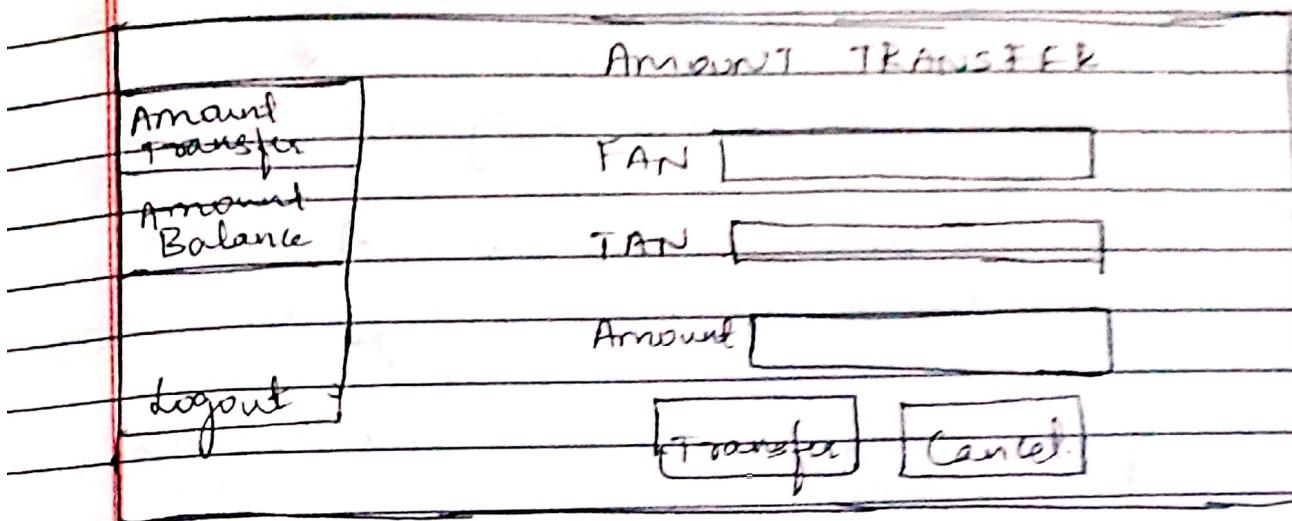
3.0.3.1 Should accept 100-5000 integer numbers

3.0.3.2 Should not accept amount greater than Balance.

- we have to document the scenarios.

① Functionality Testing:

Checking each and every component of an application thoroughly according to requirement specification is called Functionality Testing



when functionality is not working with
Requirement → Defect.

we can do functionality in different ways:

- ① OverTesting ② UnderTesting ③ Optimized testing
- ④ Positive Testing ⑤ Negative Testing

① OverTesting: Testing application with same set of scenarios or test cases in different ways

② UnderTesting: Testing application with functions with minimum scenarios is called under Testing

③ Optimized Testing: Testing application with those scenarios which makes sense.

④ Positive Testing: Testing the application with unexpected data or invalid data

⑤ Negative Testing: Testing the application with unexpected data or invalid data

⑥ In OverTesting, lots of bugs will be missed because we spend lots of time in testing the same scenario.

⑦ In Optimized testing, we will miss less number of bugs.

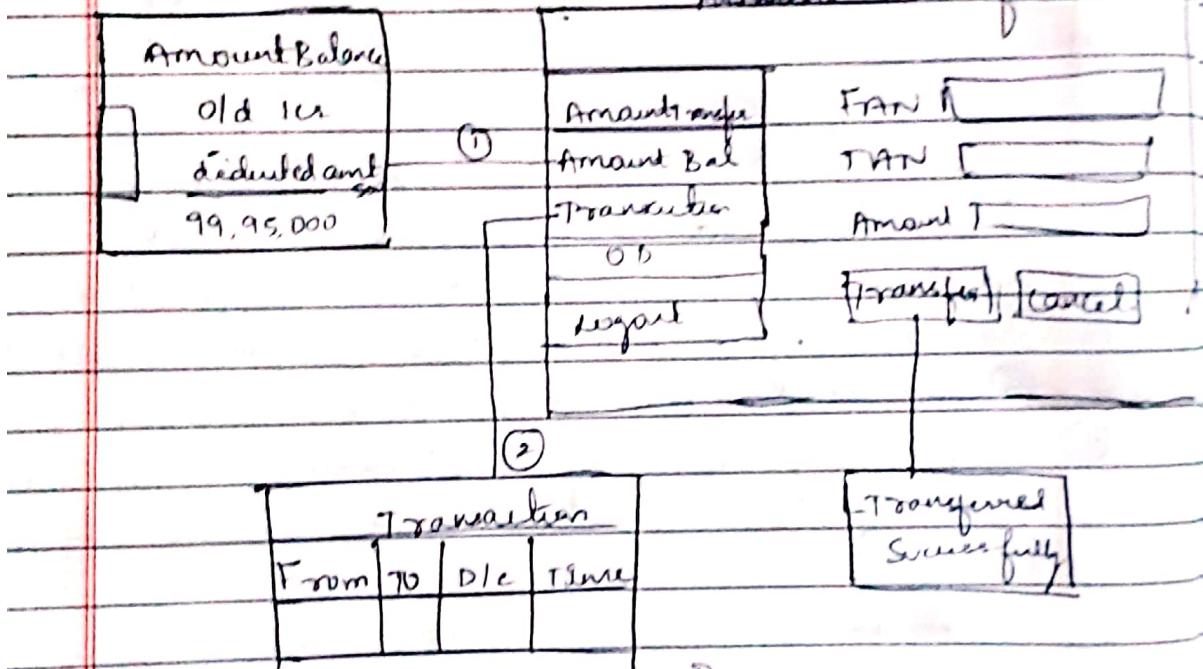
⑥ Integration Testing : Testing the data flow b/w one module to another module is called Integration Testing

or

Testing data flow b/w one application to other application is called Integration Testing

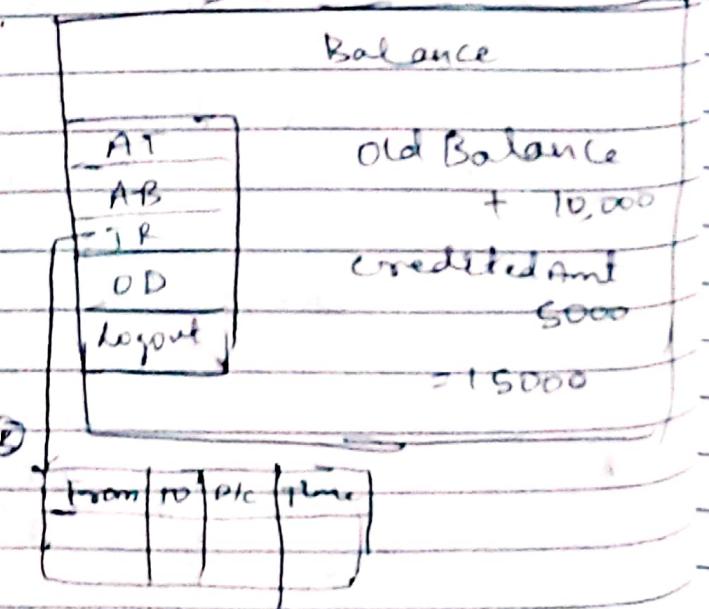
A

Account Transfer



B

Balance



How to do Integration Testing?

- ① understand the application or requirement thoroughly.
- ② understand each component how it works & how it is related to other component.
- ③ Identify all possible integration scenarios.
- ④ prioritize the identified scenarios.
- ⑤ Document the identified scenarios.
- ⑥ Execute the application according to the documented scenarios.
- ⑦ while testing if you get any defects, report it to development team.

Software Testing

- ~~Defining~~ Verifying the functionality of an application against Requirement specification is called Requirement Based Software Testing
- ① The process of finding defects in SW is called SW Testing.
 - ② Execution of a program within an intention to find bug is called SW Testing.

Why Software Testing?

- ① Every SW is built to support business if there are any problems in the SW it will affect business.
So, before you launch it to market, recognize all problems & fix them.
- ② To improve the quality of the product
- ③ To make sure SW works according to customer requirements

Types of Integration Testing

- ① Incremental Integration Testing
- ② Non-Incremental Integration Testing

① Incremental Integration Testing

i) Top-Bottoms

ii) Bottom-up

iii) A Parent

B child

C child

D child

iv) A Child

B child

C child

D Parent

CEO

cataloger

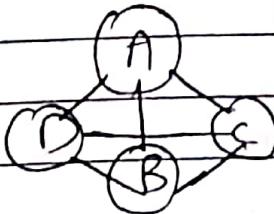
manager

TB, RL

Report

TE, DE

② Non-Incremental Testing (Big Bang)



WhatsApp Profile Functionality Testing:

- ① Test the image upload feature with image files of different extensions like `psd`, `BMP`, `JPEG` etc.
- ② Check image upload with image size greater than the maximum allowed size.

proper message should be displayed

3) Check image upload feature with file types other than images like txt, doc, etc, pdf etc.

4) Test for the cancel button functionality is working or not while uploading the image.

5) Check multiple images upload functionality.

6) Check for image quality after upload. The quality of image should not be changed after uploading the image.

7) keep tapping on upload button several times continuously to see what happens.

8) what are choices for upload your drive, cloud etc.

9) How much time does it take for a picture upload to take place.

10) Simulate a slow network condition & test what happens.

11) Keep uploading pictures & see if the previous picture gets replaced.

12) Profile picture optional during sign up process. default picture icon should be displayed.

13) Check if swapping front & rear camera is possible.

14) Zoom In and Zoom out should be possible.

15) After cropping, image is rendered correctly or not.

16) If user delete picture, it should be deleted & be replaced with default one.

17) If no profile picture, tapping on delete should show error.

- 18) Clicking cancel button during upload
- 19) Try to upload video instead of pic
- 20) upload image of large size

Different types of Integration Testing:

1) Incremental Integration testing:

Take 2 modules, check if the data flow b/w 2 is working fine - If it is working then add one more module and test again - Continue like this incrementally, add the modules test the data flow b/w modules.

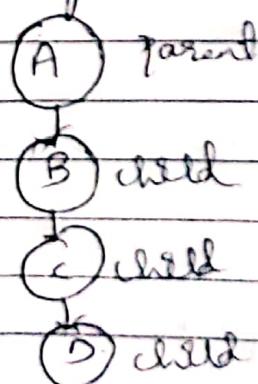
There are 2 ways:

1) Top Down Incremental Integration Testing

2) Bottom up Incremental Integration Testing

1) Top Down Incremental Testing:

Incrementally add the modules and test the data flow b/w the modules make sure that the module that we are adding is child of previous one

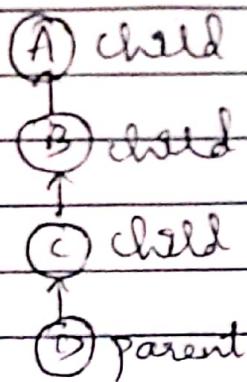


2) Bottom up Incremental Testing:

Testing Starts from last child up to parent. Incrementally add the modules



test the data flow b/w the modules. Make sure that the module you are adding is the parent of previous one.



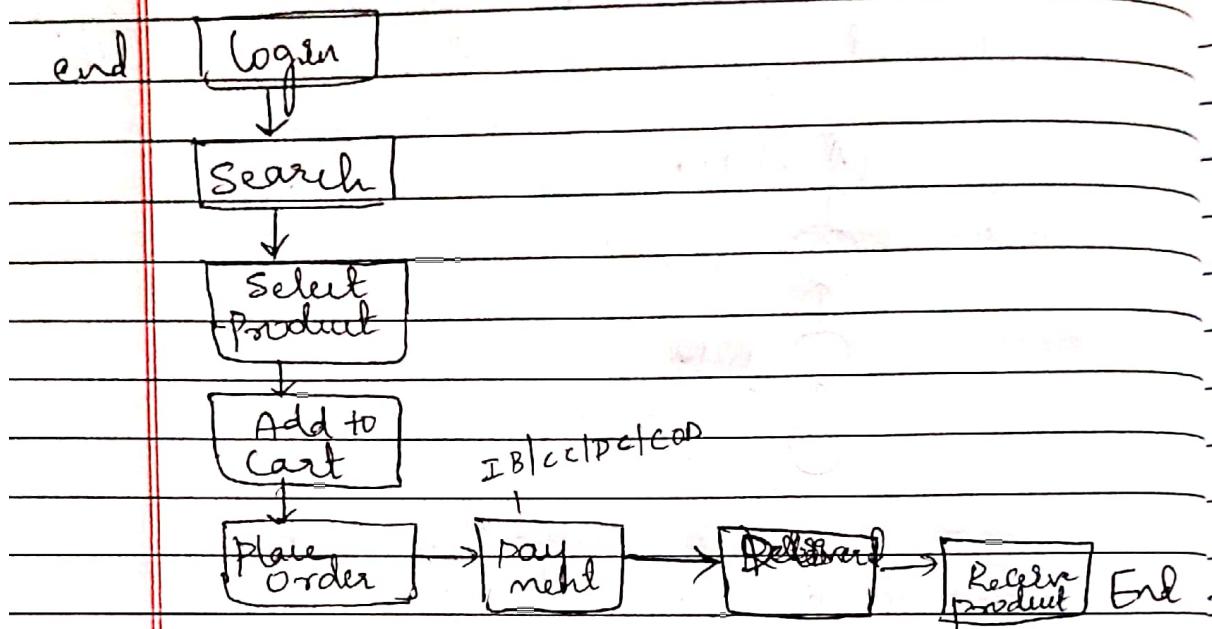
- 2) Non-Instrumental Integration Testing:
- we use this method when
- i) when data flow is very complex.
 - ii) when it is difficult to identify whose parent & who is child.
 - iii) This is also called as Big bang method
- Here, Combine all the modules at a shot and Start testing the dataflow b/w the modules.

Disadvantages are:

- 1) we may miss to test some of the modules.
- 2) Root Cause analysis of the defect is difficult. Identifying the bug from where it has arisen will not get to know the origin of bug.

- 3) System Testing: It is an end to end testing where test environment is similar to production environment.

Ex: Flipkart



① Scenario 1 OD 40000

800 Interest

200 Activation And

41000.

② Scenario 2 OD 40000

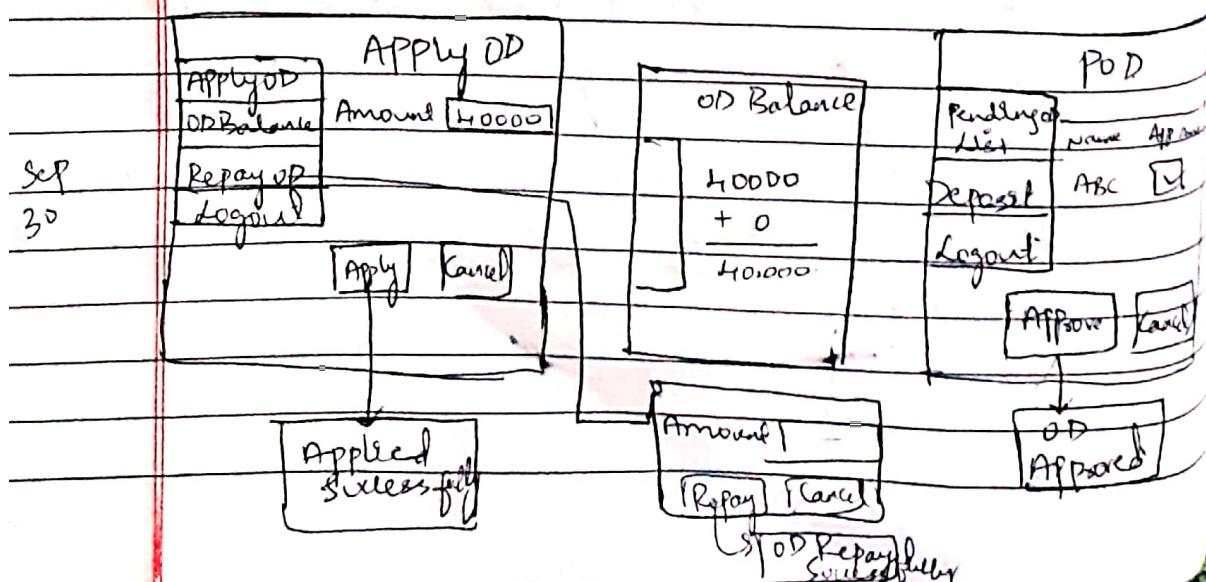
800

40800

③ Scenario 3 OD 50000

500

50500

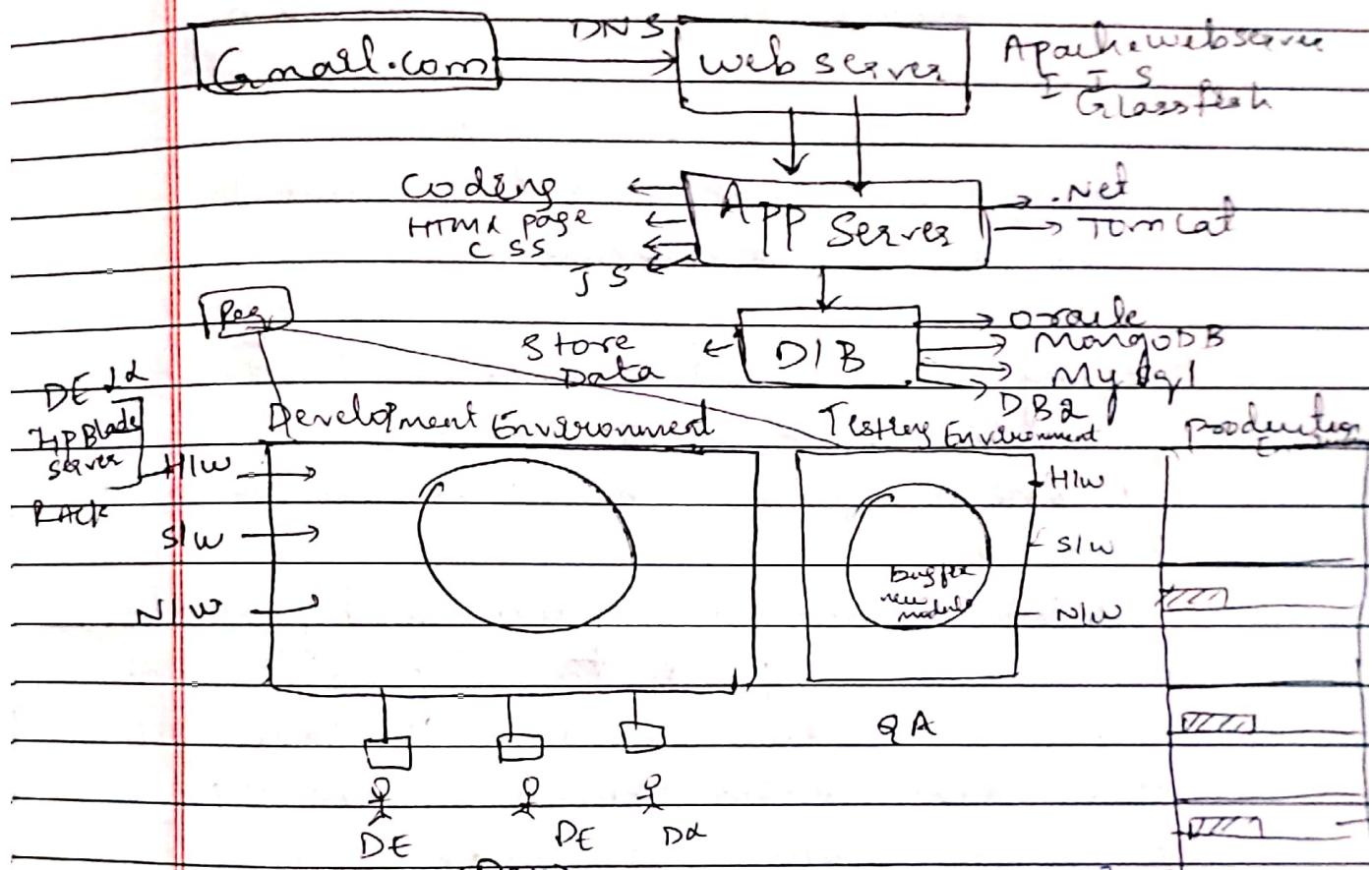




Development Environment :

It is a set up which is used to develop the Software

It consists of HW, SW and network.



Testing Environment

It is a setup which is used to test the application which consists of HW, SW & NW.

Production Environment :

It is a setup which is used to run the application which consists of HW, SW & NW.

Production Environment and Testing Environment

Should be same in respect to HW & SW.

- whenever developer develops, then he will compile & compress & give to testing team. It is called 1 build.
- Testing team will find bugs and report defect report to Development team.
- Development team fixes bugs and adds new modules to fix bug.
- Testing team first tests new module, bugs, impacted modules.
- Testing environment should be similar to Production environment means,
 - i) The hardware should be similar to production
 - a) The make (manufactured by) should be similar to production Server (for ex: if the production Server is HP Blade Server then test Server should also be HP)
 - b) Configuration and make should be similar but different capabilities (i.e., no of cpus)
 - ii) The Software should be similar to production
 - a) The OS should be similar
 - b) Application Server should be similar
 - c) Web Server should be similar
 - d) DB Server should be similar
 - iii) Data should be similar to production
 - a) we should create data similar to production
 - b) we should create a script to create a dummy data which is similar to

Q=

the production environment.

In real environment, we may make loads of entries into DB, but while testing we can't enter manually lots of entries so we are going to write test script programme which generates 1000s of users or data. This can be used for testing.

Installation of Application in Test Environment

Done by :

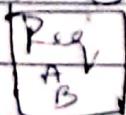
① Test testing Environment who are involved in installing SW

1) Any Body from testing team

2) Any Body from Development team

3) Build / Release Engineers

Defn: Build : A piece of Software which is Compiled, Compressed and Installed in all environment is called Build.



D:\Build
BD.ZIP

Dev

Test

Final

A.java
B.java

DDB
DE DE DE

BDB
TE TE TE

Github is a version Controlling tool.

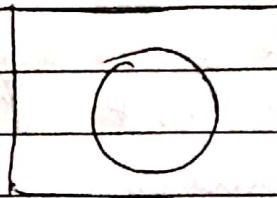
In Github, builds can also be stored.

It is a cloud app & cloud Server.

Build Engineers are responsible for managing the code that code will present in github.

Dev

SVN
GitHub



Jenkins : Automate the build process.

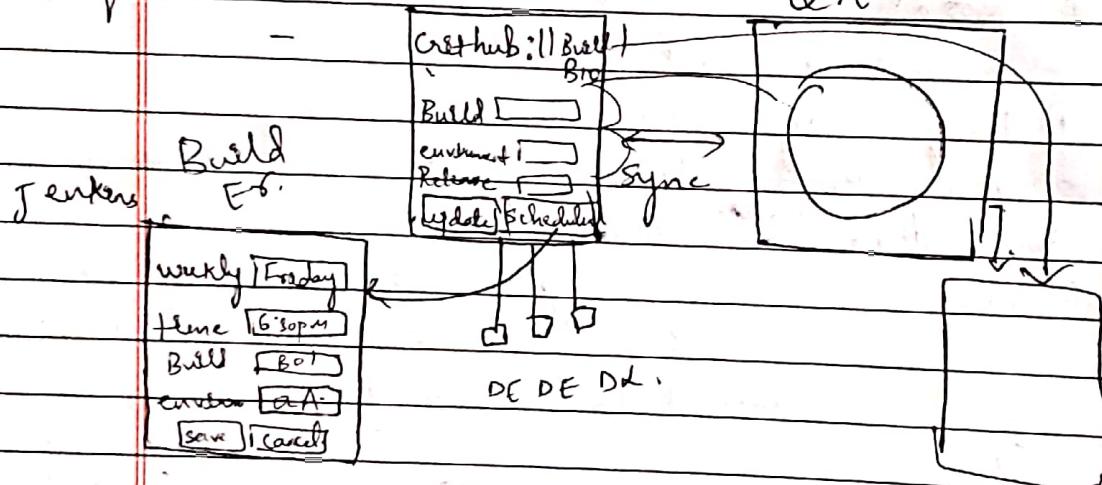
Continuous Integration tool? Syncing b/w dev environment and QA environment.

- ① There should be sync b/w dev & QA, build versions should be same.

Req

Dev

QA



Staging/qa -

Latest build will be deployed (Production) to Staging environment

- ① When there is only minor fix then we can do patch.
- ② These code will be compiled, compressed and sent to QA.
- ③ In QA there is no need of installing, just install the JSLW testing environment.
- ④ If there is a new module, that is not affecting any other modules then just



apply patch:

Jenkins

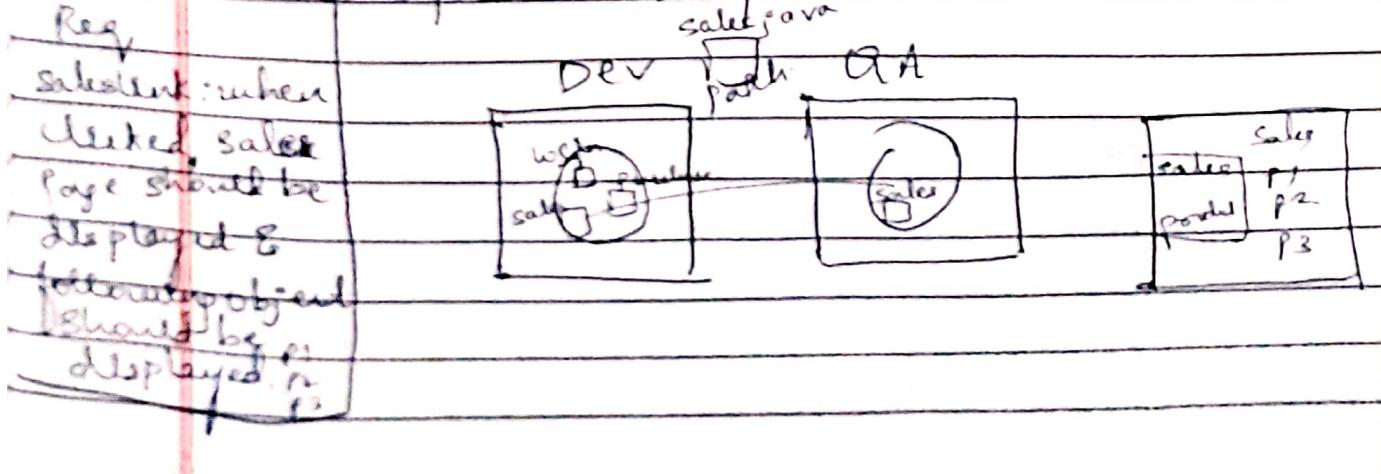
Defn: It is a Continuous Integration tool that is used to automate the build process.

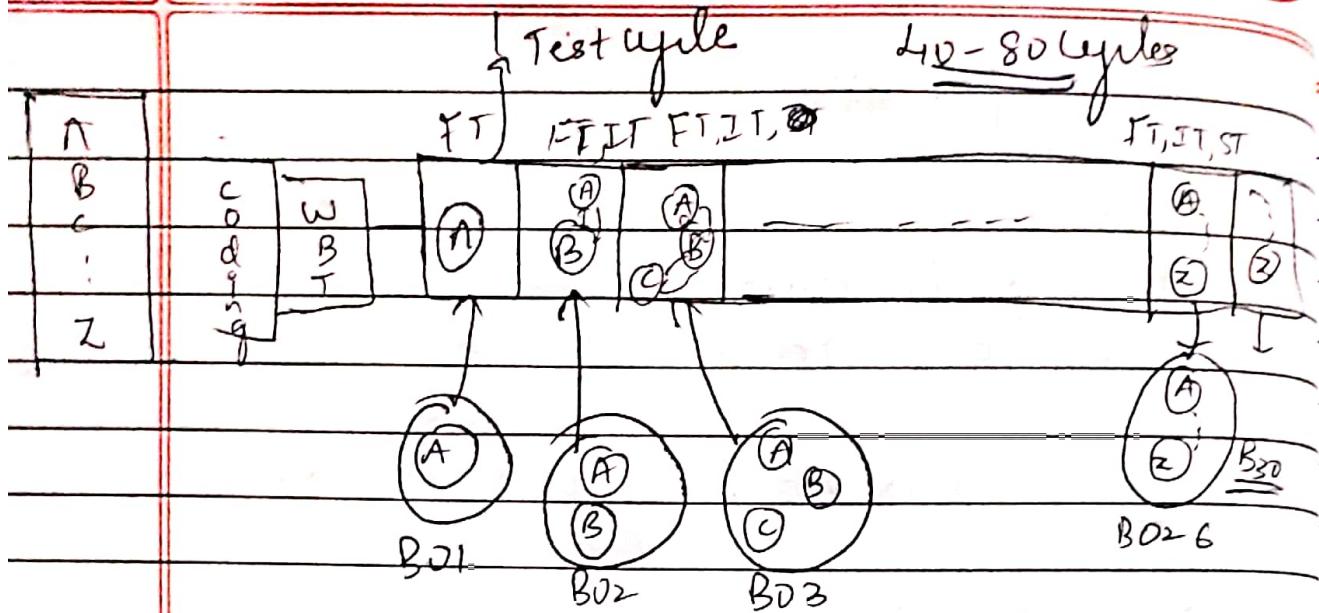
Devops: It is nothing but a development operation which involves activities like development, managing the code, Build the new build, install & uninstall the application in all the environments {Development, QA Environment}

Continuous Integration tool: It is a process which involves development operations where it will make sure that all the environments are synchronized continuously.

Defn Patch: Patch is nothing but it is a modified program or deleted program or added program which will be done in less time.

- when patch is given to QA we never going to uninstall old build and just we will update the patch.





Test cycle

Time spent on testing a build or software

Completely go called Test Cycle:

Defect types:

(1) Blocker: If blocks entire application

It can be 0

(2) Critical: If actual work like sending mail, sending mail is not working.

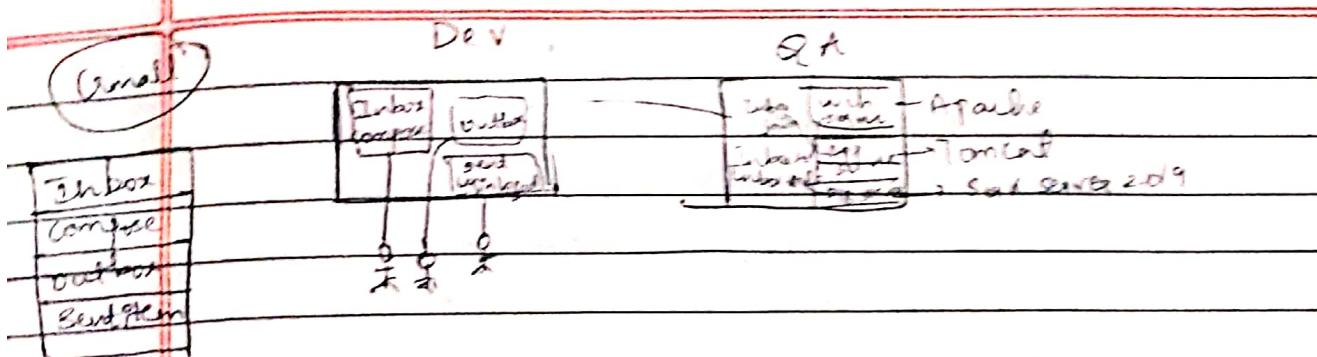
It should be 0

(3) Major: If any option is not working
It can be 0-10

(4) Minor: If process or HTML graphics not attractive or some spelling mistake is there.

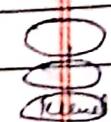
It can be 0-40

Standalone web Application client server
Architectures



Kernel → Interpreter blw H/w & S/w.

In Linux there are 7 layers but in Windows there are ~~8~~ 3 layers.



(7) Acceptance Testing

Acceptance Testing is done by the Customer. Here they use the S/w for the business for a particular period of time and check whether the Software can handle all kinds of real time business scenarios or situations.

Approaches: why to do Acceptance Testing:

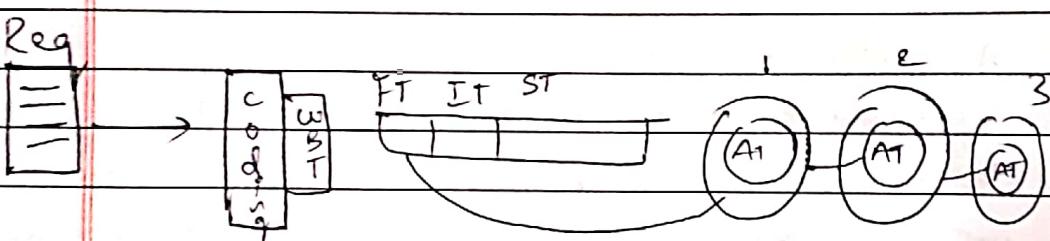
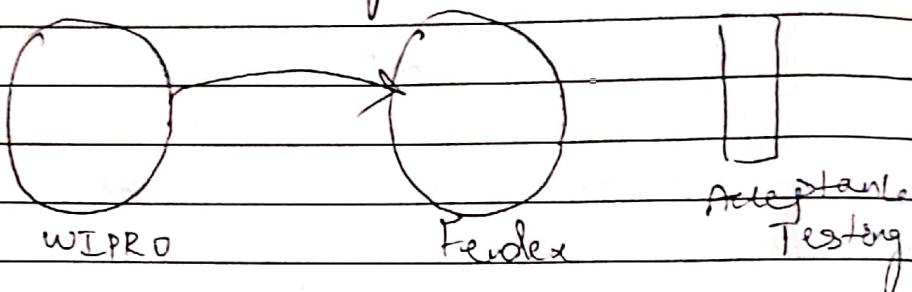
- ① Sometimes due to lot of business pressure, S/w Companies will push the S/w so, to find out the defects, customer will do acceptance testing.
- ② Developer misunderstands the requirement and develops the wrong requirement then the customer will do acceptance testing.
- ③ During testing, test engineers will miss some of the critical bugs due to negligence or misunderstanding of requirement. To find out those mistakes, customer will do acceptance testing.
- ④ The difference blw WIPRO Test engineers and



Federal Test Engineers:

a) WIPRO testing team do functionality testing & System Testing.

But at Fedex, the testing team will do only End-End Testing.



CR → Change Request

RFE → Request Feature Enhancement

FT, IT, ST, AT → Black Box Testing

(Functional Testing)

[Performance, Usability] → Non-Functional

AGTLE Model

Defn: Agile is a model where we develop and test the Software in an incremental or iterative way. They came up with this model in order to overcome the drawbacks that were there in traditional model.

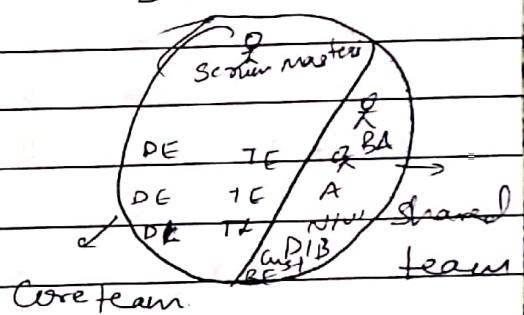
- Here they build the product in Shuote cycle - 1 month (4-8 cycles)
 - Here both development and testing

team is called as Scrum.

CBO Product Backlog

	Home loan	Bank	M&L
	Apply HL	Apply PL	Apply ML
	Approve HL	Approve PL	Approve ML
Repay	Repay	Repay	Repay
HL Balance	PL Balance	ML Balance	
Logon	Logon	Logon	
Logout	Logout	Logout	
Insurance			
	V.I	H.I	D.I
	Apply VI	Apply HI	Apply DI
	Approve VI	Approve HI	Approve DI
	Pay Premium	Pay Premium	Pay Premium
Core Banking			
Create A/c			
Approve A/c			
Balance			
Logon			

Scrum team



product Backlog: Documentation which contains the prioritized stories.

Scrum: It is a process involved in Agile where development & testing takes place.

Spent Planning Meeting $\frac{3}{8} \times 7 = 21$

Spent Backlog

Stories	Assign Task	Task	Polymer	Estimator
Create A/c.	DE	Coding (4)	4 hours	
		UML design (4)	2 hours	
Approve A/c	DE	DB design (5)	20 hours	Start
		UML (5)	10 hours	
Logon	TE	WBT (5)	10 hours	Polish
		Write TP (1)	2 hours	
Logout	TE	Write TC (2)	4 hours	
		Review TC (1)	2 hours	
		Execute TC (1)	20 hours	
		Report Defect (6)	5 hrs	
				<u>163 hrs</u>

Retrospect Meeting:

Mistakes and Improvement will discuss and discussed etc.

Types of meeting:

- 1) Sprint planning Meeting
- 2) Daily Standup Meeting
- 3) Release meeting.

Definition

① Product Backlog: It is a prioritized list of features or requirements or stories. Here stories need not be in detailed manner.

• product Backlog is owned by manager by product owner.

② Sprint Backlog: It is a list of stories & the associated tasks that must be completed within a sprint.

• It is a list of stories & tasks committed by Scrum team to be delivered in one sprint.

③ Sprint Planning Meeting: Here entire Scrum team meets together and pull the stories from product backlog

a) whatever they can build within a sprint

b) They assign the stories to the engineer.

c) The engineers deliver the task to be performed, to complete or build the complete story.

d) They estimate the time taken to



Complete each story

- (4) ~~Screen Master~~ drives the Sprint Planning meeting. His prime role is to facilitate complete meeting and coordinate b/w Stakeholders.

Stakeholders → customers

- ④ In this meeting, product owner classifies if any questions or queries are there with respect to Stories.
 - In this meeting, development engineer should derive the task for building every Story.

Ex: Assign Task Task

→

coding, UI/UX design,
DB design.

- DE prioritizes the stories and tasks
 - In this meeting, test engineers should derive the task for testing the feature built for a story.

- (5) Retrospect Meetings: Entire scrum team meets and discusses about achievements (Good processes followed) and mistakes (wrong processes followed) and we document it and that document is called as Retrospect Document.

- when next release / next sprint starts, while doing sprint planning we refer this document and we plan it in such a way that old mistakes are not repeated and good activities are again adopted
 - ③ Daily Stand-up meeting (10-15 min): team masters drive this meeting.

- Scrum master facilitates the meeting
- Discuss daily, yesterday's and upcoming task whether it is completed in progress or incomplete.
- Discuss about any impediments (some thing which is not possible to develop test) (Scrum master will solve it) if it is solvable then and there solve it else take a note on those issues and solve it later.
- Each engineer should tell
 - a) what he has done yesterday
 - b) what were the impediments faced
 - c) what you are planning to do today
 - d) what are the expected impediments
- Generally this meeting should go for maximum 10-15 minutes.
- Generally this meeting is done in the beginning of the day.

② Scrum Team: It is a group of members or individuals working together to deliver or complete the committed stories.

- Generally, Scrum team size will be b/w 7-9 peoples.
- Generally there are 2 kinds of team:
 - ① Core team
 - ② Shared team.
- ① Core team involves Scrum master, developers, test Engineers.
- ② Shared team involves architect, DB/DBA, Administrator, Product owner.

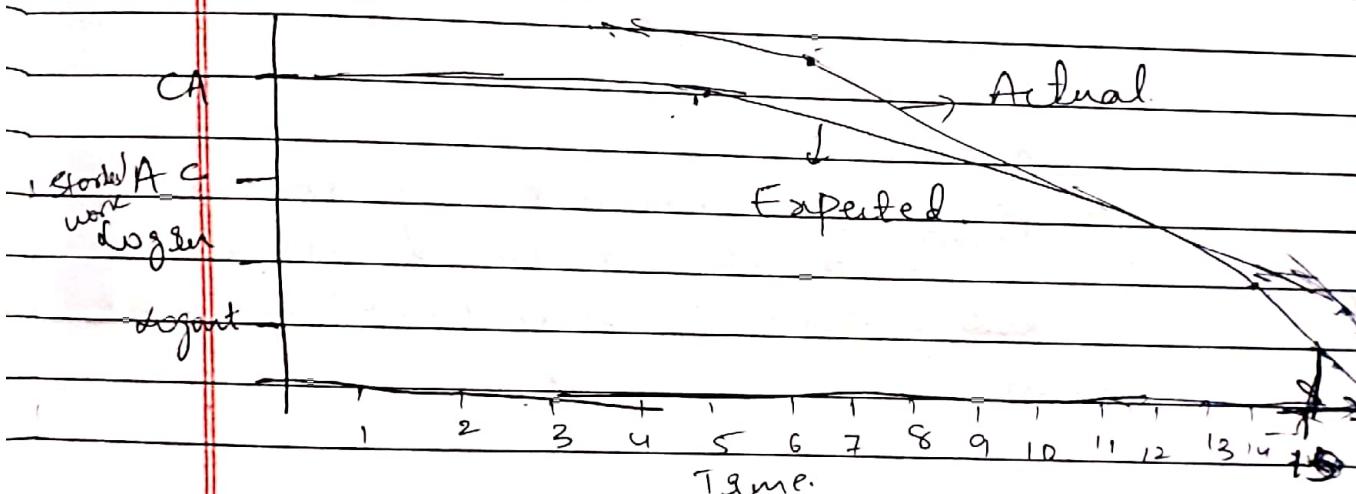
UI/UX designers

- who leads the Scrum team:
- Scrum master leads complete Scrum team and he facilitates every task to complete

Burn down chart

Graphical representation of work vs time. It is used for product view

Burn down chart



Start	In Progress	Completed
WT (RTC)	C - Create A/c WT & TC.	
APPROVE A/c PL & P		

c)

Scrum Board / Kanban / Story Board
(System)

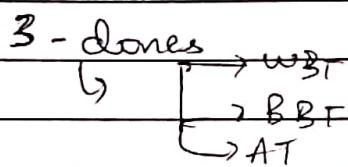
It contains list of pending and completed task of each engineer.

Stand Review Meeting:

- It is last stage of Agile.
- Here entire Scrum team meets at the end of Sprint and discusses about the

about how well Sprint went.

- Engineers will give demo of what ever they built to the Product owner.
- Product owner tells what is done & what is not done.
- Also they discuss about how to plan the next Sprint.

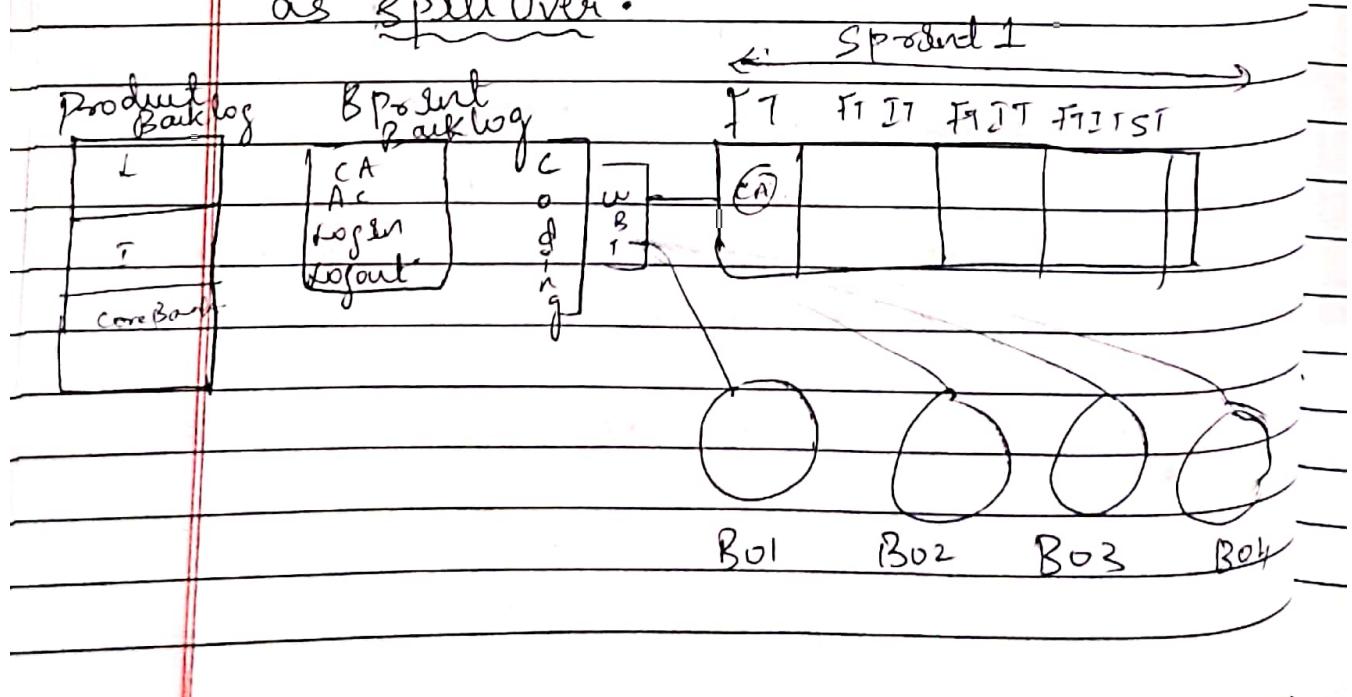


Chicken

Some people who don't actually do the work ^{or there in Sprint} in project but they will be there to observe what is happening in the Sprint.

Spill Over:

There are certain features or stories that you cannot build in current Sprint and it will be moved to next Sprint as known as Spill Over.



In automation, bugs ~~can~~ will be converted to defect tool.

Q) Smoke Testing: Testing the basic / critical feature of an application before doing thorough testing is called Smoke Testing.

- Smoke Testing is also called as Sanity Testing / Dry run Testing / Skim Testing / Build verification Testing.
[whether the application is testable or not]
- we will do Smoke testing in Acceptance testing also and do Smoke testing in production environment.

Why to do Smoke Testing?

- a) just to ensure that the product is testable or not.
- b) we will do smoke testing in beginning to catch bugs in basic features and send to development team so that development team has sufficient time in fixing it.
- c) Just to ensure product is installed properly or not.

Important points

- 1) when we do Smoke testing, we will do Positive Testing only. (because it should work according to requirement)
- 2) Here we test ^{only} basic / critical feature.
- 3) Here we will take basic feature & test for important scenarios

- 4) whenever the build comes to the customer before the customer does acceptance testing he also does Smoke testing.
- 5) when the product is installed in Product we should do quick Smoke Testing to ensure product is installed properly. Also it acts as health check for SW. It will show whether application is good or not.

F) Adhoc Testing (It is also called as monkey Testing):

Testing the application Randomly is called Adhoc Testing. {Gorilla Testing/Monkey testing}

why we do Adhoc testing?

(F) End users use the application randomly, they may find the defect, but test engineers use the application Systematically so he may not find the same defect.

In order to avoid such scenarios test engineer should go & test application randomly (i.e., behave like end users).

(2) Development team looks at the requirement & do development. Testing team also looks at requirement & do testing. By this method, testing team may not catch many bugs. They think like everything works fine. In order to avoid this we do Adhoc Testing.

(3) Adhoc is a Testing where we don't follow the requirements.



When to do Adhoc Testing?

- ① When product becomes stable, then only go for adhoc testing.
- ② When the feature is new, we will not do adhoc Testing.
- ③ While doing smoke testing, we will not do adhoc Testing because if we do Adhoc Testing we will not get time to test basic features after testing already.
- ④ Whenever we are free. We will testing the product if some time is left out then we should spend time in doing adhoc testing.
- ⑤ If you get too many adhoc scenarios during beginning stage then note down the scenarios and execute it whenever you get time.
1) Logical 2) Illogical.

⑧ Reliability Testing :

Crmstl

Uname:	<input type="text"/>	Home Page
Pwd:	<input type="text"/>	<input type="button" value="Logout"/>
<input type="button" value="Login"/>	<input type="button" value="Cancel"/>	<input type="button" value="1st"/>
		<input type="button" value="2nd"/>

Testing the application continuously for a functionality for a particular period of time is called Reliability Testing.

9) Recovery Testing :

Testing the application to check how it recovers from crashes / disasters is called Recovery Testing.

10) Accessibility Testing :

{ American Disability Act }

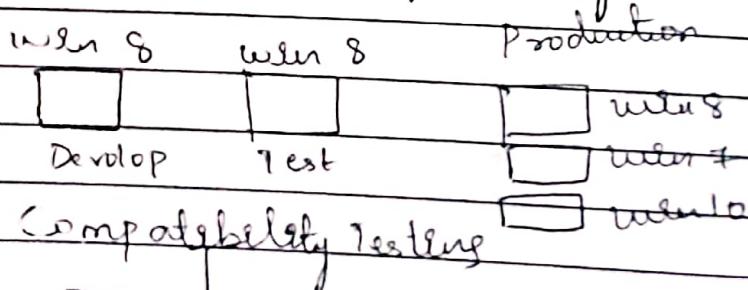
Testing the application for physically challenged people is called Accessibility Testing.

11) Exploratory Testing :

Testing the application without following any formal document / requirement is called Exploratory Testing.

12) Compatibility Testing :

Testing the functionality of application in different HW & SW ^(environment) platform is called Compatibility Testing.



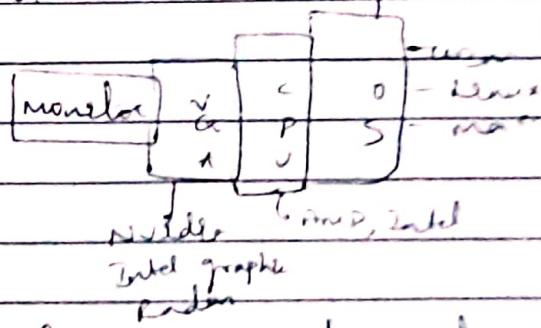
why we do Compatibility Testing?

i) Developers develop the application in one platform. If test engineers test the application in same platform when it is launched to the business. Any customer may use the application in different platforms. Because of this certain features might not work and this spreads bad name in the market. The customers who will buy the product will reduce. To reduce such situations we should do Compatibility Testing.

ii) To check whether the features are working consistently in all platforms we do Compatibility Testing.

iii) Developers would have written common code for all platform or browser. We may have to test it on all platform to confirm that really works.

iv) Developers would have written platform specific code. We may have to check whether code works in corresponding platform.



R.I.T. Based on google analytics, product based companies get know the the regular visitors and usability of application

When we do Compatibility Testing

- i) when the product becomes stable in base platform, then we think about testing application in different platform.
- ii) Different hw & sw renders the GUI in diff. we should check whether our app rendered properly in diff. How to do Compatibility Testing on different platform?
- Nowadays VirtualBox or VMware is used

→ mobile devices v/s OS version matrix document:

Install the VMware and install many OS and test your application.

i) Testing done using cloud like cloud testing

ii) Using Simulator, we can do testing

iii) On Real basis we can do testing

iv) Buy the different mobiles with different OS and test the application

Hardware Testing:

Hardware Compatibility Testing:

• Testing the functionality of an application in different hw environment.

• Here we might test the application in different hw like different Processor

 i) Different Speed different make (Intel, AMD)

 ii) Different bit size (64bit, 32bit)

 iii) Different Mother Board

 iv) different make { Intel, mercury }

 v) different VGA Cards vi) different resolution

13) Usability Testing:

Testing the user friendliness of an application is called Usability Testing.

Eg: we have 2 applications A & B which are different but doing the same job, we see which one is user friendly.

Given below are some of the parameters we look into for testing :-

- i) Speed ii) Help iii) Navigation Should be Simple
- iv) Compatibility v) Look & feel vi) features
- vii) Location of Components.

- One important parameter other than the above said parameter is "Effort needed to learn the Application"

- Consider an example :- Blw A takes 2 hrs to understand but we take 2 hrs to understand Blw B.

- Let us see different cases here

- i) Since we understand Blw A in 2 hrs. It becomes user friendly compared to B.

- ii) Suppose look & feel is not good to A.

- In this case, though we understand A in 2 hrs we can't say that A is user friendly

- iii) So we look into many parameters before we say user friendliness of a Blw what is look & feel?

- The application should be in such a way that it should be pleasant looking

- Q How to conduct usability Testing?

- Prepare or derive checklist.

- If you don't prepare a checklist, we may miss certain features in application.



Ex: of checklist for an application:

- i) For this application, one of the checked includes color of already checked link should be changed to red.
- ii) All images should have alt tag.
- iii) All pages should have link to home page.
- iv) Login feature should have forget password link.
- v) Likely about checklist, we can derive as many checklist as possible based upon the application product.
- vi) While deriving checklist, we should derive a common checklist which can be execute for all pages.
- vii) These is another case where customer gives checklist for an application

Home > search > contact > about

Breadcrumbs

Test Cases

Document that contains

what are the drawbacks if we test the application by seeing the requirement?

i) There will be no consistency in test execution

ii) Quality of testing over the period of time depends upon:

a) Memory power of Test Engineer.

b) Quality of Testing depends on mood of Test Engg.

c) Quality of Testing depends on



person to person. If engineers are more experienced, they derive more scenarios. If engineers are less experienced, they will derive less scenarios.

What is Test Case?

It is a document which contains all possible scenarios with specific requirement.

- Test Cases also contains different sections like: 1) Step number 2) Description
- 3) Input 4) Expected Result 5) Actual Result
- 6) Status 7) Comments
- Initially test cases were written in Excel. But now we use Test Management tool.

When to write the test cases?

- 1) When developers building the product, the tester will start to build the test cases.
- 2) When developers add a new feature, test engineer will write test cases for new feature.
- 3) When developers modify the feature, test engineer will modify the test cases.
- 4) When developers delete the features, test engineer will delete the corresponding features.
- 5) When developers develops the product & give the build to tester, tester will start testing the product according to the test cases written.



Testcase Review process:

- ① Developer start building the product
- ② Tester will start writing test cases
- ③ Once he/she finishes writing all pre-built test cases, tester will give the test cases to his/her TD through email.
- ④ TD will assign the test cases for review to Senior Test Er
- ⑤ Once the Senior Test Er reviews the test cases & gives whether comments or additional or modification or missing scenarios
- ⑥ Now Test Er should correct all the notes & get reviewed by Sr TE; the Sr TE should review again and check whether all the corrections are made and send it to Team lead for approval

Peer Review:

Both the TE, Sr as well as Jr TE will send their test cases to each one of them and get reviewed by each other.

Why we write test cases?

- ① We write test cases to have better test coverage when requirement comes by, the developers are busy in building the product. At the same time test Er are free so they identify all possible scenarios and document it. When the build comes you can spend time in executing the scenarios, because of this, the number of scenarios you are



Coverage is more -

- ⑥ To have consistency in test execution that means, if we document the scenarios, we can make sure that we are executing all the scenarios in all the test cycle / build / product / release.
- ⑦ To depend on policies rather than Do Differently.
- ⑧ To avoid training to every new ETS on the product.
- ⑨ Test Case is the only document which acts like a pool for Customer, development team and management team. So that we can ~~fill~~ the form that we have covered all the possible scenarios.
- ⑩ Test Case acts like a base document for writing the automation script if you refer the test case and write automation script, we can ensure that same kind of coverage is there even for automation script.
- ⑪ If we have documented the test case, you don't have to remember the scenarios.
- ⑫ If we have documented the test case, test case execution happens in a very organized way.
- ⑬ If we have written the test case, time taken to execute the test case will be very less.

Format of Test Case:

Date _____
Page _____

Test Case name: c BO - Amt Transferfield Severity: Critical

Requirement no: 3.0, 3.2, 3.0.3.1, 3.0.3 (BO SK)

Test Data:-

Header

Precondition:

Brief Description: Verify Am-

Test Case Type: Functionality

textfield

Step no	Description	Input	Expected Result	Actual Result	Status
Body 1	Enter -ve integer value into Amnt text field	-100	Should throw appropriate message.		
2	Enter characters into Amnt text field	hundred	Is only	Should throw appropriate message.	

footer Author: chenky

Approved by: Pankaj

Reviewed by: Pankaj

Approved date: 11/09/19

why to write | Do apply Test case design technique

To optimise the test cases we follow Test

Case Design Technique

Test Case Design Technique:

(1) Error Guessing

(2) Equivalence partition

(3) Boundary Value Analysis (BVA)

(4) Error Guessing: Guess all possible defects or errors - we guess the error based on

i) experience ii) Intuition iii) By Requirements

Ex: For amount transferfield:

i) Blank ii) characters iii) # \$ iv) -100

② Equivalence partition:

i) Preussen Method

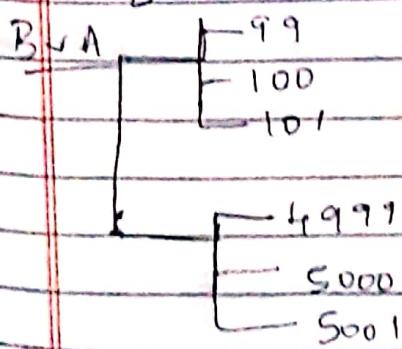
ii) practice Method

Reg.

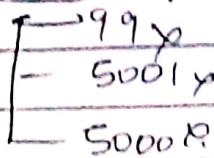
- i) If I/p is range values
then design test case for
valid value & invalid
value
- ii) If amount is 1000-2000
deduct 1% & transfer
- iii) If amount is 2001-3000
deduct 2% & transfer
- iv) If I/p is set of values
then design test case for
valid & invalid values
- v) If amount is 3001-4000
deduct 3% & transfer
- vi) If amount is
4001-5000
deduct 4% & transfer
- vii) If amount is
>5000 throw error
- viii) If I/p is b/w range of value then
go for Preussen Method
- ix) If there is some variation in input
go for practice method.

Boundary value Analysis (BVA)

If the input is b/w range of values b/w
A-B then we are going to design
Test Case for A, A+1, A-1 and B, B+1,
B-1.



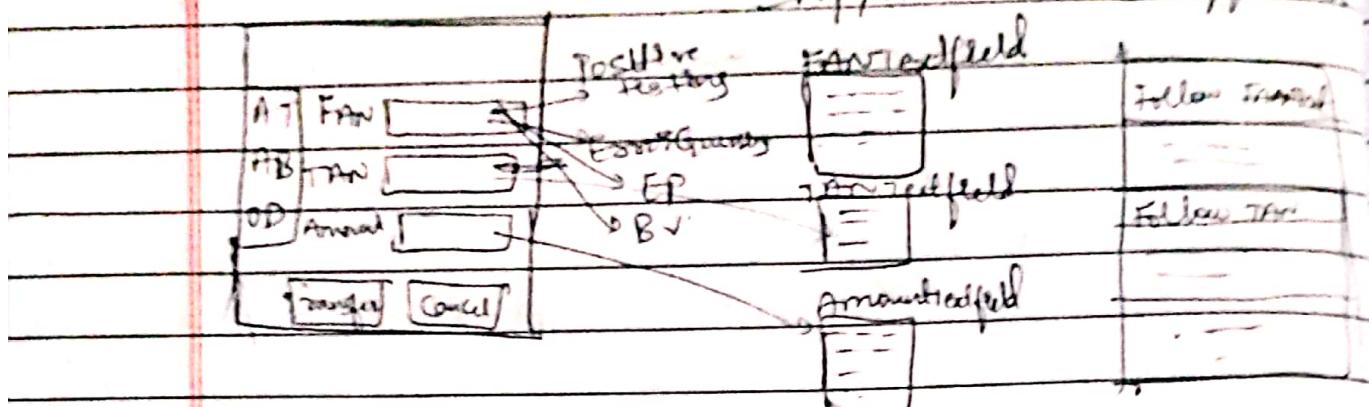
EP (Equivalence Partitioning)



If (amt > 100 and < 5000)
transfer();
else
throw error;

I Approach

II Approach



Lesson 1:

Before we actually start writing the test case we have to come up with options.

Pass/Fail/Block ^{no}

Step no	Description	Input	ER	Status	Comment
---------	-------------	-------	----	--------	---------

① Open the Browser ^{http://www} login page.
Enter URL ^{www} ^{should be displayed}.

② Select the valid user & valid pin.
click on login ^{pwd.} Home page ^{should be displayed.}

③ Click on Add Transfer ^{link}.
Amt transfer page ^{should be displayed}

④ Test the following
FAN text field.
^{should not receive}

⑤ Enter valid AMT ^{amount}
lesson 2 : Select transfer button

Always start writing the test cases with navigation steps.

Lesson 3:

Always use word Should be (must be) in Expected Result (ER), do not use may be or can be.

Lesson 4:

Always write generic test case don't hard code the test case.



Lesson 5:

Elaborate only those steps in which you have fails, do not elaborate all the steps.

Lesson 6:

When we write the test case, we have to always imagine the application.

Lesson 7:

If we organize the scenarios properly, the total number of steps will be reduced.

Lesson 8:

Some company might use input column or Some Companies will remove actual columns.

Lesson 9:

If you cover scenarios in functional testing do not cover same scenarios in Integration testing.

If you cover scenarios in integration testing do not cover in System Testing.

What is the approach to write functional test cases?

- (1) Go to the body of test case.
- (2) Start with navigation steps.
- (3) Take first field and start writing test cases.
 - 3.a) Start with valid data
 - 3.b) Cover the error guessing scenarios
 - 3.c) Cover all equivalence partitioning scenarios
 - 3.d) Cover all boundary value analysis.
- (4) Take second field (component) repeat 3.a, b, c, d.

Test Case Review process

(i) On what basis they assign. Some one for to review.

1) There is a fellow working on similar module in the Project.

2) There is a fellow worked on same module in the previous project

3) There is a fellow working in the Project since beginning & knows every corner of the product

4) There is a fellow who is very responsible he well understand the requirement very fast and identifies more scenarios mistakes.

How do they ensure reviewer does his job?

i) It should assign primary and secondary reviewer.

ii) It should also randomly review and find mistakes.

iii) Intentionally introduce the mistakes and give it to the reviewer and check whether it is found (reviewed) by reviewer or not.

Test Case Review Ethics

1) Always review the content not the author.

2) while reviewing Spend time in identifying the mistakes not in finding the solution for it.

3) Even after the review, if there are any

mistakes, both author and reviewer are responsible.

what will you do while reviewing the test case? or

why we review test case?

- ① we will look into the header of the test case and understand the requirement for which the test case is written
- ② Then go to the body of the test case try to find following mistakes
 - a) Missing scenarios
 - b) Repeated scenarios
 - c) Wrong scenarios
- ③ we will check whether the ~~yourself~~ scenarios are organized or not and your test case should take less time for execution.
- ④ we will check whether the test cases are simple to understand. So that when given to new Fox he or she should be able to execute it without asking question
- ⑤ Look into headers of test case & try to find
 - 1) Check all attributes are there or not
 - 2) Check whether the content in all attributes are correct or not.
- ⑥ Check whether the test case format or template is correct or it should be according to the standard given in the Project.

Tc Review Tracker



Sl.no	Test case name	Step no	Reviewing Commands	Severity	A.H.
1	BO-AI_all scenarios. x ls.	PreCondition	not getting Condition	major fixed	critical fixed
2	BO-AI_all scenarios. x ls	5	Transfer more than balance Scenario is missing	critical not fixed	not fixed

Defect Tracking

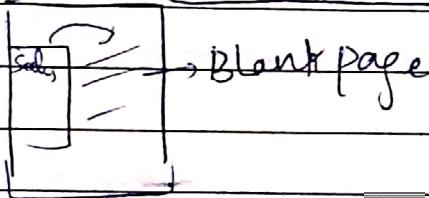
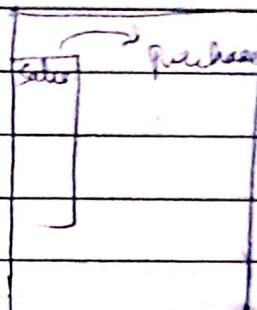
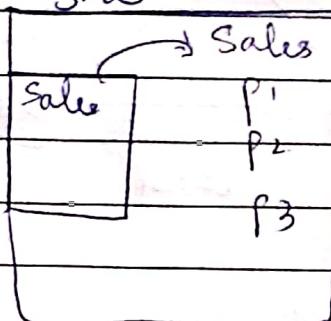


Defect: Any feature that is deviating from (not working) customer requirements is called Defect (or) Deviation from requirement Specification.

Req.

S/w.

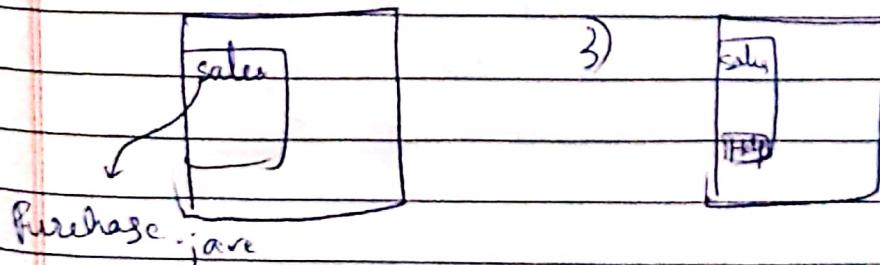
click on Saleslink
should
sales page display
Products P1-P3
Should display



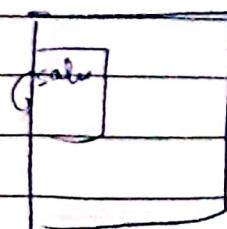
Defect may occur due to the following 3 reasons:

- 1) Wrong Implementation
- 2) Missing Implementation
- 3) Extra Implementation

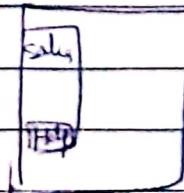
1) wrong Implementation



2)



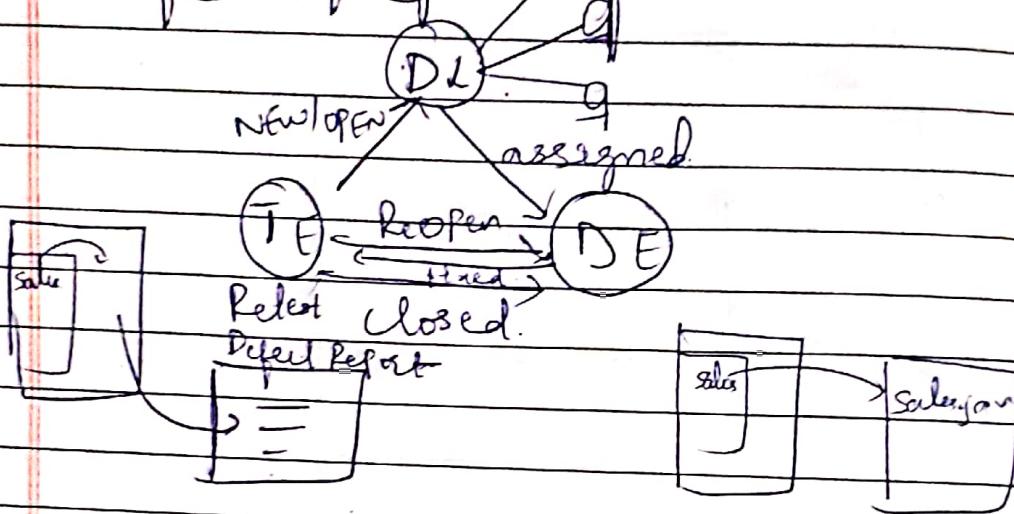
3)



Defect / failure / error

Bug : Informal name given to defect is called Bug.

Defect life cycle



Test Engineer :

- ① Test Engineer finds the defect
- ② He will prepare defect Report
- ③ puts the status as NEW/OPEN.
- ④ Sends report to Development lead

Development lead :

- ① Development lead reads the report and understand the problem.
- ② Identifies developer who did the mistake
- ③ change the status to assigned.
- ④ Sends it to development engineer

Development Engineer

- ① Development Eng reads the report and understand the problem
- ② Comes with source code and fix the bug
- ③ change the status to fixed.
- ④ Send the report to Test Engineers and



cc. to Development lead

Test Engineer

- ① He will read the report and understand the problem.
- ② Re-test the bug if the bug is fixed he will change the status to Closed.
- ③ Otherwise change the status to Reopen and send report to development lead and Development Engineer.

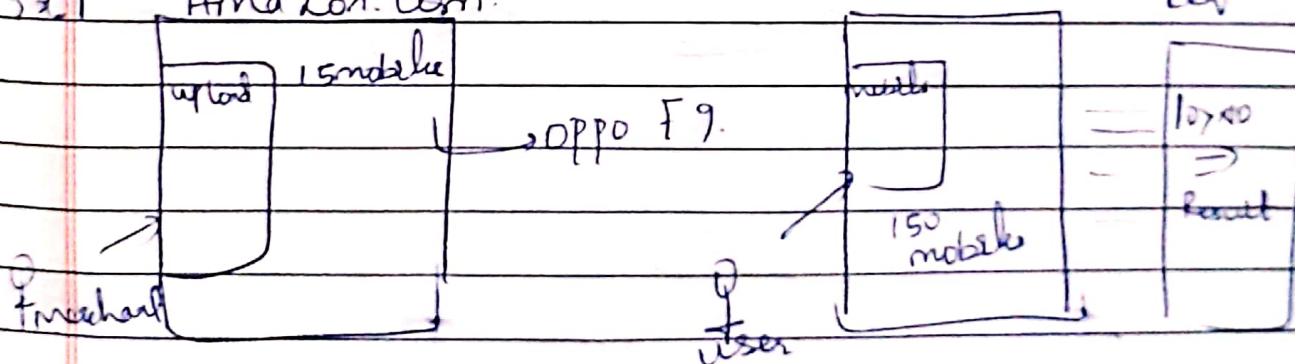
Q. What is Reject status?

Test Engineer finds the defect and sends it to Developers and developers say that it is a feature and change the status to Reject.

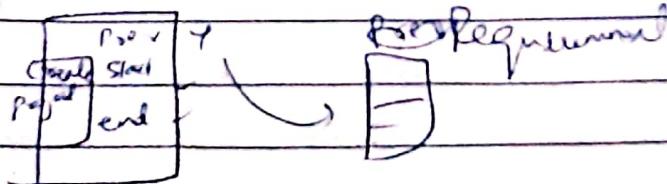
Reasons for getting Reject Status:

- ① Because of misunderstanding the requirement.

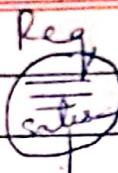
Sx:1 Amazon.com



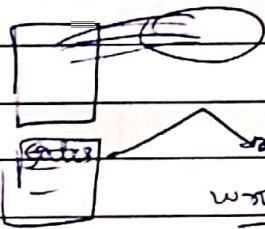
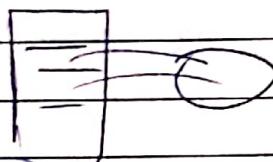
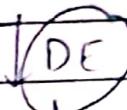
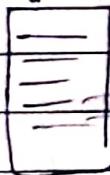
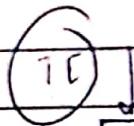
Sx:2 Adeltime



- ② Because of referring old requirements.



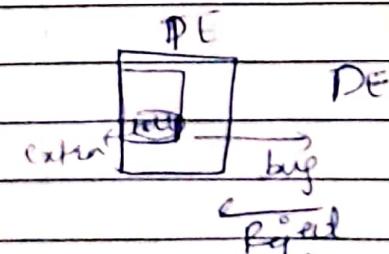
later introduced new one.



wrong

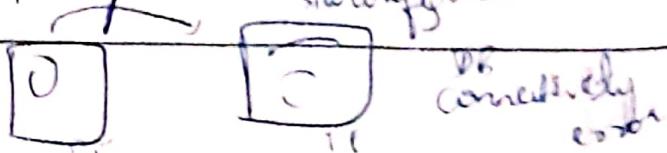


- ③ when you find a extra feature as a bug & send it to developers, developers might reject it. In such a case, test engineer should Reopen it and ask developer to fix it



Reopen. and say update requirement

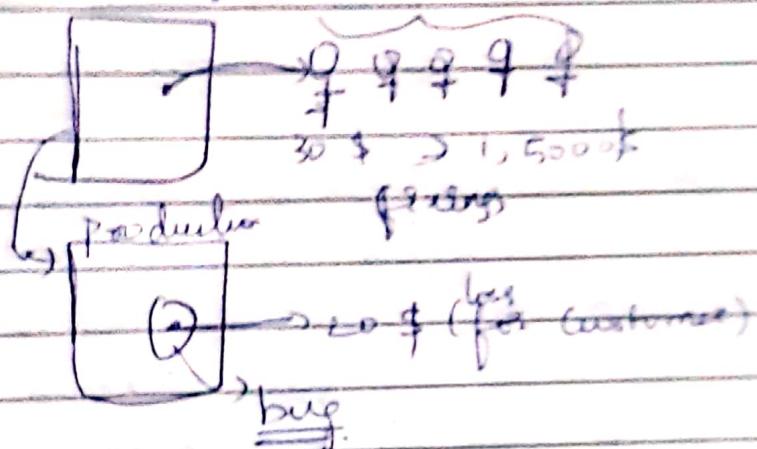
- ④ Because of wrong installation or configuration of Software, there might be a defect. when you send it, developers reject it saying that you have not installed properly.



④ What is Cannot be fixed States?

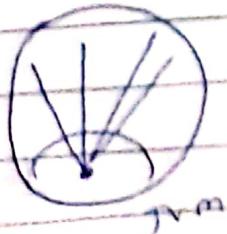
Developers are accepting that it is a defect but they say that they are not in a position of fixing it because of few reasons:

① Cost of fixing the bug is more than cost of bug it does in the business because of having bug in the Software.



② If there is a bug present in the root of the product and if it is minor, fixing that defect might have lot of impact on other features.

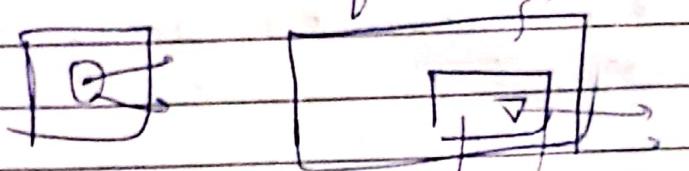
③ If technology itself is not supporting to fix the bug and if that bug is minor or major, they might reject the bug.



(bug) ~~can't fix~~
critical feature → should be fixed



If bug is major or minor, they can say cannot be fixed but if it is critical, should come up with alternative sol'n to develop the same feature.



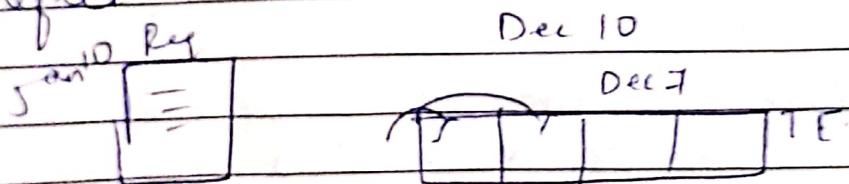
(they said like ~~not~~ instead of this
In Potnet it will keep)

Sometimes customer say use another technology

Q) what is postpone or Deferred Stacks?

→ Developers are accepting that it is defect but they want to fix it little later because of few reasons

i) There is a minor/major bug developer are not having enough time to fix it In such a case, they might postpone the defect

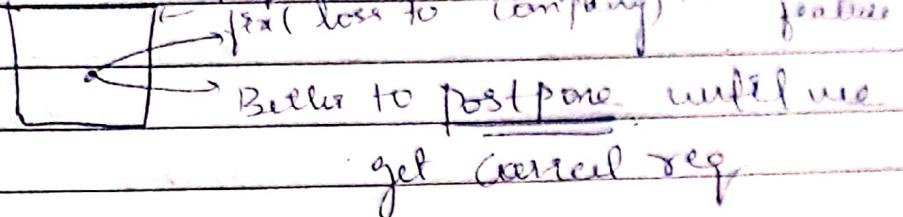


0-Blocked
0-critical
50-major
100-minor

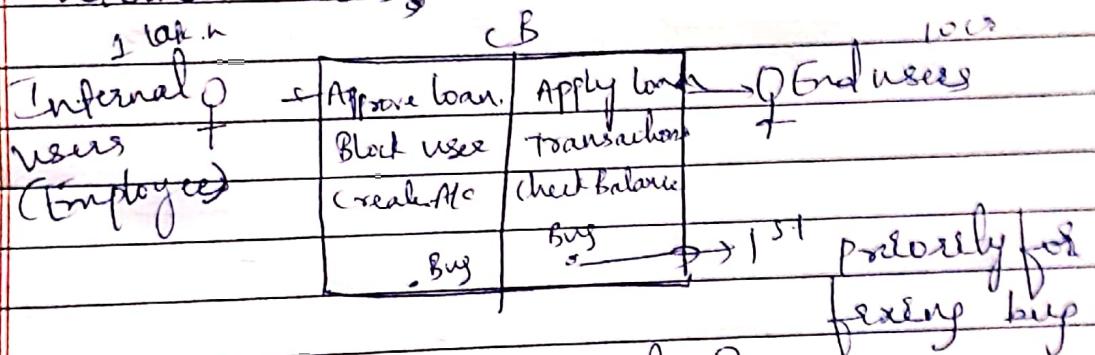
50-Blocked
30-critical
50-major
100-minor

Q) If sends to DFT that there is a bug in my def that from DFT send to TE as postponed status

(ii) When you send bug to developer they say that customer is expecting lot of changes in the same feature. So better postpone the bug until we get a clarity of what they are changing.



(iii) If there is a minor bug exposed to internal users,



Q. What is Duplicate status?

→ If you send defect which is already sent by some one else then developers say duplicate.

Reasons:

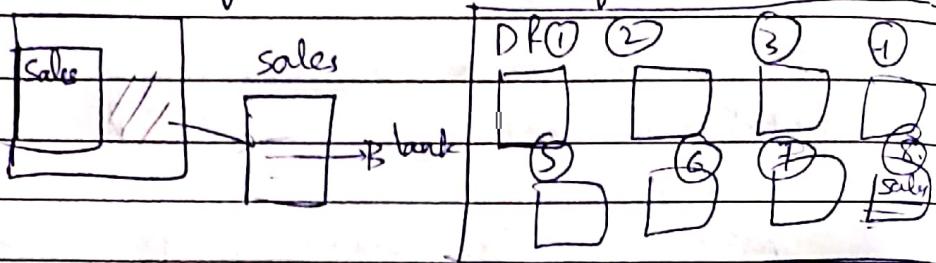
- 1) Because of common features.
- 2) Some one else might come and catch defect in your module and send it to developers. If you find same bug and send it to developers they say duplicate.
- 3) Old test Engineers have found lots of bugs and send them to developers and some old bugs are still pending. If you open same old project and send same old bug, they change

Status to duplicate

Q How to avoid duplicate Bugs?

- Before we prepare the defect Report & logged over the bug, we should make sure that it is not a duplicate or You should search for duplicate bugs in the tool by entering certain keywords. [In this ex sales is a keyword].

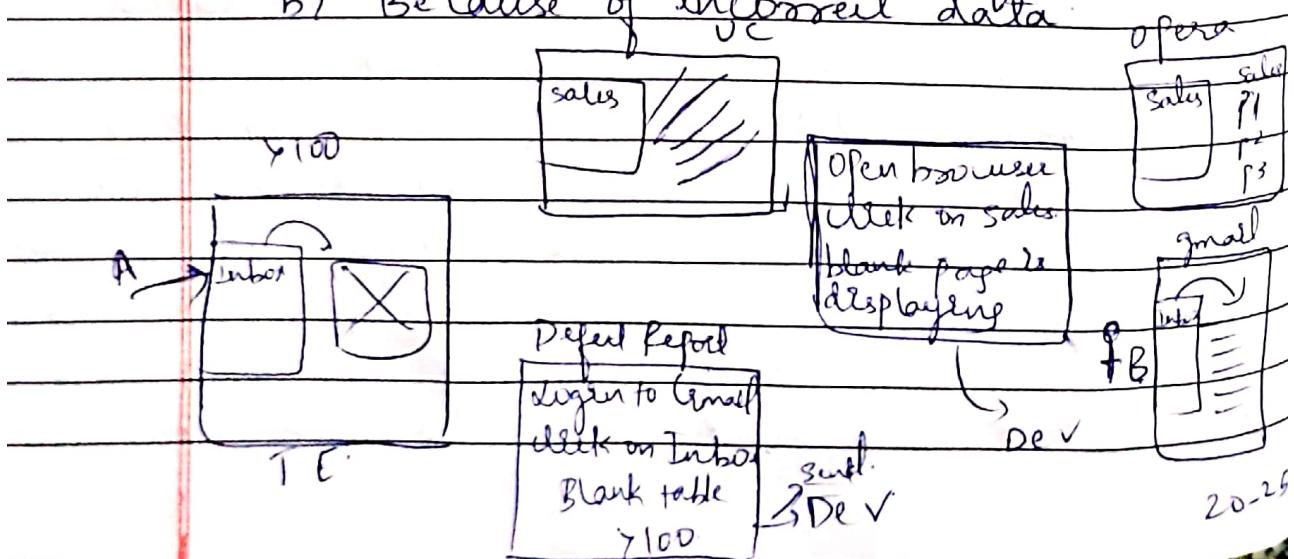
Defect tracking tool



Q What is Not Reproducible?

- Test Engineer will be able to see defect but Developers will be not be able to see the defect. In such a case, they will say that it is not reproducible.
- Reasons:

- Because of improper defect report;
- Because of incorrect platform
- Because of incorrect data



Q E D

Inconsistency Bug

Because of Inconsistency Bug, Some time it appears sometimes it disappears. That is called as Inconsistency Bug.

RFE/CR

It is a problem which is not a part of Requirement. When it is called as Request for Enhancement / Change Request.

RELEASE NAME: Defect ID :- D-001

BUILD NAME: Build ID: -b-01

STATUS: ~ NEW/OPEN ASSIGNED, FIXED
Closed/Reopen

SEVERITY: ~ Blocker, Critical, Major, Minor

PRIORITY: ~ HIGH, MEDIUM, LOW

TEST DATA: ~ UN-abc, pub=123

TEST ENVIRONMENT: ~ w7, ws, w10

TEST CASE NAME: ~ Email - s1 mailer

FOUND BY: ~ UR NAME

DETAILED DESCRIPTION: ~ ST is not displaying in ST page.

BRIEF DESCRIPTION:

Following are the procedure to reproduce the bug:

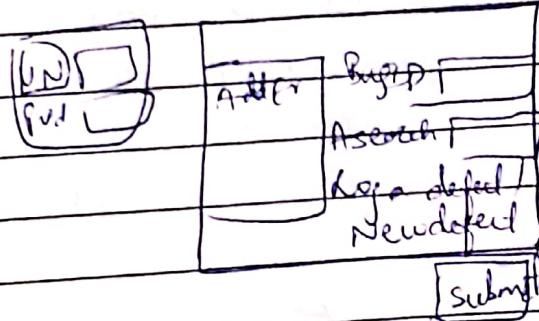
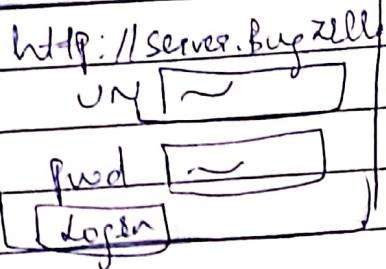
- a) login to Gmail
- b) click on Compose link
- c) Enter all valid data and click send button
- d) Click on ST link

ER (Expected Result): ST should display s1.aspx

AR (Actual Result): ST not displaying in ST

→ screenshot (optional) page

- 1) Find the defect
- 2) X duplicate
- 3) Prepare Defect Report
- 4) Send to Dd
- 5) Manage defect life cycle



* unique ID (Bug ID)

* Stored in DB

* Send it to Dd

* It should get updated → Dd