

12/9/19

SDLC

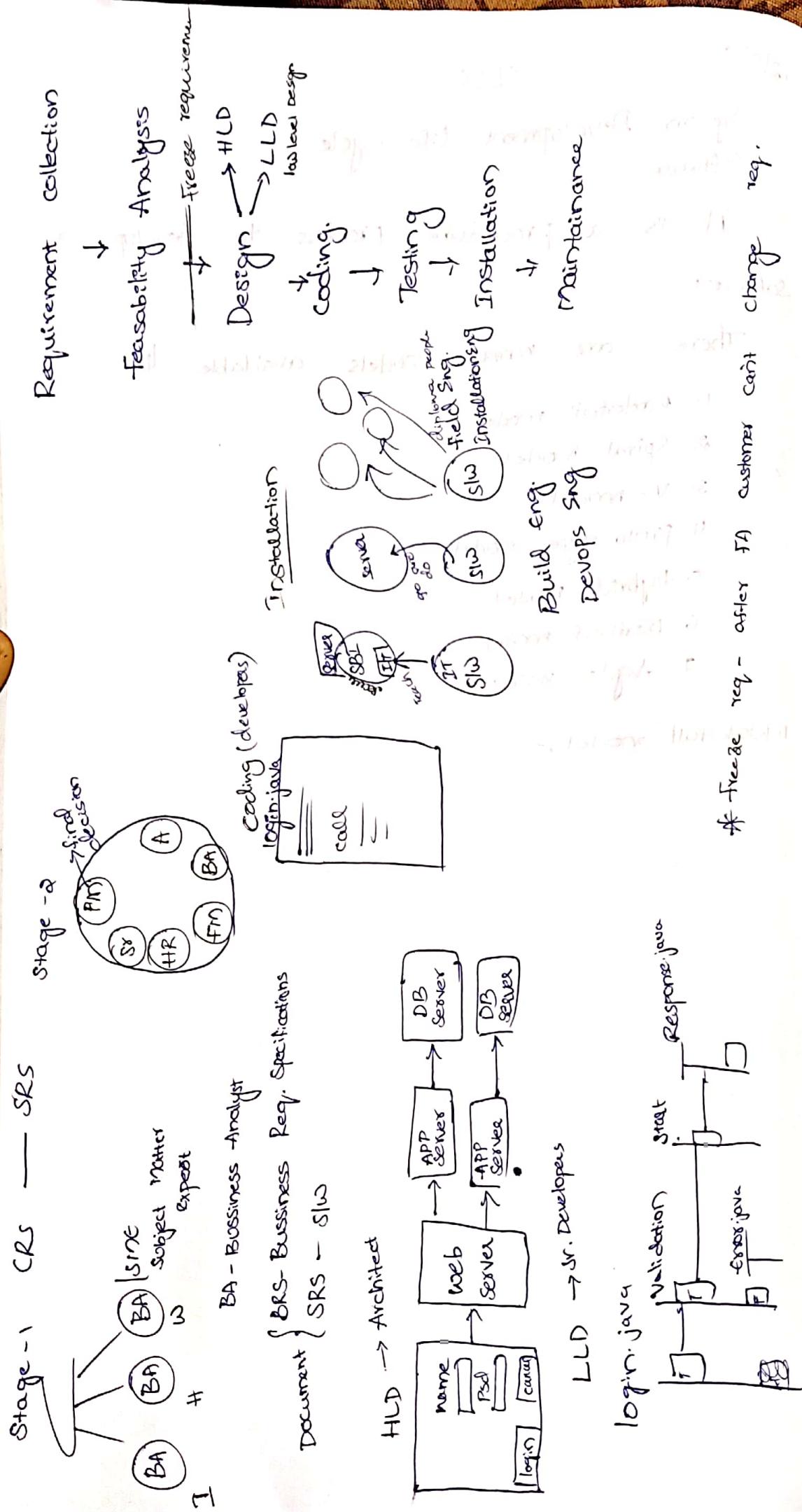
System Development Life cycle Software

It is a procedure / process to develop a software.

There are many models available like

1. Waterfall model
2. Spiral model
3. V - model
4. proto type model
5. Hybrid model
6. Devolved model
7. Agile model

1. Waterfall model :-



Advantages:

1. Initial investment is less
2. Simple to understand
3. Easy to adapt

Disadvantages:

1. Testing is done only after coding.
2. Rework is more.
3. Total investment is more
4. Developers only involved in testing.

Applications :-

1. When we are going for a single project
2. short-term

~~Now it is not in use.~~

Why Developers should not be involved in testing

1. They never see the product from -ve point of view.
2. They can't see the build broken. (slw)
3. Even after finding the bugs they might hide the bugs.
4. They will spend more time on development than on testing.

Why to freeze requirements?

1. changes can allow to some more bugs.

Software Testing:-

Verifying the functionality of an application against requirement specification is called Software Testing.

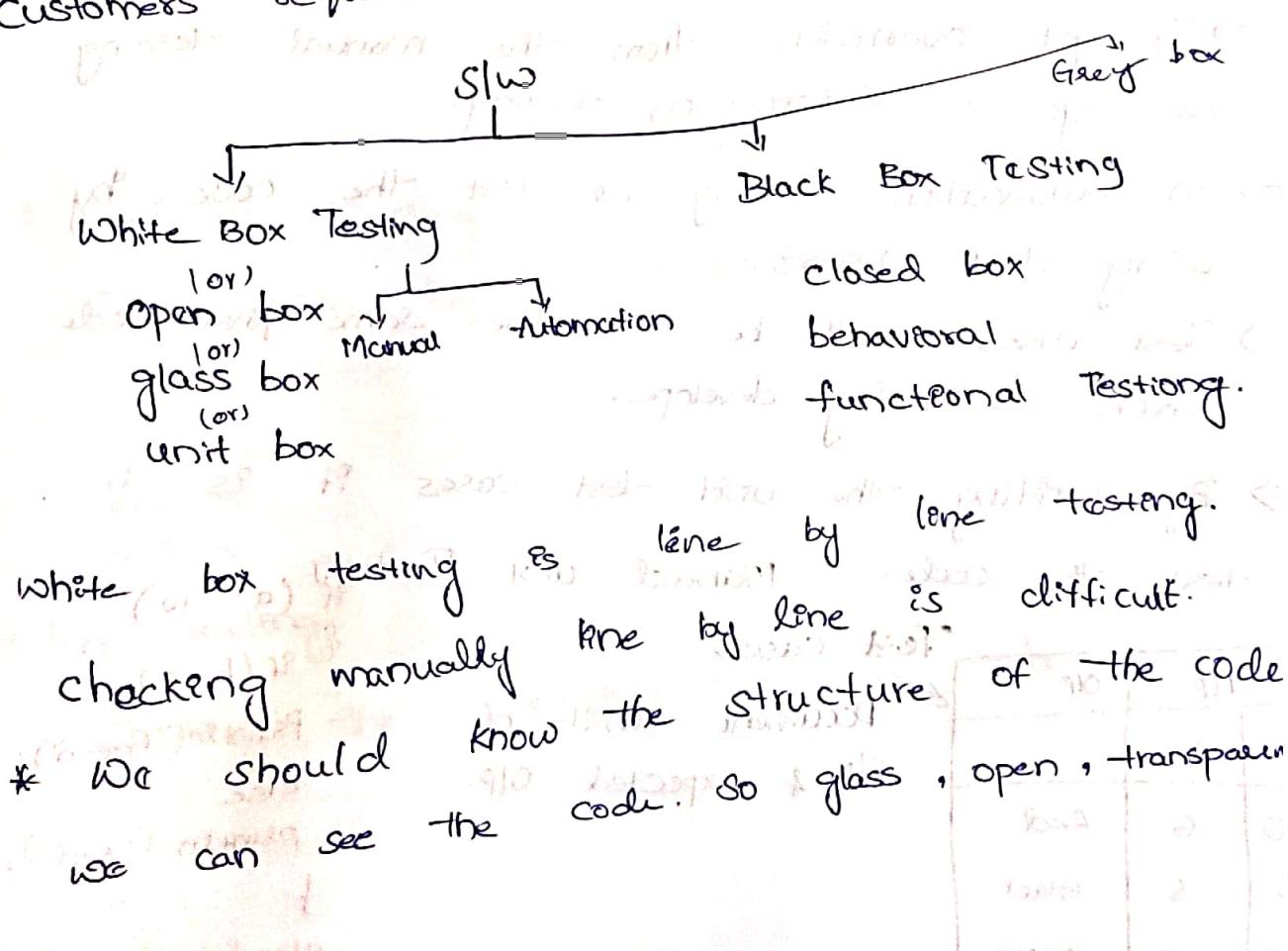
The process of finding defects in the software.

Execution of the programs with an intention to find bugs.

Why?

1. Every software is build to support a business if there are any problems in the software it will effect business. So, before you launch it to market recognize all problems and fix them.
2. To improve the quality of the product.
3. To make sure software works according to customers requirements.

Customers requirements.



Black box Testing:
 Testing the behavioral or functionalities of an application is known as Black box testing.
 We can't see the code & we can see the application.

White box

- ⇒ Testing each and every line.
- ⇒ We can see the code.
- ⇒ We should know structure.
- ⇒ We should have knowledge on coding.
- ⇒ Developers will test it.

Grey box :-

- ⇒ Combination of both white box & Black box.
- ⇒ We should know developing & testing.
- ⇒ Also one uses.
- ⇒ To get overridden from the manual testing we go to Automation testing.
- ⇒ In Automation testing we test the code by using test cases.
- ⇒ Test cases will be in the same java code which is being developed.
- ⇒ By writing the unit test cases it is to test the code. Manual unit Testing if ($a == 10$)

I/P	O/P	Test Cases
a	b	Document consists of I/P & expected O/P.
10	6	Red
8	5	Black
10	10	Green

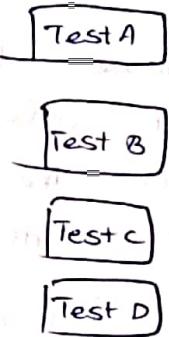
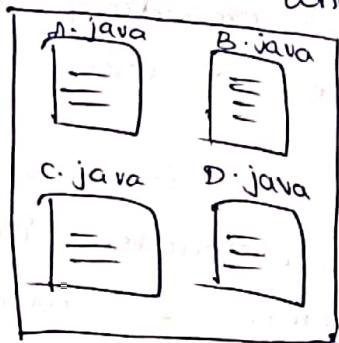
```

        {
            if (b > 8)
                println("Green");
            else
                println ("Red");
        }
        else {
            println ("Black");
        }
    }
}

```

Req.

- [A]
- [B]
- [C]
- [D]

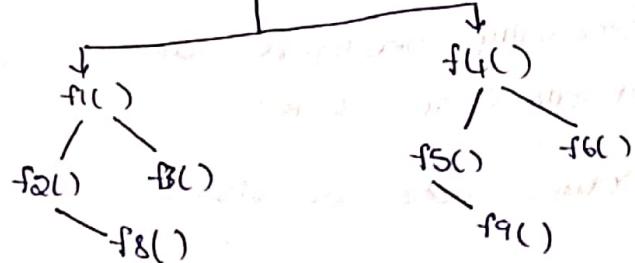


Types of white box testing:

1. path testing
2. condition testing
3. loop testing
4. white box testing
5. W.B.T performance

1. path Testing:

Testing the path of the independent functions.



2. condition Testing:

Testing the logical condition if is true or false

if
if else
else if
switch

3. loop Testing:

Testing the repetition of the code.

4. Memory point of view: If exceeding memory

- should decrease the lines of code
- Efficiency of code
- should check the code if it is used or not same line.
- for ex initialization & declaration in writing a method & should use that method.
- method by calling that method.
- using inbuilt functions.

5. Performance point of view:

- Speed
- Should use inbuilt functions.
- should decrease lines of code.

Difference

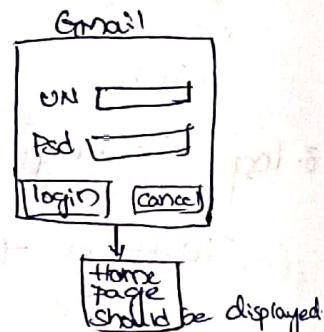
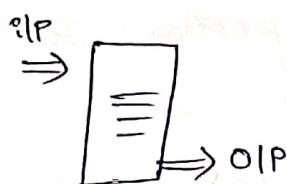
blw W.B.T

& B.B.T

B.B.T

W.B.T

1. Testing each & every line of a code.
 2. Generally developers are involved in W.B.T
 3. Source code is visible.
 4. We should know the programming language.
 5. We should know the internal structure of the code.
1. Testing the functionality (by behaviour of application).
 2. Test Engineers are involved here
 3. We need not know programming language
 4. Source code is not visible.
 5. We need not know internal structure of the code.



No one does Greybox testing.
desprog + app

SDTE

Black Box Testing :-

Testing the behaviour or functionalities of an application is called BBT.

Types of BBT :

- functionality Testing
- 2. Integration Testing
- 3. System Testing
- 4. Acceptance Testing
- 5. Ad hoc Testing
- 6. Smoke Testing
- 7. Compatibility Testing
- 8. Usability Testing
- 9. Regression Testing.
- 10. Exploratory Testing
- 11. performance Testing
- 12. Accessibility Testing
- 13. Globalization Testing
- 14. Recovery testing
- 15. Reliability Testing

(I18N)

Internationalization

→ Initialization Testing

Localization Testing

(L10N)

1. Functionality Testing :- Checking each & every component of application through based on requirements.

CBO-SRS

3.0 - Amount Transfer page

3.0.1 TAN Text field

3.0.1.1 should accept 10 digit valid Alc number which are created by manager

3.0.2 TAN Text field

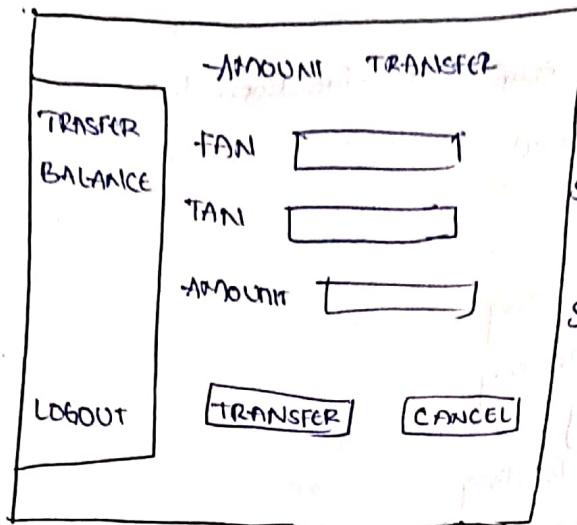
3.0.2.1 on Page

3.0.2.2

3.0.3 Amount Text field. It's content of atleast 100-5000 integer numbers.

3.0.3.1 should accept the amount to balance.

3.0.3.2 shouldn't accept



TAN, FAN, Amount are called components.

Step-1: 1st +ve testing based.

Step-2: -ve testing.

defect :- When it is not working according to the requirements.

Over testing: Testing the scenario of same set in diff. ways

Under testing: Testing the app with less scenarios

Optimize testing: Testing the app utmost req. set of scenarios which make sense.

+ve testing:

-ve testing:

We can do functionality testing in diff. ways

1. Over testing: Testing the app on functionality with same set of scenarios in diff. ways

2. Here we are going to loose lot of bugs because we spend lot of time in testing the same scenarios.

2. Under testing: Testing the app with min. set of scenarios

3. Optimized testing: Testing the app only with those scenarios which makes sense.

* Here we are going to miss less no. of bugs

4. +ve testing: Testing the app with expected or valid data.

5. -ve testing: Testing the app with unexpected data (or) invalid data.

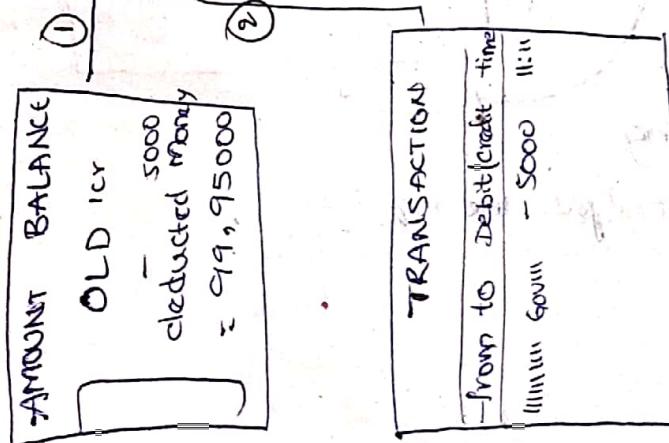
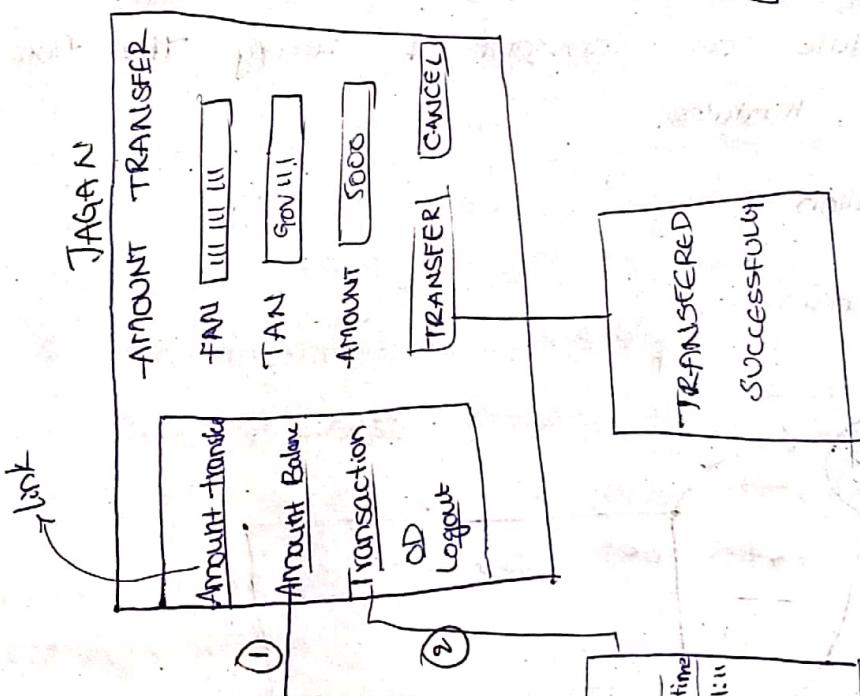
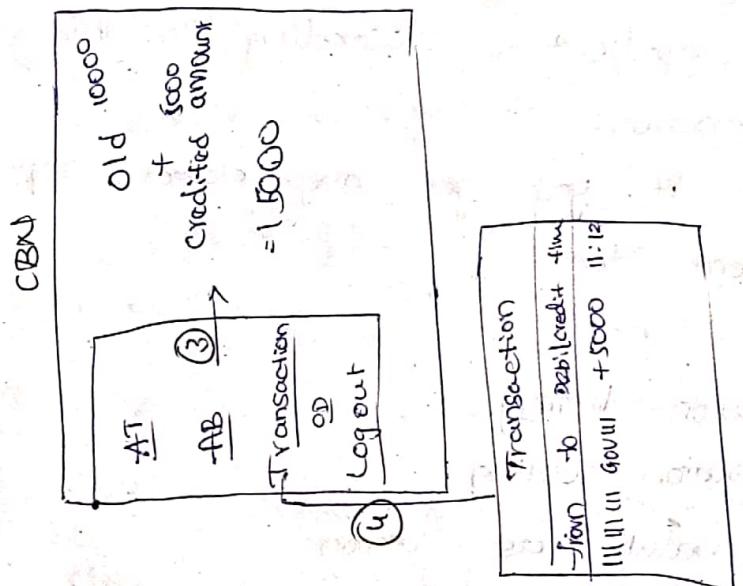
2. Integration Testing :-

Testing the data flow between two module to another module.

(or)

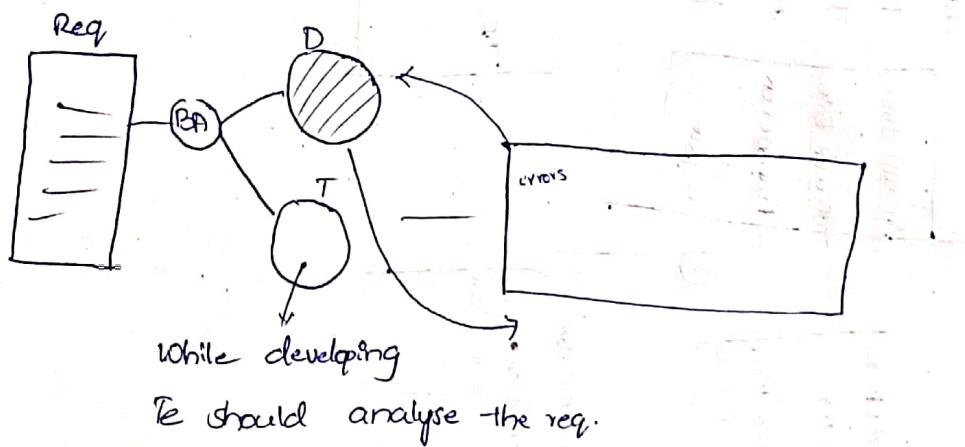
Testing the data flow b/w two application pages.

* modules is nothing but features.



- How to do Integration testing?
1. Understand the application or req thoroughly.
 2. Understand each component how it works and 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 defect report it to development team.

- * Positive Integration testing
- * Negative Integration testing
- * Keep the one module as component
- * Keep one module as constant & verify the flow with all other modules.
 - One way Integration
 - No Integration
 - Two way Integration



Search:-

If we give shirts it should go to shirts.

Valid search data we should give
Sort should apply.

filter should apply to the data.

If the search product is not available

it should show No stock available.

Types of Integration Testing:

1. Incremental Integration

2. Non Incremental (Big Bang)

A Parent

B is the parent of C

B Child

C is the parent of D

C Child

Bottom-up is vice-versa D becomes parent

D Child

Ex: CEO
|
MANAGER
|
TL
|
TE

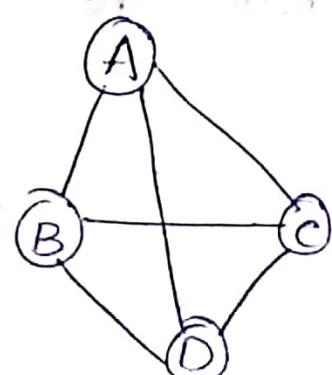
↓ Top-bottom credentials

↑ bottom-up Reporting

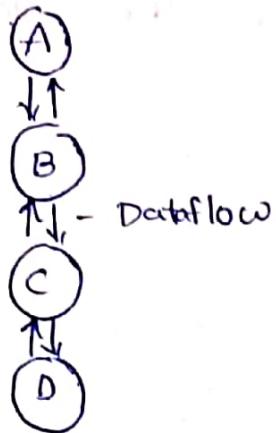
2. Non-incremental Testing:-

This is also known as Big Bang theory.

Here data is in one module
and connected to all the other
components.

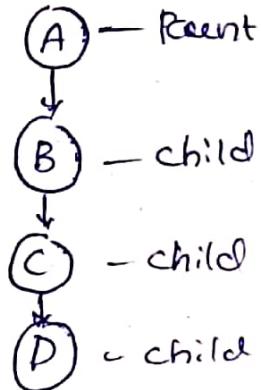


2. Bottom-up In



1. Top-down Incremental Testing:

Incrementally add the modules and test the data flow between the modules.
Make sure that the module that we are adding is child of previous one.

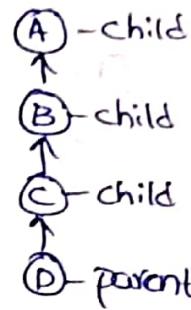


2. Bottom - Up Integration Testing:

Testing starts from last child upto parent.

Incrementally add the modules and test the data flow between the modules.

Make sure that the module you are adding is the parent of the previous one



Non-Incremental Integration Testing:-

We use this method when

- When data flow is very complex
- When it is difficult to identify who is parent and who is child.

This is also called as Big Bang Method.

Here combine all the modules at a shot and start testing the data flow between the modules.

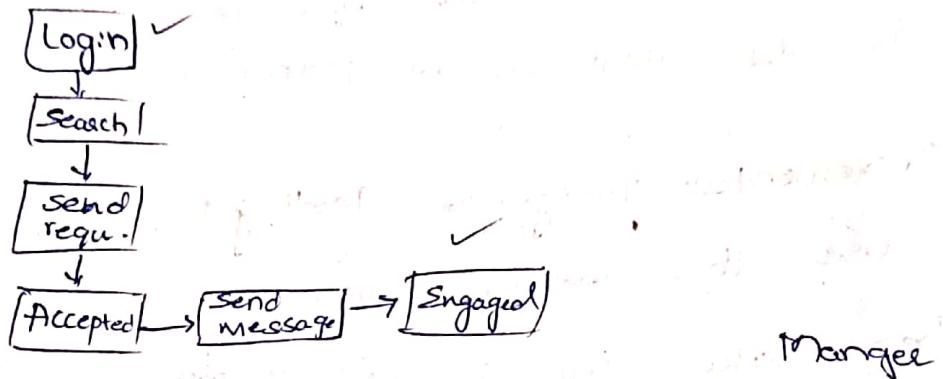
The Disadvantages of Big Bang method is

- We may miss to test some of the modules.
- Root cause analysis (RCA) of the defect is difficult. (Identifying the bug from where it has arisen and we will not get to know the origin of the bug.)

3. System Testing:-

End-to-end testing where in test environment is similar to production environment.

Ex:
Shaadi.com



Manager

Kaveri

$$\textcircled{1} \quad OD \rightarrow 40,000$$

800 - interest
 $\frac{800}{200} \rightarrow \text{Activation fee}$
 $40,000 - \frac{800}{200} = 41,000$

Apply OD	40000
Balance	+ 0
Repay	+ 0

Pending Deposit
(APPROVE)

$$\textcircled{2} \quad OD \rightarrow 40000$$

800
 $\frac{800}{0} = 40800$

- ① Apply OD
- ② First it will be 40,000 in OD bal.
- ③ Manager should accept

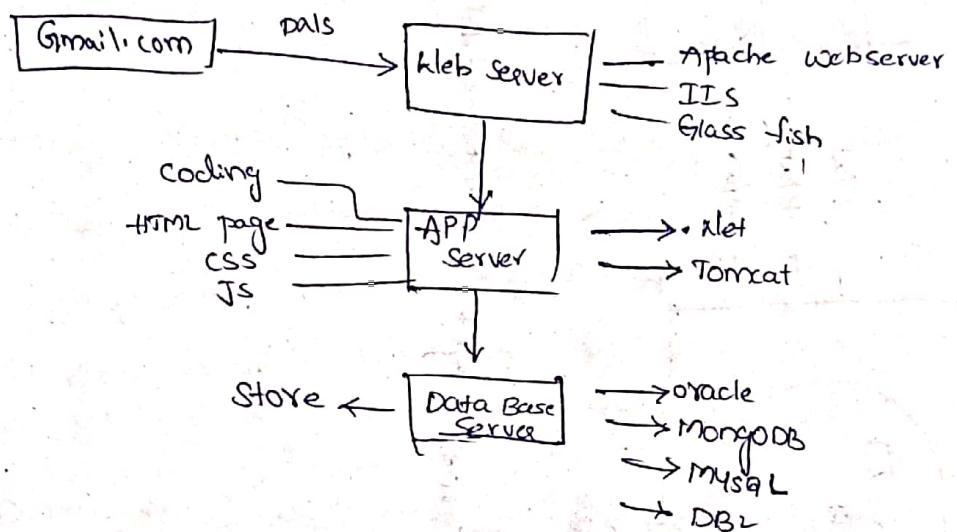
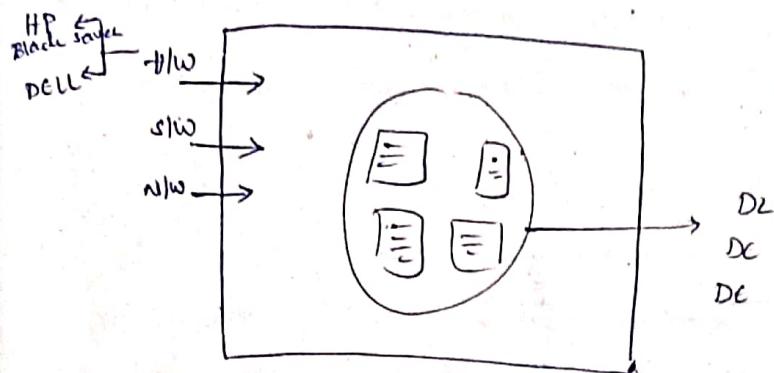
$$\textcircled{3} \quad OD \rightarrow 50000$$

500
 $\frac{500}{0} = 50000$

- ④ Alex change the service date
- ⑤ Check the bal. $\rightarrow 41,000$
- ⑥ Her sal is 20,000 so go to her account deposit 21,000 to her

- ⑦ Go to repay OD page & Pay.

Development Environment:



It is nothing but a setup which is used to develop the software. It consists of hardware; software and network.

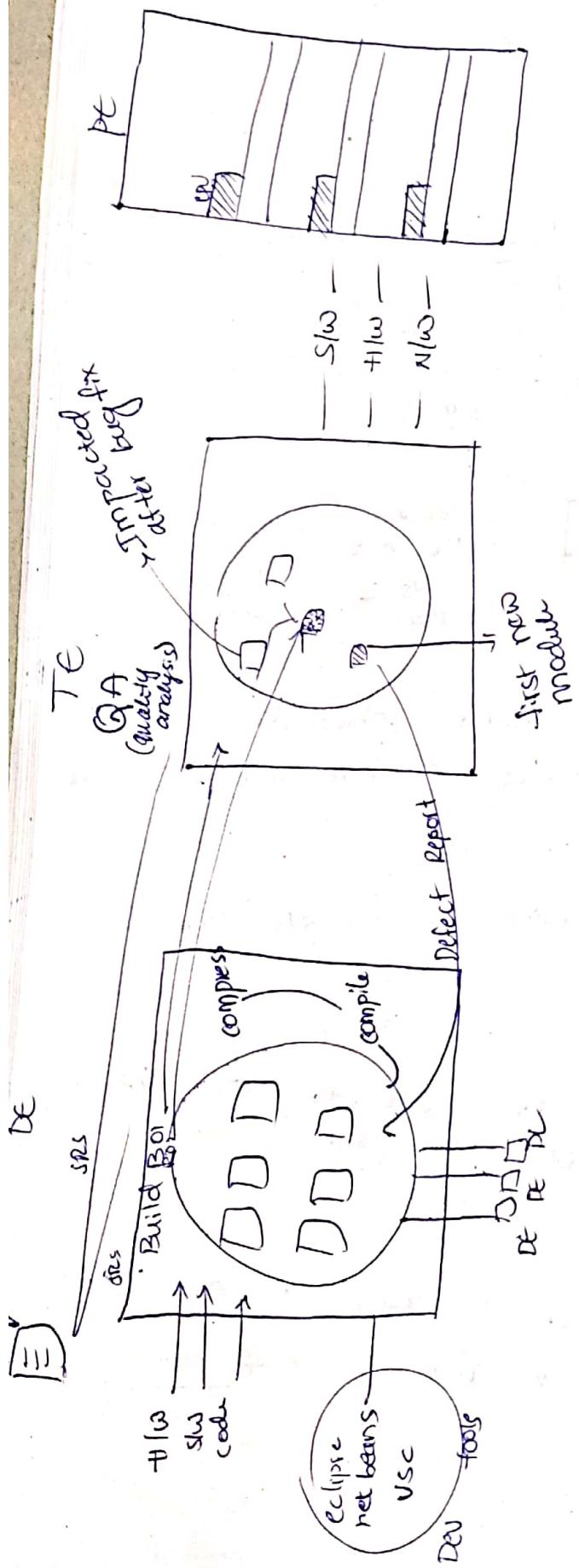
DNS : Domain name service — converts the name to IP address.

Testing Environment:

It is a set up which is used to test the application which consists of H/W, S/W & N/W.

Production Environment:

It is a set up which is used to run the application which consists of H/W, S/W & N/W.



If we use diff hardwares we get compatibility problem

$\Rightarrow TL, TE \quad \} \text{ can install application}$
 $\Rightarrow DE, DL \quad \} \text{ in the QA(TE)}$
 $\Rightarrow \text{Build E}$

* Testing Environment should be same as the development Environment means.

1. Hardware should be similar to production.
 - a) The make (manufactured by) similar to production Server (For example if the production Server is t1P then the testing Server should be hP).
 - b) Configuration and make should be similar but different capacities (i.e no of CPU).
2. The software should be similar to the production
 - a) The O.S. should be similar.
 - b) Application server should be similar.
 - c) Web Server should be similar.
 - d) DB server should be similar.
3. 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 the production environment.
 - c) In real environment we ^{may} have lakhs of entries into Database. But, while testing we can't enter manually lakhs of entries. So, we are going to write test script program which generates 1000's of users (or) data and this can be used for testing. Thus it is for testing.

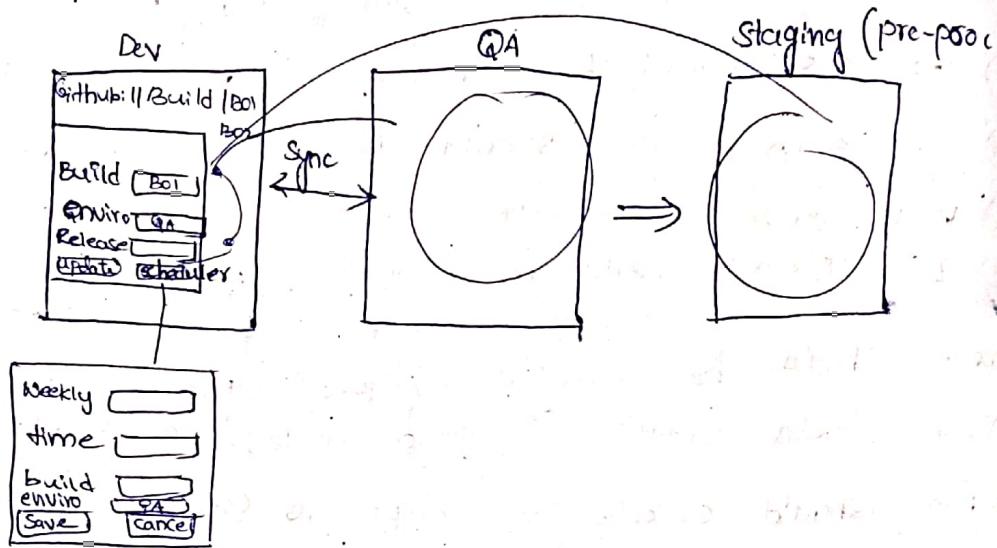
In TE who is involved in installing SW

1. Anybody from testing team.
2. Anybody from development team.
3. Build / Release Engineers

What is Build?

It is a piece of software which is compiled, compressed and installed in all environments.

Devops:-



Jenkins tool: Jenkins tool which is used to automate the build process.

This tool will take care all the operations build, deploy.

This is also known as Continuous Integration.

Continuous Integration:- Keeping QA and Dev build in sync.

We go to devops Eng for the complex things.

Even → the jenkin tool reports → the errors
in → the code while build. (syntactical)

We can also build in the production through jenkin tool but it should deploy the latest build.

patch:-

When → there is a bug it will be reported to the DE then after fixing that minor bug or changes he will compile and compress and the build should be updated.

(We won't build the new build)

If we have new module which is not impacting the other modules no need build new build.

Jenkin:-

It is a continuous integration tool which is used to automate the build process.

Devops:-

Devops is nothing but development operation which involves activities like Development, managing code, build a new build, install & uninstall the application in all the environments is called Devops (development, Q environment).

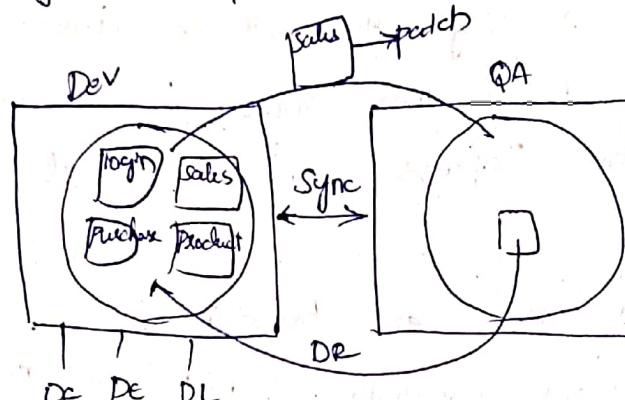
Continuous Integration:-

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

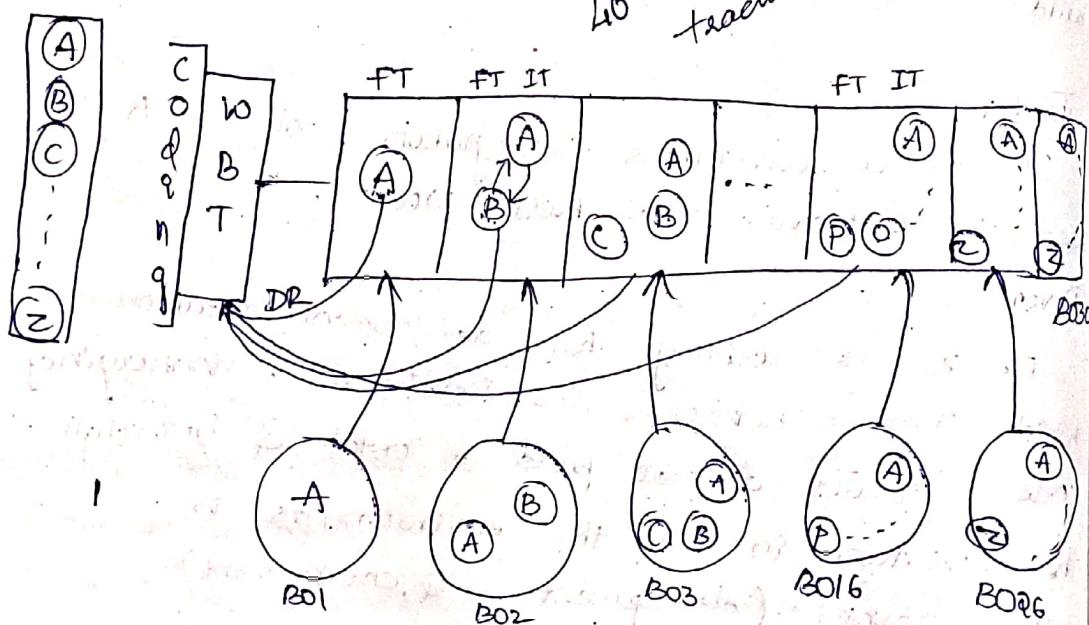
patch:-

patch is nothing but it is a modified program or deleted prgm or added prgm which will be done in less time.

While patch is given to QA we never going to uninstall the old build and just we are going to update the patch.



What is Test cycle?



Time Spent on testing a build or software completely is called test cycle.

Defects:

Block - 0

Critical - 0

Major - 4 - 10

Minor - 20 - 40

Stand alone

calc
Note pad
Auto calc

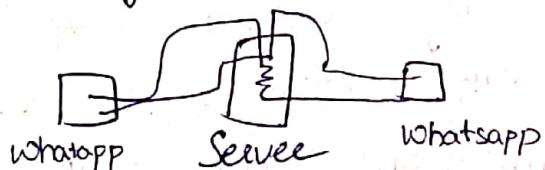
client server:

client app
Server app

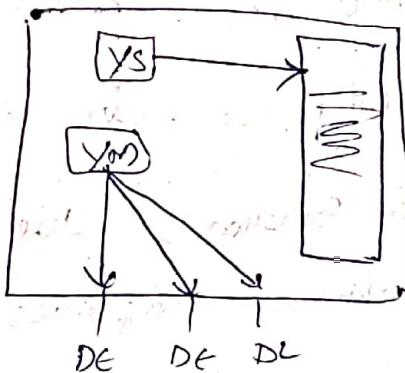
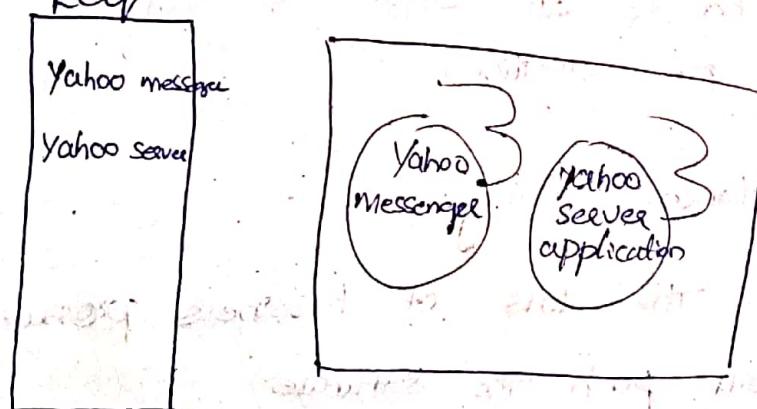
one to one

Yahoo messenger whatapp

Two type Architecture



Req:

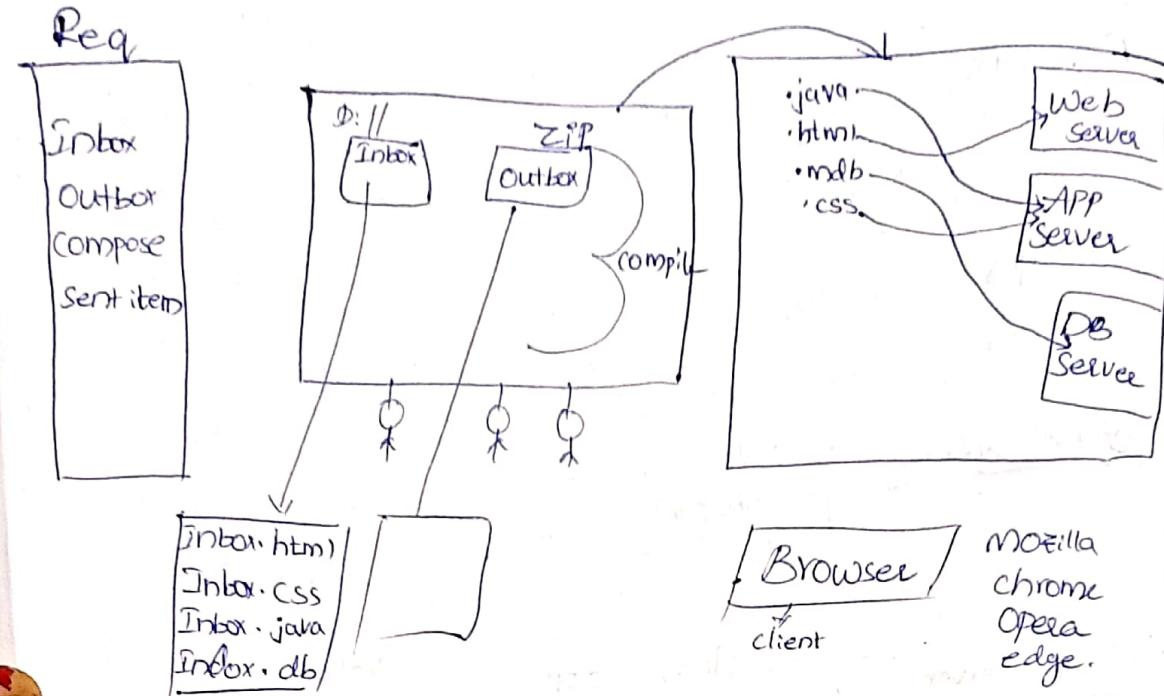


Web application:-

n - type architecture.

User can access server by the browser.

GNOME server application.



16/9/19

Testing the software from customer point of view.

4. Acceptance Testing :- (end to end)

Acceptance testing is done by customer here. They use the software 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.

Why we do Acceptance Testing:-

1. Sometimes due to the lots of business pressure software companies will push the software.

So to find out the defects customer will do acceptance testing.

2. Developers misunderstands the requirement and develops the wrong req. Then this customer will do acceptance testing.

3. During Knit Testing test engineers will miss some of the critical bugs due to negligence or misunderstanding of requirements to find out those mistakes customer

will do testing.

Difference between Wipro TE & Fedex TE?

1. Wipro testing team do functionality testing, integration testing, system testing.

But fedex team will do end to end testing.

CR - Change Request

RFE - Request Feature Enhancement

Four Approaches for Acceptance Testing:-

1. IT team in the fedex
2. Testing team will do in the business point of view.
3. Testing team will go to fedex and they will do
4. Agents (end users)
- They know different business scenarios.

Agile

- overcomes the drawbacks in traditional model.
 - * long duration
 - * More cycles
 - * Release at end

In Agile

- * Split requirements
- * Last & release to the production on monthly basis.
- * DE & TE considered as a whole team Scrum team.

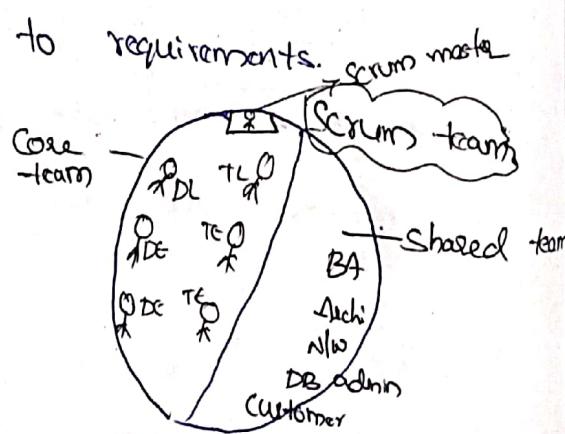
What is Agile?

Agile is a model, where we develop the and test the software in incremental and iterative way.

They came up with this model in order to overcome the drawbacks which were there in traditional model.

- Here they build the model in shorter cycle (4-5)
- Agile works in monthly basis.

Stories are similar to requirements.



product Backlog:- It is used to prioritize the stories.
Customer will prioritize the stories. (Wt bussiness)
Scrum: The process involved in the cycle for developing & testing the application.

Two types:- Core team
Shared team

Task + Story = Sprint Backlog.

Scrum master: Manage or Bridge between Shared team & Core team.

Day 0: planning (Sprint planning meeting)

Daily stand up meeting: 15 min.

Retrospect Meeting: More than 1 hr
mistakes & achievements

use: Doesn't repeat the old mistakes.

product Backlog: It is a prioritized list of stories or features or requirements. Here stories need not to be detailed.

2. The product Backlog is owned and managed by product owner.

Sprint Backlog:- It is a list of stories and the associated task that must be completed within a sprint. It is called as Sprint Backlog. It is a list of stories and task committed by Scrum team to be delivered in one sprint.

Sprint planning meeting: Here entire Scrum team meets together and pull the stories from the product Backlog.

- ① Whatever they can build within a sprint
- ② They assign the stories to the engineers.
- ③ The Engineers derives the task to be performed to complete or to build complete story.
- ④ They estimate the time taken to complete each story.

Scrum master: Scrum master drives the Sprint planning meeting. His prime role is to facilitate complete meeting and coordinate between the Stake holders (customers).

- 2. In this meeting product owner clarifies, if any questions or queries are there w.r.t stories.
- 3. In this meeting develop engineer should write the task for building a story.
- 4. Development Engineer prioritize the stories and test
- 5. In this meeting Test Engineers should derive the task for testing the feature build for a story.

(^{the}) Retrospect Meeting: Entire scrum team meets and discuss about achievements (good processes followed) and mistakes (wrong processes followed) and we document it and that document is called as Retrospect document.

a. When next Sprint starts while doing Sprint planning we refer this document and we plan it in a such a way that old mistakes are ^{not} repeated and good activities once again adopted.

Daily stand up meeting (10-15 min) Scrum master drives this meeting.

- ② Scrum master facilitates the meeting.
- ③ Discuss daily, yesterday's and upcoming task whether it is completed, in progress or incomplete.
- ④ Discuss about any impediments (doubt) in the meeting if it is solvable then & there you solve it or else take a note that issues and solve it later.
- ⑤ Each engineer should tell @ what he has done yesterday
 - ⑥ what were the impediments faced
 - ⑦ what you are planning to do today.
 - ⑧ what are expected impediments.

- ⑨ Generally this meeting should go for max. 10-15 min
- ⑩ Generally this meeting is done in the beginning of the day.

Scrum team: Is a group of members or individuals working together to deliver or completely committed stories.

- ⑪ Generally scrum team size will be 7-9 people.
- ⑫ Generally there are two kinds of teams
 - ⑬ core team
 - ⑭ share team

Core team: Core team involves Scrum master, developers, test engineers.

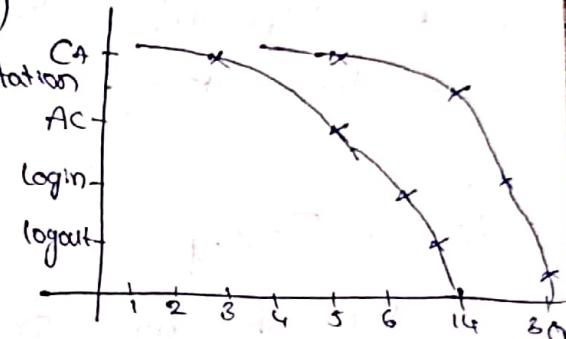
Shared team: Shared team involves BA, architect, database / network administrator, UI / UX designers, product owners.

Who leads the Scrum team?

Scrum master leads the complete Scrum team and he facilitate everyone to complete the task.

Burn down chart:- (Productivity.)

It is a graphical representation of work left vs time.



White board:-

Kanban board

Story board

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

Start	In progress	Completed
1 st day Create Alc	Create Alc	
2 nd day Create Alc	Create Alc	
3 rd day Create Alc	Create Alc	
4 th day Create Alc	Create Alc	
5 th day Create Alc	Create Alc	
6 th day Create Alc	Create Alc	
7 th day Create Alc	Create Alc	
8 th day Create Alc	Create Alc	
9 th day Create Alc	Create Alc	
10 th day Create Alc	Create Alc	
11 th day Create Alc	Create Alc	
12 th day Create Alc	Create Alc	
13 th day Create Alc	Create Alc	
14 th day Create Alc	Create Alc	
15 th day Create Alc	Create Alc	
16 th day Create Alc	Create Alc	
17 th day Create Alc	Create Alc	
18 th day Create Alc	Create Alc	
19 th day Create Alc	Create Alc	
20 th day Create Alc	Create Alc	
21 st day Create Alc	Create Alc	
22 nd day Create Alc	Create Alc	
23 rd day Create Alc	Create Alc	
24 th day Create Alc	Create Alc	
25 th day Create Alc	Create Alc	
26 th day Create Alc	Create Alc	
27 th day Create Alc	Create Alc	
28 th day Create Alc	Create Alc	
29 th day Create Alc	Create Alc	
30 th day Create Alc	Create Alc	

Sprint review meeting:- (Last stage of Agile)

1. Here entire scrum team meets at the end of Sprint and discuss about the how well the sprint went.
2. They give demo to the customer about whatever they built.
3. They will plan for the next.
4. Product owner tells what is done and what is not done.
5. Also they discuss about how to plan next Sprint.

3 clones

UAT

BBT

AT

Chicken: person comes who and check.

Spill Over: There are certain features or stories that you can't build in current Sprint and will move to the next Sprint.

Chicken: some people don't work in sprint but they will be there to observe what's happening.

5. Smoke Testing:

Testing the basic or critical feature of an application before doing thorough testing is called Smoke testing.

It can also be called as Sanity testing (or) Skin testing (or) Dryrun Testing (or) Built Verification Testing.

Why we do smoke testing?

Just to ensure the product is testable.

We do smoke testing in the beginning.

To catch bugs in basic features and send it to development team. So that they have sufficient time in fixing it.

Just to ensure the product is installed properly or not.

When we do smoke testing we do only positive testing.
Here we test only basic or critical feature.
Here we take basic feature and test for important scenario.
Whenever the build comes to the customer before the customer does acceptance testing we also do smoke testing.

When the product is installed in production we do quick smoke testing to ensure product is installed properly.
Also it act as a health check for software.

6. Adhoc Testing:-

Testing the application randomly is called Adhoc testing (Monkey testing (or) Gorilla testing).

Why we do Adhoc Testing:

End users use the application randomly and he may find the defect, but test engineer test the application systematically so he may not find the same defect.

Inorder to avoid this scenarios Test engineer should go & test the application and behave like a end user.

Development team look at the requirements and build the product & Testing also look at the requirements and test the product, by this method the testing team may not catch many bugs they think everythings work fine. Inorder to avoid this we do Adhoc testing.

Adhoc is a testing where we don't follow the requirements.

When we do Adhoc Testing:

When product becomes stable then only we go for Adhoc testing.

When the feature is new we will not go for Adhoc testing.

While doing Smoke testing we don't do Adhoc testing. Because if we do Adhoc testing we will not get time to test the basic feature.

When ever we are free or after testing the requirement if sometime is left out we should spend time in Adhoc testing.

If we get too many Adhoc scenarios note down the scenarios and execute it whenever we get time.

7. Reliability Testing:

Testing the functionalities of an application for a particular period of time is known as reliability.

8. Recovery Testing:

Testing the applications to check how well it recovers from the crashes or a disaster.

9. Accessibility Testing:

Testing the application for physically challenged persons.

It is also known as ADA

(American Disability Act)

10. Exploratory Testing:

Testing - the application without following any formal Document (requirement).

→ They won't give req

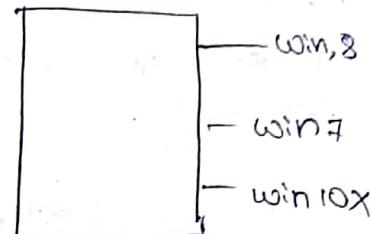
→ They will give document which is having picture or Data flow.

17/9/19.

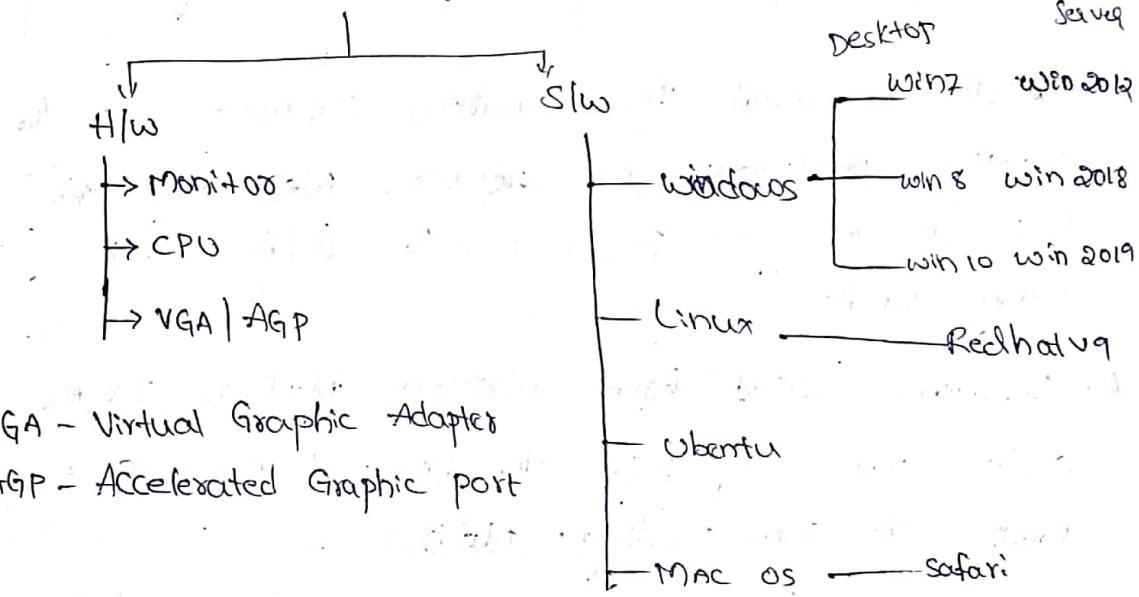
11. Compatibility Testing:



Production



Compatibility



VGA - Virtual Graphic Adapter

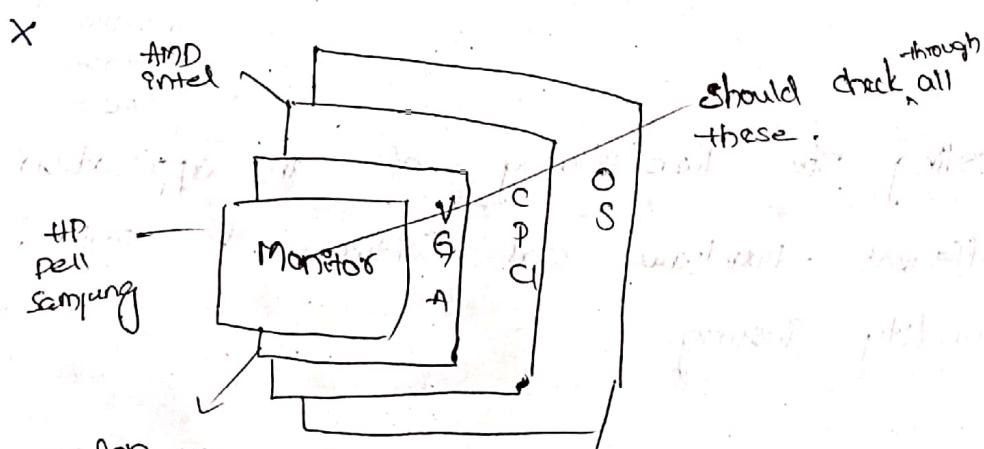
AGP - Accelerated Graphic port

Testing the functionality of an application

on different Hardware and Software is called
Compatibility Testing.

Why we do Compatibility Testing?

1. Developers develop the application in one platform. If test Engineers test in the same platform when the application is launch to business end customer might use the application in different platform. Because of that certain features may not work and that spreads the bad name in the market and the number of customers who buy this product will reduce. To avoid such situation we do Compatibility Testing.
2. To check whether features are working consistently in all the platform we do compatibility Testing.
3. Developers would have written common code for all the platforms or browser. We may have to test it on all platforms to confirm that really works.
4. Developers would have written platform specific code we may have to check whether code works in corresponding platform.



- . Different HW & SW vendors GUI in different ways. So we should check whether our application is rendered properly in diff combinations of HW & SW.

ROI: Return On Investment.

When we do compatibility testing?

1. When the product becomes stable on base platform then we think about testing the application in different platforms.

How to do compatibility testing?

Hardware Compatibility Testing:

Testing the functionalities of an application in different hardware environment, here we might test applications in different hardwares like

1. Different processors

1. Different Speed

2. Different make (intel, AMD)

3. Different Bit size (64, 32 bit)

2. Different Mother boards

1. Different make (intel, mercury)

3. Different VGA cards

4. Different Monitor with different resolution

12. Usability Testing:-

Testing user friendliness of an application.

Let us consider an example we have two applications which are different by doing the same job, we see which one is user friendly. Given below are some of the parameters we look into for testing.

1. Speed
2. help
3. Navigation should be simple.
4. compatibility
5. look & feel.
6. Features
7. location of components

One important parameter other than the above said parameter is "Effort needed to learn the application".

Consider an example - the SW A takes 2 hrs to understand but we take 4 hrs to understand Software B.

Let us see different cases here.

1. Since we understand SW A in 2 hrs it becomes user friendly compare to B.
2. Suppose look and feel is not good for A in this case though we understand A in 2 hrs we can't say that A. is userfriendly.
3. ∵ we look into many parameters before we say userfriendliness of a software.

What is look & feel?

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

How to conduct usability testing?

Prepare or derive checklist. We don't prepare a checklist we may miss some features in the application.

Example of checklist of an application:-

* whenever we click on click

1. For the application, one of the checklist include color of already checked link should be changed to red.

2. All images should have alt tags.

3. All the links pages should have link to home page.

4. Login page should have forgot password link.

5. Like the above check list we can derive as many checklist as possible based on the application of product.

6. While deriving checklist, we should derive common checklist which can be executed for all pages.

7. There is another case where customer gives the checklist for application.

Test Cases

What are the drawbacks if we test the application by seeing requirements?

1. There will be no consistency in test execution.
2. Quality of testing over the period of time depends on:
 - a) Memory power of Test Engineer.
 - b) Quality of testing depends on mood of TE.
 - c) Quality of testing varies from person to person.
 - d) If Engineers are more experienced they derive more scenarios.
 - e) If Engineers are less experienced they derive less scenarios.

What is test case?

It is a document which contains all possible Scenarios for specific requirement.

Test cases contains different sections like

1. Step number
2. Description
3. Input
4. Expected Result.
5. Actual Result.
6. Status
7. Comments

When do we write the test cases?

1. When developers building the product the tester will start writing the test cases.
2. When developers add a new feature test engineers write a test cases for the new features.
3. When developers modify the feature test engineers will modify the test cases.
4. When developers delete the feature test engineers will delete the corresponding test cases.
5. When developers completes the product and gives the build to testers, tester will start testing the product according to the test cases written.

process of reviewing the test cases:

1. Developers Start building the product.
2. Testers will start writing the test cases.
3. Once he (she) finishes all the possible test cases tester will give all the test cases to his/her TL through mail.
4. Allow TL will assign this test cases for review to STE.
5. Once the STE reviews and gives his/her comments on addition or modification or missing scenarios.
6. Allow STE should correct all the mistakes and get reviewed by STE, the STE should review again and check whether all the correction made and send it to TL for approval.



Why we write Test Cases?

1. If we write the test cases to have better coverage. When requirement comes in the developers are busy in building product. At the same time test engineers are free so they identify all possible scenarios and document it.
When build comes we can spend time in executing scenarios because of this number of scenarios we are covering is more.
2. To have consistency in test execution that means if you document the scenarios we can make sure that we can are executing all the scenarios in all test cycles of sprint or release.
3. To depend on the process rather than the person.
4. To avoid training to every new engineer on the product.
5. Test case is the only document which acts like a proof for customer, development team and management team. So that we can tell to all the teams that we have covered all the possible scenarios.
6. 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 some kind of coverage is there in automation script.

7. If we have documented the test case we don't have to remember the scenarios.
8. If we have documented the test case "TC execution will happen in very organized way."
9. If we have written the TC time taken to execute the TC is very less.

Test Case Name: CBO - Amount text field
 Requirement No: 3.032, 3.031, 3.0.3 - (BO-SRS)
 Severity: Critical
 Test Data:
 Pre-condition:
 Test Case type: functionality
 Brief description: Verify Amount text field

Step No.	Description	Input	Expected Result	Actual Result	Status	Comments
1.	Enter -ve integer value into text field	-100	Should throw appropriate message			
2.	Enter characters into Amount text field	Hundred Rupees only	Should throw appropriate message			

Author: Chinky (TE) Approved by: Pinky (TL)
 Received by: pinky (STE) Approval date: _____

Test Case Design Techniques:

1. Error Guessing
2. Equivalence partition
3. BVA (Boundary Value Analysis)

1. Error Guessing:

Guess all possible defects or errors we guess the errors based on experience, intuition, By requirement.

Ex: -100, Blank, 100.0, ABC

2. Equivalence partition.

straight forward.

Pressman method

If I/P range values
then design TC for
1 Valid & 2 Invalid

If I/P is set of values
then design TC for
1 valid & 2 invalid

If I/P is boolean
the design TC for
true & false.

deviation

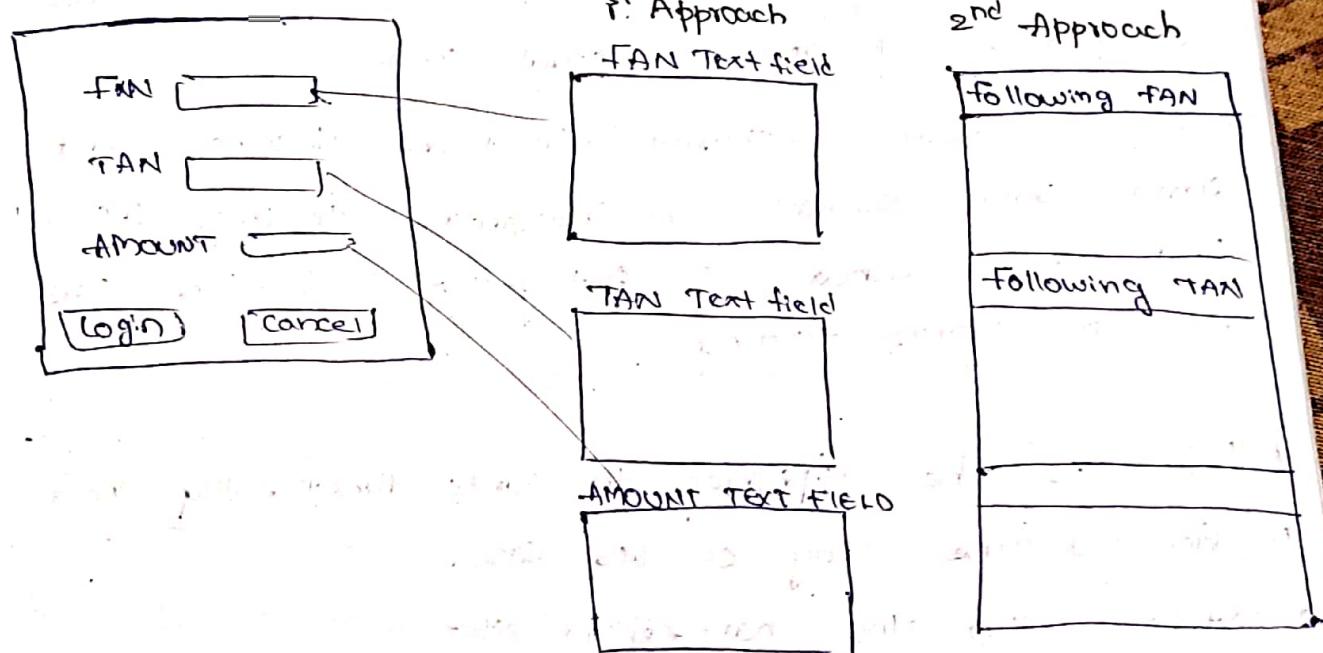
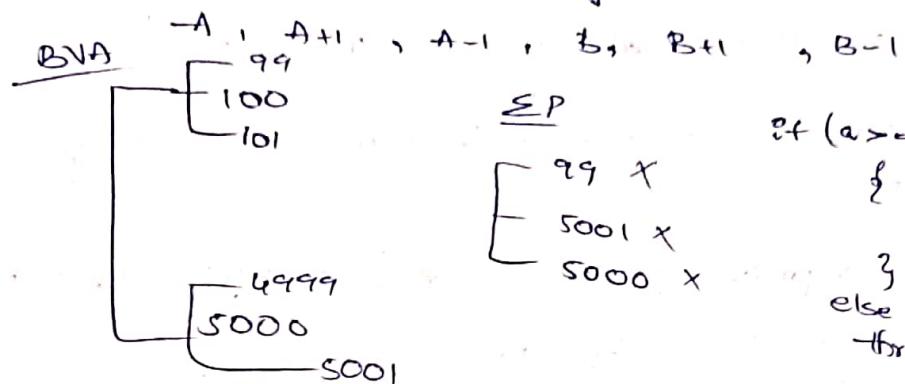
Practice method

req	deduct	transfer
If amount 100-2000	99	1000
deduct 1% & transfer	2000	3000
If amount is 2001-3000	3000	4000
deduct 2% & transfer	5000	5001
If amount is 3001-4000	5000	4001
deduct 3% & transfer	4001	5000
If amount is 4001-5000	5000	4001
deduct 4% & transfer	4001	5000

3 Boundary Value Analysis:-

If the input is range of values between A - B

then we are going to design TC for



lesson - 1

1. Before we actually start writing the test case we should come up with options.
2. Always start writing the test cases with navigation steps
3. Always use word 'should be' or 'must be' in expected result, don't use may be or can

4. Always write Generic test case, don't hard code the test case.
5. Elaborate only those steps in which you have focused, don't elaborate all the steps.
6. When we write the test cases, imagine the application.
7. If you organise the scenarios properly total number of steps will be reduced.
8. Some Company might use input column or some companies will remove actual columns.
9. If we cover scenario functional testing don't cover same scenario in integration testing. If you cover scenario in integration testing don't cover it in system testing.

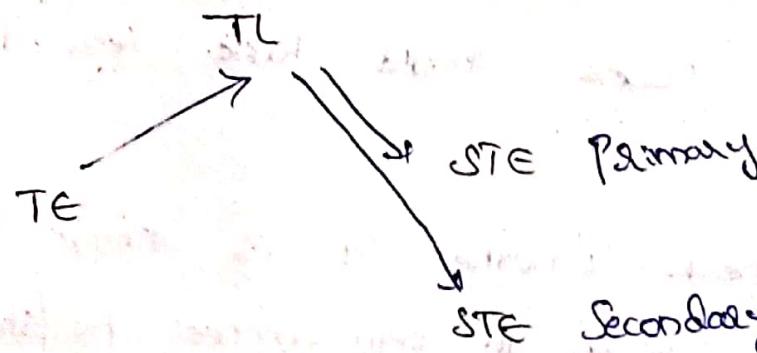
What is the approach to write functionality TC?

1. Go to the body of test case.
2. Start with the navigation steps.
3. Take first field and start writing the Test Case
 - 3.a) Start with valid data
 - 3.b) Cover the error guessing scenarios.
 - 3.c) Cover all equivalent partitioning scenarios.
 - 3.d) Cover all Boundary Value Analysis.
4. Take second component repeat 3.a, 3.b, 3.c, 3.d.

Step number	Description (actions)	Input	ER	Status
1.	Open the Browser enter the URL	Server names	Login page should be displayed	pass
2.	Enter the valid UN & pwd & click on login	Valid UN & pwd	Home page	fail block return
3.	click amount			

On what basis they assign the TC for some one to review?

1. There is fellow working on similar module in the project.
2. There is fellow worked on same project module in the previous project.
3. There is fellow working in this project since beginning and knows every corner of the product.
4. There is a fellow who is very responsible, he well understand the requirement very fast and understand more scenarios mistakes.



How do they ensure reviewer does his/her job.

1. TL will assign primary & secondary reviewer.
2. TL also should randomly review and find mistake.
3. Intentionally introduce the mistake and give it to the reviewer and check whether it is found by reviewer or not.

Test Case Review ethics :-

1. Always review the content not the author.
2. While reviewing spend time in identifying the mistake not in finding the solution for it.
3. Even after the review if there any mistake 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 header of the test case and understand the requirement for which the test case is written.
 - a) Missing Scenario
 - b) Repeated Scenario
 - c) Wrong Scenario
2. We will check whether the scenarios are organized and your test cases should take less time in execution.
3. We will check whether it is simple to understand so that when given to new engineer he/she should

able to execute it without asking questions.

4. Look into header of the test case and try to find
 - a) check all attributes covered or not
 - b) check whether content in all the attributes are correct or not.
5. Check whether test case format or template is correct or it should be according to the standard defined in the project.

Test Case Review Tracker.

S.No	Test Case Name	Step no.	Review Comments	Severity	Author Comments
1.	CBO - AT - allscenarios x 1s	Pre-condition	Missing pre-condition	Major	fixed
2.	CBO - AT - allscenarios x 1s	5	Transfer more than balance Scenario is missing	Critical	fixed

Review Comments

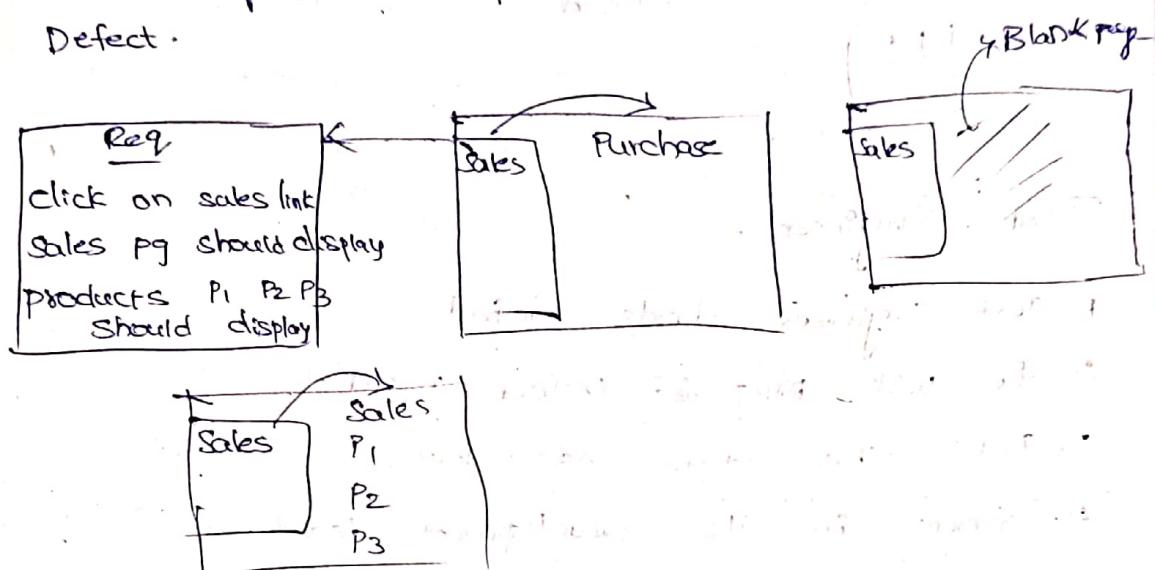
Closed

Not fixed

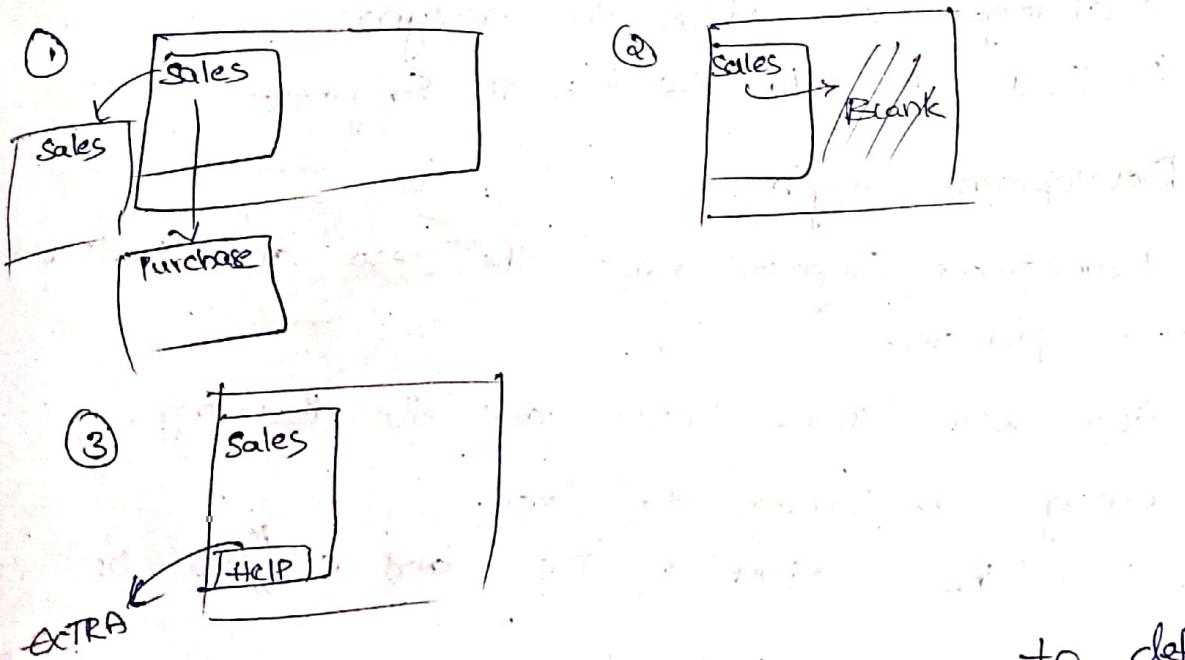
19/9/2019

Defect tracking:

Defect :- Any feature which is not working according to customer requirement or deviation from requirement specification is called as Defect.

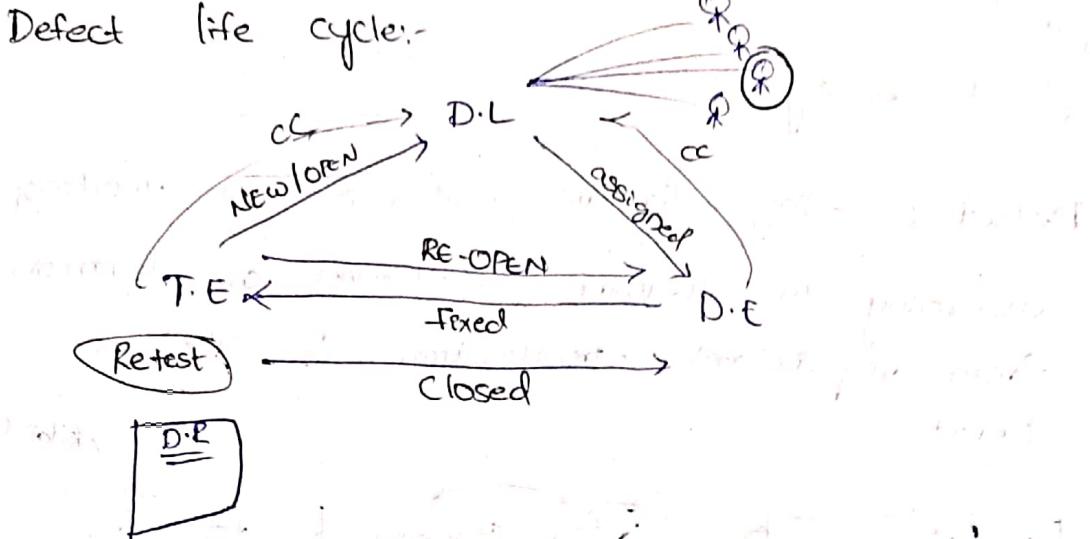


1. Wrong Implementation
2. Missing Implementation
3. Extra Implementation
→ coding.



Bug: It is a informal name to detect.

Defect life cycle:-



Test Engineer :-

1. Test Engineer finds defect
2. He will prepare defect report
3. puts the status as new
4. Sends it to development lead

Development lead:

1. Development lead reads the repo understands 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 Engineer reads the repo and understand the problem.
2. Goes with source code and fix the bug.
3. change the status to fixed.
4. Send the report to T.E and CC it to DL.

Test Engineer:

1. He will read the report and understand the problem.

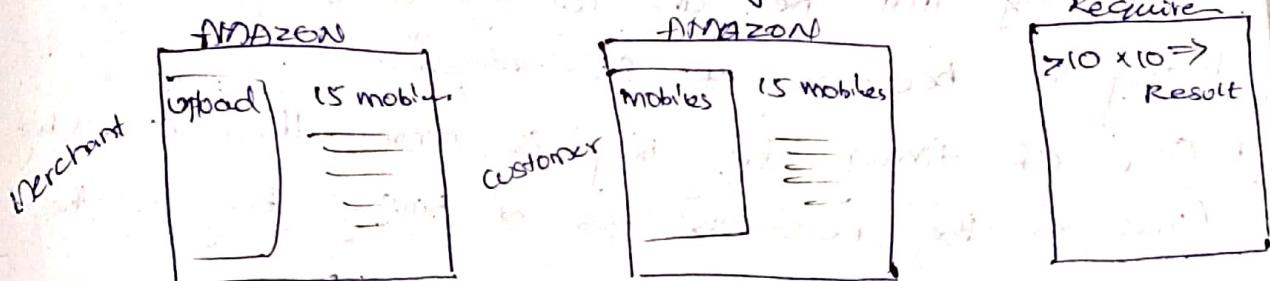
8. Retest the bug. If the bug is fixed he will change the status to closed.
9. Otherwise change the status to Reopen and send report to DL & DE.

What is Reject Status?

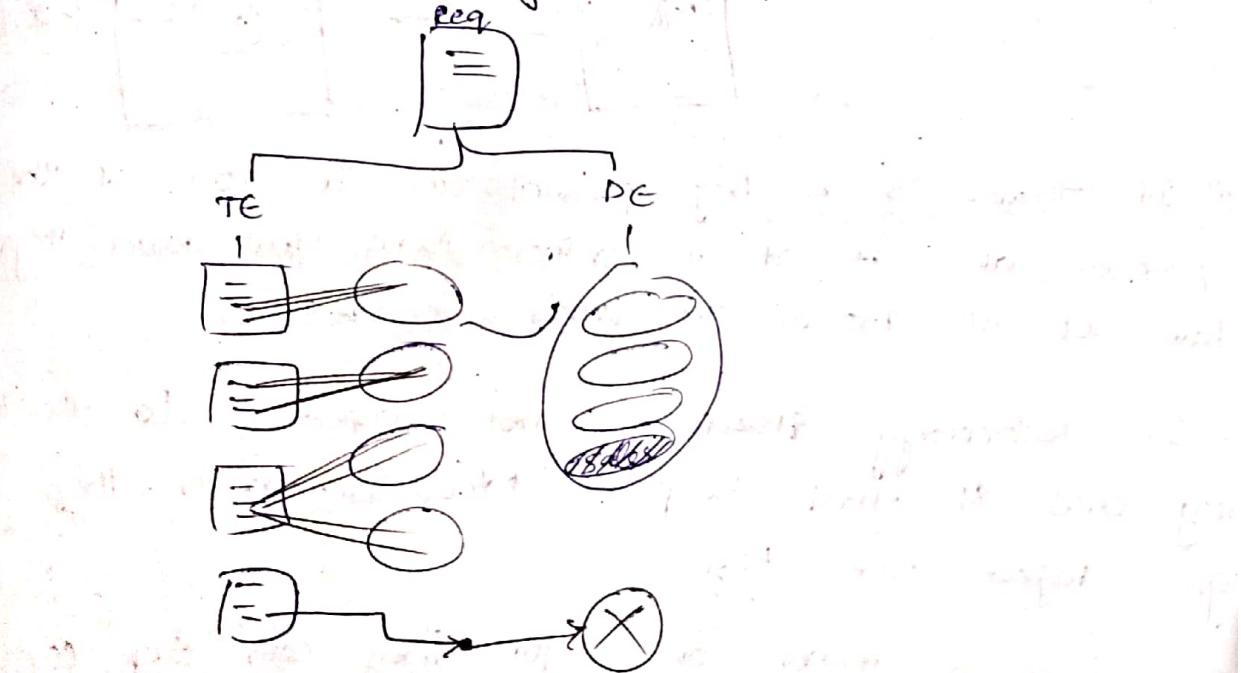
Test Engineer find the defect and sends it to Developers and Developers say that it is a feature and change the status to reject

Reasons :-

1. Because of misunderstanding the requirements



2. Because of referring old requirements



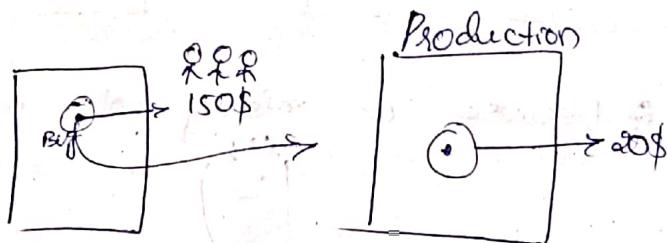
3. When you find extra feature as a bug and send it to developers. Developers might reject it in such a case TE should Reopen it and ask developers to fix it, so to update the req.

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

What is Cannot be fixed status?

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

1. Cost of fixing the bug is more than cost of bug. (loss in the business bcs 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.

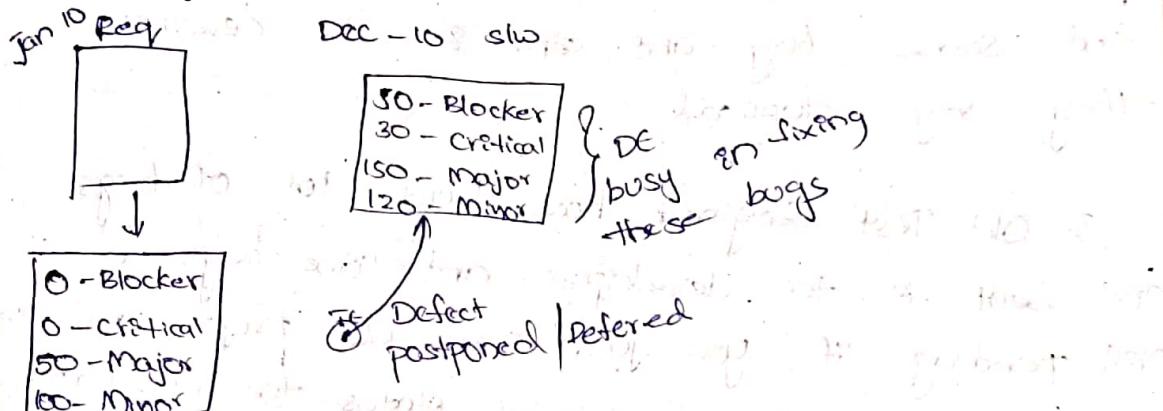
3. If technology itself is not supporting to fix the bug and if that bug is minor or major they might reject the bug.

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

What is postpone status?

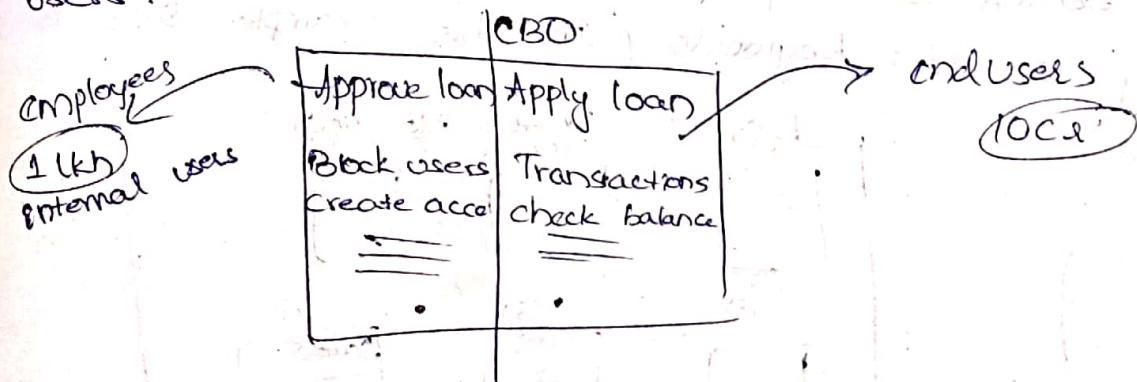
* Developers are accepting that it is a defect but they want to fix it little later because of few reasons.

1. There is a minor/major bug, developers are not having time to fix it in such a case they might postpone the defect.



2. 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.

3. If there is a minor bug exposed to internal users.



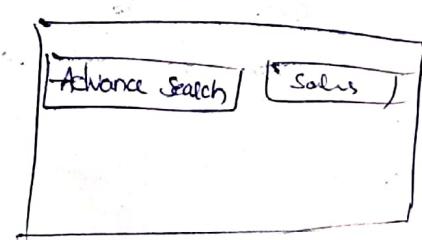
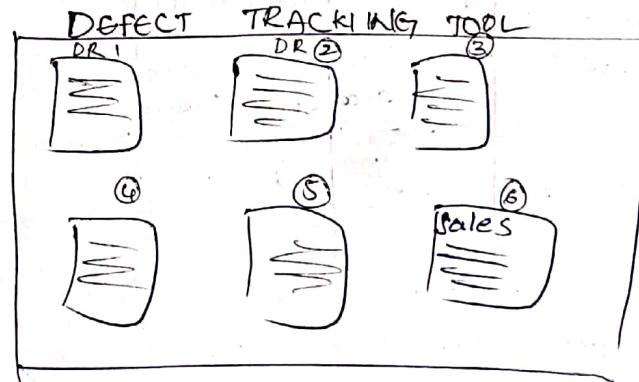
What is Duplicate Status?

* If you send a defect which is already sent by someone else then developer say Duplicate reasons

1. Because of common features.
2. Someone 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 Engineer have found lot of bugs and sent it to developers and some bugs are still pending if you join same old project and send same old bug they change status to Duplicate.

How to avoid Duplicate Bugs?

Before we prepare the report and log the bug should make sure that it is not Duplicate or we should search for duplicate bugs in the tool by entering certain keywords (In this example Sales is a keyword)



What is a lot Reproducible Status?

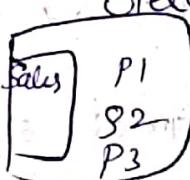
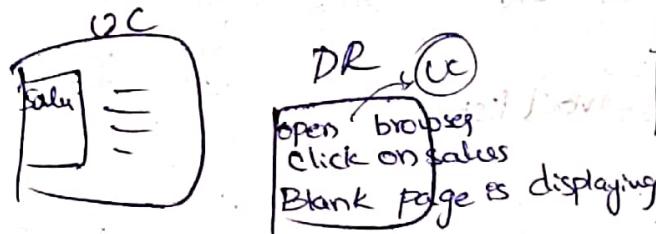
Test Engineer will be able to see the defect but developers will be not able to see the same defect in such a case they will say that is not reproducible.

Reasons:

1. Because of improper defect report

a) Because of incorrect platform

b) Because of incorrect data



Because of Inconsistent Bug:

Some time it appears some time it disappears

It is a problem which is not the part of requirement then it is called as

Request for Enhancement or change Request.

Request for

RELEASE NAME: Tiger DEFECT ID: 101 REl 20/9/186

BUILD NAME: SI BUILD ID: b04 BRIEF DESCRIPTION

STATUS: New/Open Assigned - fixed closed/reopened

SEVERITY: Blocker Critical Major Minor

PRIORITY: High Medium Low

TEST DATA: Gxl=abc, Psl=123

TEST ENVIRONMENT: Win 7, Win 8, Win 10
OS | browser

TEST CASE NAME: Gmail-SI-Mail list

FOUND BY: User Name

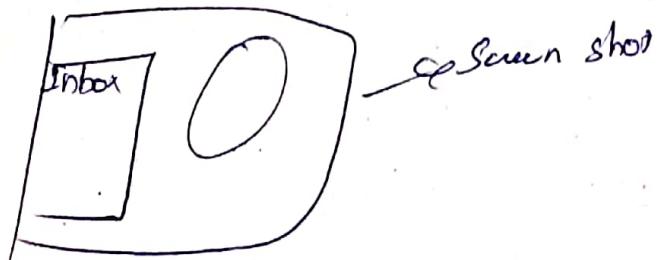
DETAILED DESCRIPTION: SI is not displaying on SI page.

BRIEF DESCRIPTION: Following are the procedure to reproduce the bug.

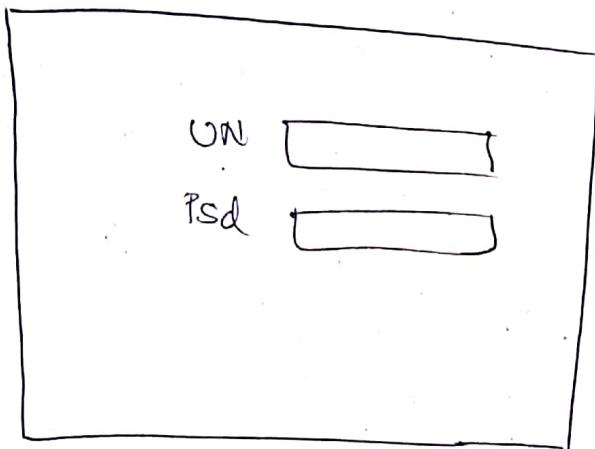
- ① login to gmail
- ② click on compose link
- ③ Enter all valid data & click on send button.
- ④ Click on SI link

E.R: SI should display on SI page

A.R: _____ Not displaying _____



1. Find Defect
2. Duplicate
3. Prepare Defect Report
4. Send it to D.L
5. Manage Defect life cycle. (DLC)



Add Engineer Bug ID
 A. Search
 log a defect. / New defect:

* Unique ID (Bug ID)

* stored in DB

* Send it to PL

* It should get updated \Rightarrow