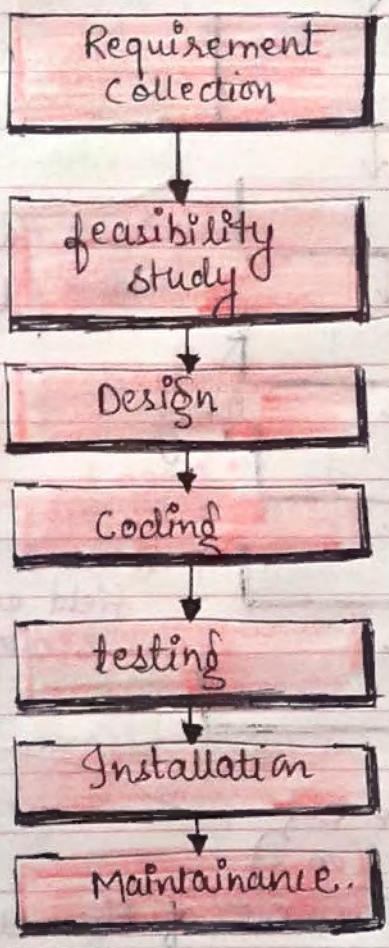
SDLC

Software Development Life Cycle.

* It is the standard procedure to develop the software.

* It has different stages :-

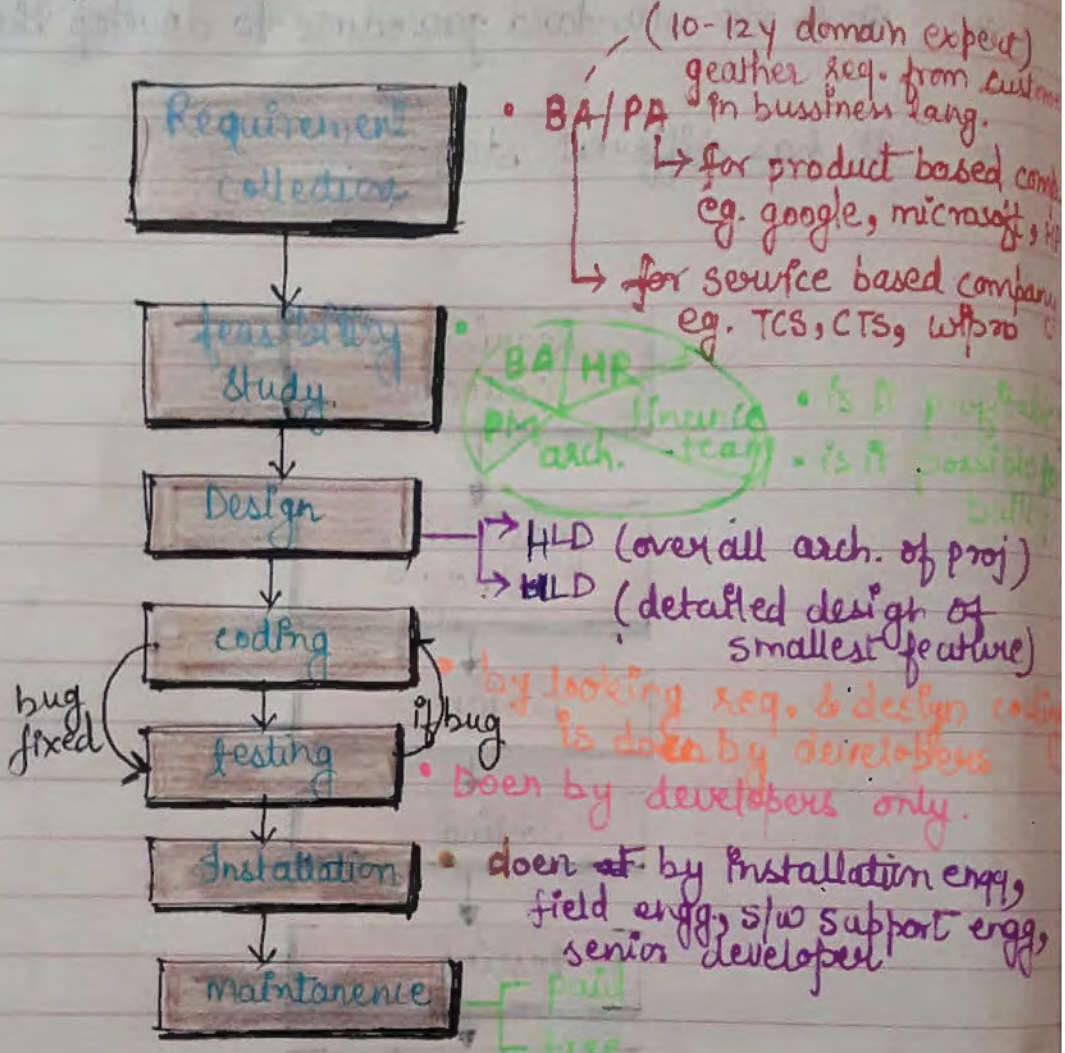


* Different models of SDLC are :-

1). Waterfall model
2). Spiral model
3). V model
4). Prototype model
5). Derived model
6). Hybrid model
7) Agile Model (used in current working project)

Waterfall Model

* It is also called traditional model or basic model.
* Stages of this model are :-



Requirement collection :

In this phase requirement of the customer will be collected by either BA (business analyst) or PA (product analyst) in some bussiness language. And then he will convert it into software language and explain it to developers, project manager, architect etc. Here BA acts like a bridge between customer & software company. BA will be hired, only if they have many years of experience in the specific field & they get trained & then sent to the client place.

who can become BA?

1). Domain expert (The person who worked on same domain for 10-15 y & has got very good knowledge)

2) Senior developer (8-10 y of exp in the same sort of proj. & has got very good project knowledge)

3). Senior test engg. (8-10 y of exp. in the same sort of proj. & has got very good project knowledge).

Feasibility study:

following group of people sit's & decide.

BA: who converts Bussiness language to software language

Architect: who checks whether the project is technically feasible?

HR team: HR team decides whether if there is a resource needed & how many needed?

Finance team: In the finance team they used to check the profit % or loss if they gets the needed profit %. they approve.

Project manager: All the people who send the report to the project manager about their analysis. He is the final decision authority of the project.

once software requirement collection is completed then only we go for feasibility study. This is done by a team which consist of pm, BA, HR, architect, finance team. This is the stage where company will decide whether

to take up the project or not. If we take up the project company checks do we have sufficient resource, sufficient technology & sufficient lab setups.

This is the stage where company will get to know if they take up the project, do they get profit or not.

Design:

Designing of the project is done by either Architect, senior developer. as well as content design & web designer also.

Design is of 2 types

- HLD (high level design)
  where we design the overall architecture of the project.
- LLD (low level design)
  where detailed design of the smallest feature is done.

Coding

By looking on to requirement & design of the project line-by-line code is written for each module by set of group of developers (senior developers, junior developers & fresher developers)

Senior developer: he will handle critical section (daily used/ frequently used)
Junior developer: he will handle major section (eg. edit, forgot pwd)
Fresher: he will handle minor (eg. about, help, profile)

Testing

Here the application / project is tested & if there is any bug it is reported & sent back to the developer to fix it. once fixed it's retested until the s/w gets stable. It is again done by the set of testers (senior tester, junior tester, freshers testers)

Installation

- In one time installation, senior developer will do the installation.

In case of many installation (eg. health app.) - field engg/ installation engg. will do the installation.

onces s/w testing is completed & s/w is ready we will install the s/w in customer place so that the customer can use the s/w & run the bussiness.

Maintainence :

onces the s/w is installed in customer place customer will use the s/w & run the bussiness. while using the s/w, customer ~~can~~ face any problem, company will fixed the defect for freely (for limited period eg 1-6 months) & will cost some amount after that fixed period or for major change depanding on the aggrement the s/w company & the customer signed before. this phase is called maintainence phase.

- CRS (Customer requirement specification)
- BS (Bussiness specification)
- BRS (Bussiness requirement specification)
- BRD (Bussiness requirement document)
- SRS (software requirement specification)
- FS (functional specification)

Advantages:

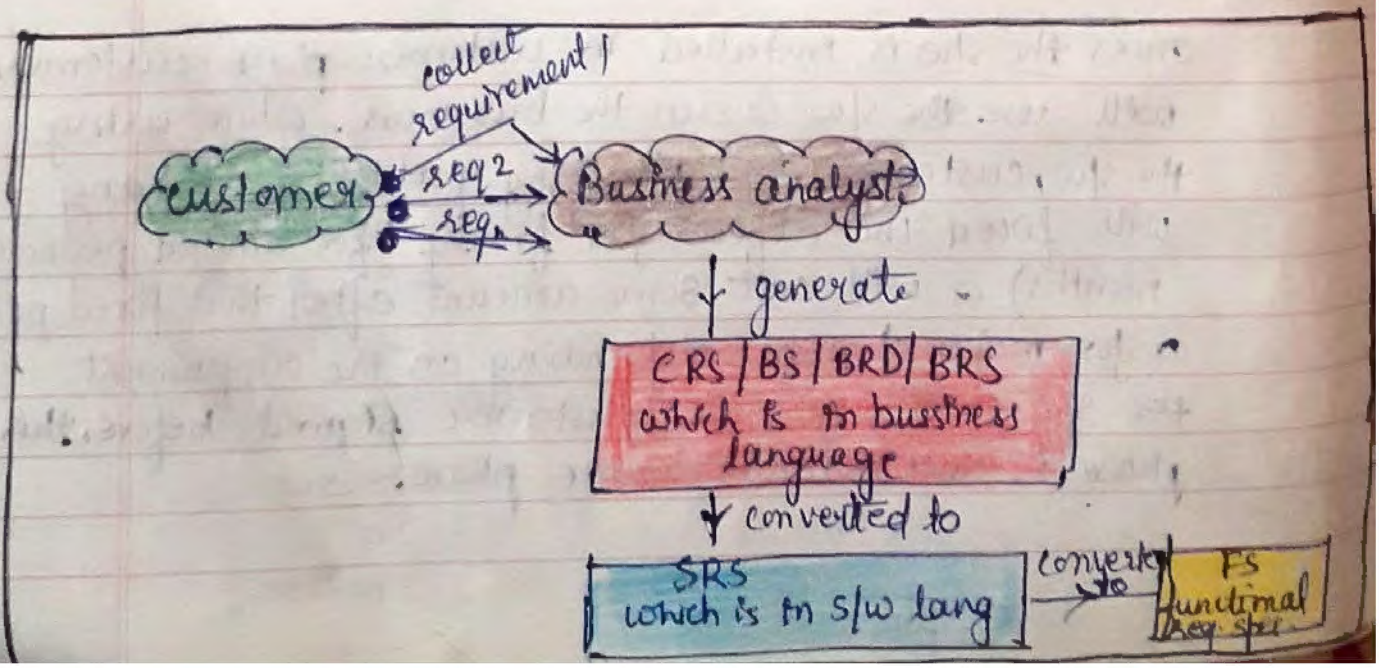
1. Simple model.
2. Initial investment is less
3. As requirements are freezed so we can expect good quality product at the end.
4. very 1st model

Disadvantages:

1. Only developers are evolved for testing.
2. Testing is done only ones coding is completed i.e there the requirement & design phase is not tested.
3. No back tracking can be doen.

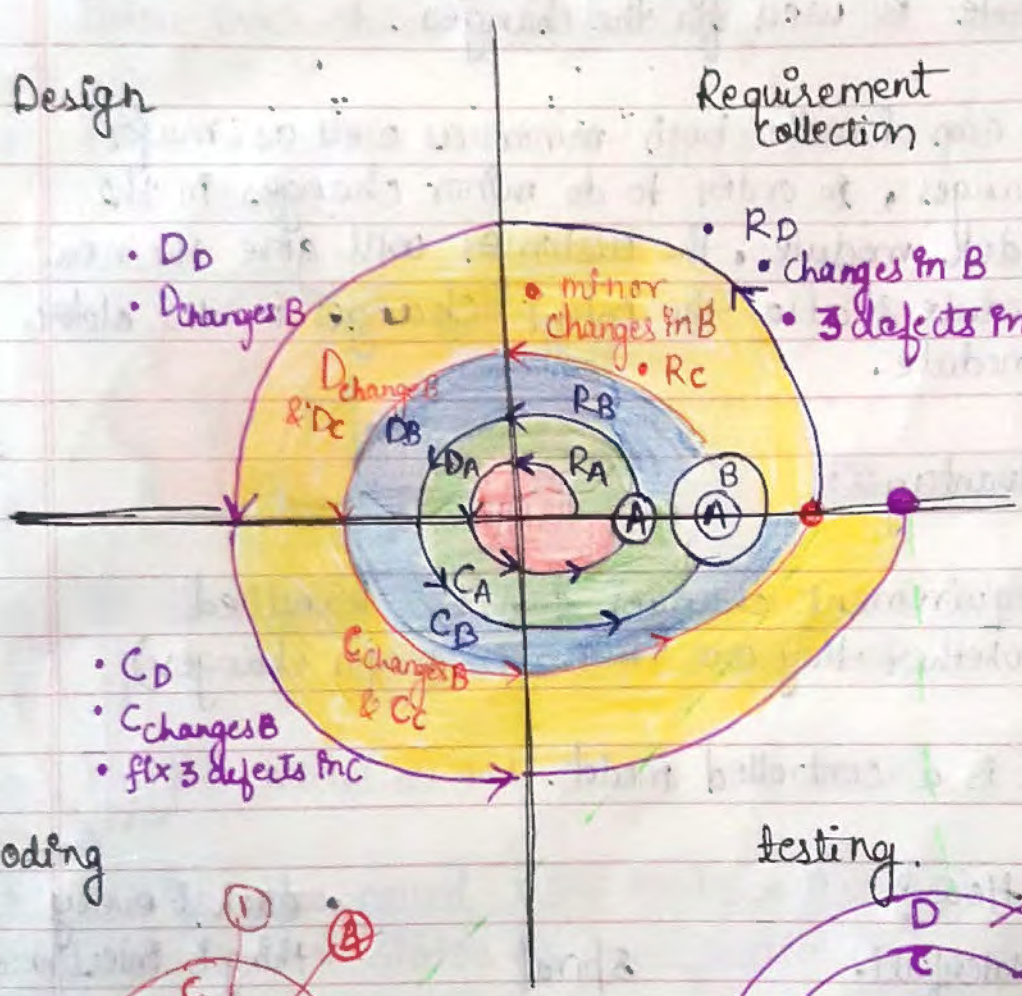
Applications:

- → for simple project
- → for short term project
- → for small applications
- → for projects where requirements are freezed at the beigning only.

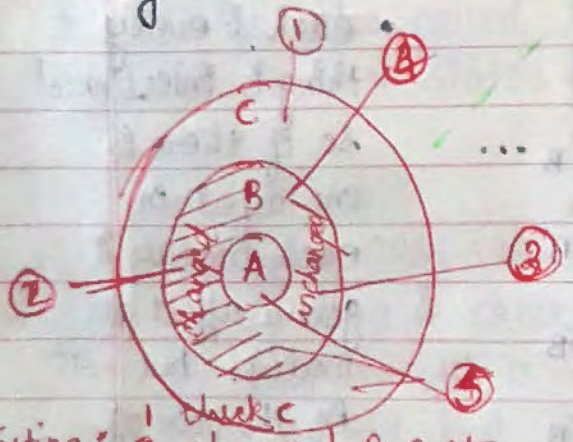


Spiral Model

* It is a systematic approach for developing the software, It has 4 phases ie. Requirement collection, Design, coding & testing.

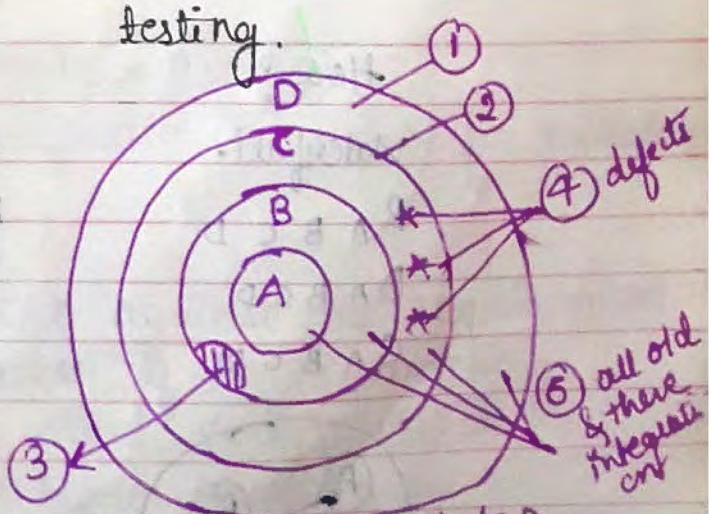


Coding



Testing:
1. check C
2. changed B part
3. unchanged B part
4. Integration of B & C
5. all old modules & there integration

testing.



Testing:
1. New module D
2. Integration with last module ie C & D integration
3. changes in B.

* When ever there is dependency in between the modules of the project we go with spiral model.

* In this model we can do changes once after the module is completed.

* In order to change the design the complete cycle ~~circle~~ is used for the changes.

* It can handle both minor as well as major changes, In order to do minor changes in the older module, the customer will give the new module & also the minor changes in the older module.

Advantages:

1). Requirement changes can be handled (whether they are minor or major changes).

2). It is a controlled model.

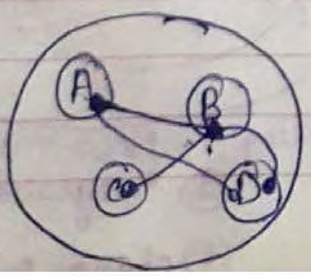
How?

Waterfall.

R_A B C D

D_A B C D

C_A B C D



Spiral

R_A R_B ...

↓ ↓

D_A D_B

↓ ↓

C_A C_B

↓ ↓

T_A T_B

↓ ↓

A B

- Each & every thing is interlinked so if there is one defect in each module it gonna affect all modules but not in the case of spiral model.

3). Customer gets an opportunity to see the s/w in every stage.

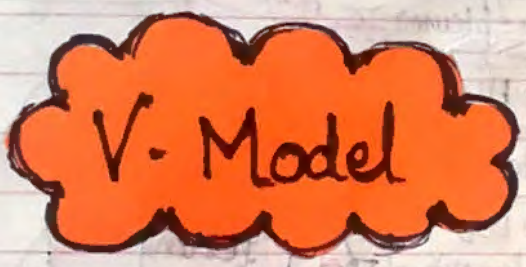
4). Customer gets an opportunity to ask for the changes in every cycles.

Drawback - 1, 2 of waterfall.

Applications

1). When ever there is dependency in the modules.

2). When ever the customer gives the requirement in stages.



* It is a step-by-step procedure or a standard procedure to develop a new software.

* In this model all the stages are tested.

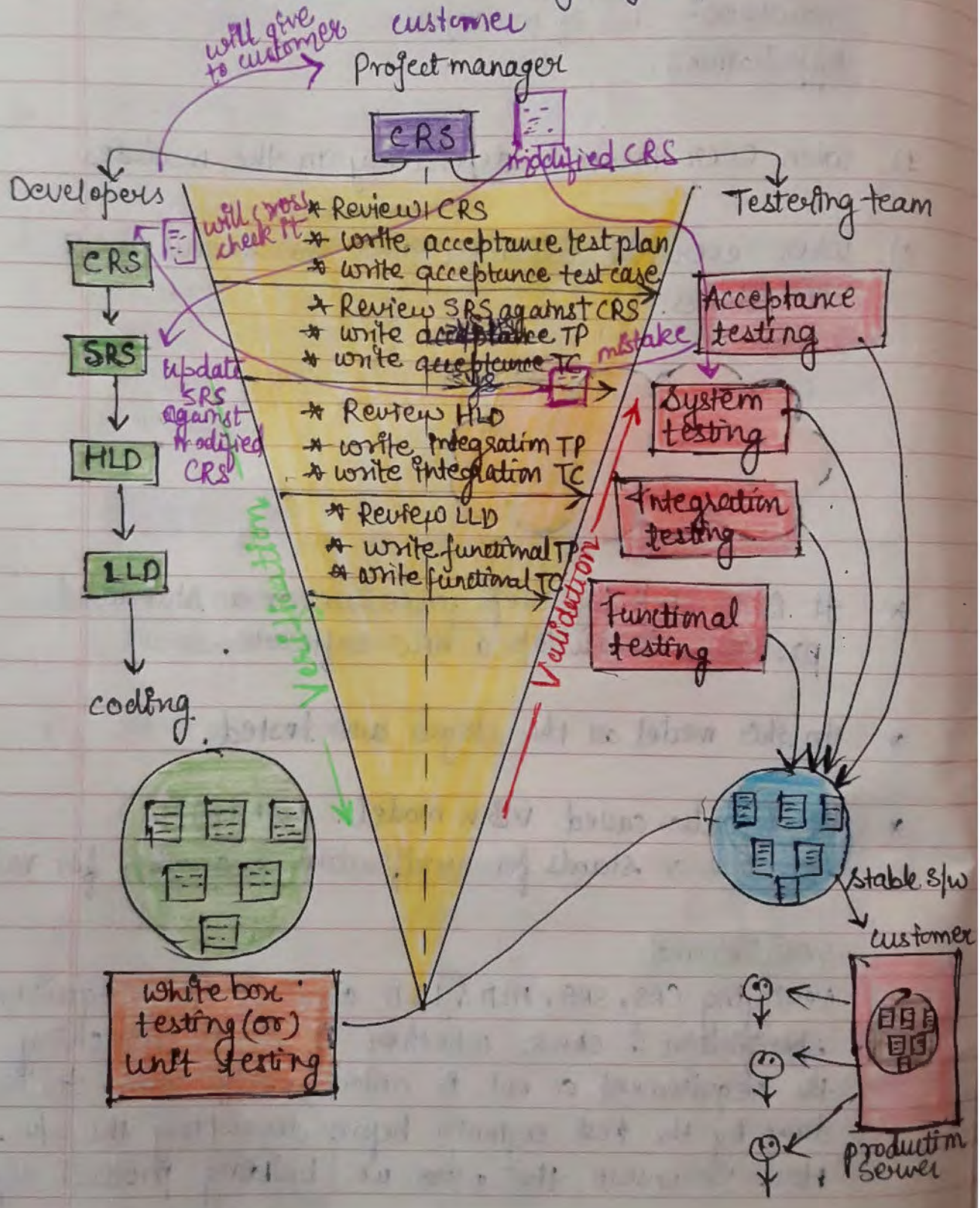
* It is also called V&V model. (why?)
here one V- stands for verification & another for validation

Verification:

Verifying CRS, SRS, HLD & LLD against the requirement specification & check whether it is matching the requirement or not is called verification. It is doen by the test engineer before developing the s/w. Here we ensure that, are we building product right?

Validation:

Testing the functionality of an application by executing the test cases is called validation. It is doen by test engineer onces s/w is developed. Here we ensure that are we building right product.



* name V-model is given becoz of the V shape, we get we for whole all ## get.

Firstly the CRS is prepared by BA and than the project manger provides this CRS to both the teams ie. Development team & testing team.

Now Developer will start converting this CRS to SRS and test engg. will simultaneously Review the CRS and will write acceptance Test plan & ~~the~~ acceptance test case. & ~~thus~~ they while reviewing the CRS they may get defects like:

1). Missing requirement
2). conflict requirement
3). wrong requirement

Missing requirement:

Any requirement that is mentioned but not explained is called missing requirement.

eg. 
CRS

page4: OD will be used (overdraft feature)
- but not explained in whole CRS

Conflict requirement:

eg1.

Loans.
5th page: loan manger should not approve loan. 15th page: <10k - Loan manger should approve. >10k <50k - Branch manager -"- >50k - Regional manager: -"-

eg2.

Health care application.
5th page: followup fee < consultant fee. 15th page: followup fee = consult fee.

wrong requirement:

eg:

Type of loan [select]
- Home loan
- personal loan
- car loan

Amount [ ]
- Fixed Intrest
- floating Intrest

[Submit] [Cancel]

In SRS

Fixed Intrest
- when ever user apply the loan through fixed Intrest, same Intrest should be charged till he repays.

Floating Intrest
- when ever user apply the loan through floating Intrest same Intrest should be charged till he repays

copy → paste

wrong req

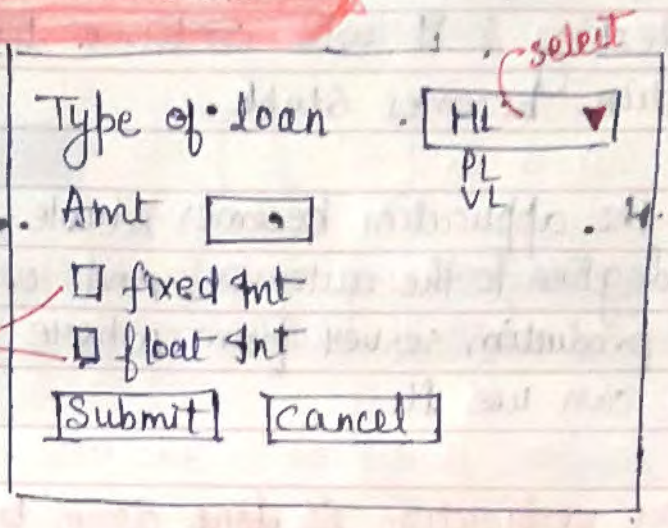
Now we will send this mistake to development team, then they will cross check it & if its a mistake they will send back to customer. customer will update the CRS & send back. Now with updated CRS developer's will update the SRS & test engg's will again review the updated CRS. this cycle continues till the test engg's approve it all alright, here 1st phase is completed.

Now, in second phase developers will be involved in converting SRS to HLD till then testing team will review the SRS against CRS will write system test plans & system test case.

while reg reviewing SRS against CRS the test engg. may find defects like:

1). Wrong conversion
2). Missing conversion
3). Not converted..

e.g.



• mistakes.

If mistakes are there, it will be send back to developement team. then they will fix it & send back to the testing team.

Now in third phase, the developers will converted HLD to LLD & testing team will review HLD as well as will write integration test plan & integration test case.

In forth phase, the developers will be involved in writing code for LLD and the testing team will review LLD & will write functional test plan & functional test case.

Now the developers will do white box testing on the code they have written & will send to the testing team. And testing team will apply all the written test cases & test plans on the given application & will perform functional testing, integration testing, system testing & acceptance testing. If they find

any defect it will be reported back to dev. developers simultaneously. Then the developers will fix it & send the build back to testing team to retesting & it will continue till the software/ application becomes stable.

onces the application becomes stable, it will be sended given to the customer. And will be installed on its production server from where all the end users can use it.

So, here verification is done before the s/w is developed & validation is done after the s/w is developed.

Advantages:

1). Testing starts in the early stage of product development, this avoids downward flow of defects which intern reduces lots of rebug.

2). All the stages are tested.

3). Deliverables are simultaneous, means developers are delivering SRS same time testing team delivering acceptance TC & TP so on becoz of which work gets completed faster.

4). Total investment is less.

Drawbacks:

1) Initial Investment is high.
2). Involves too much documentation work.

Application:

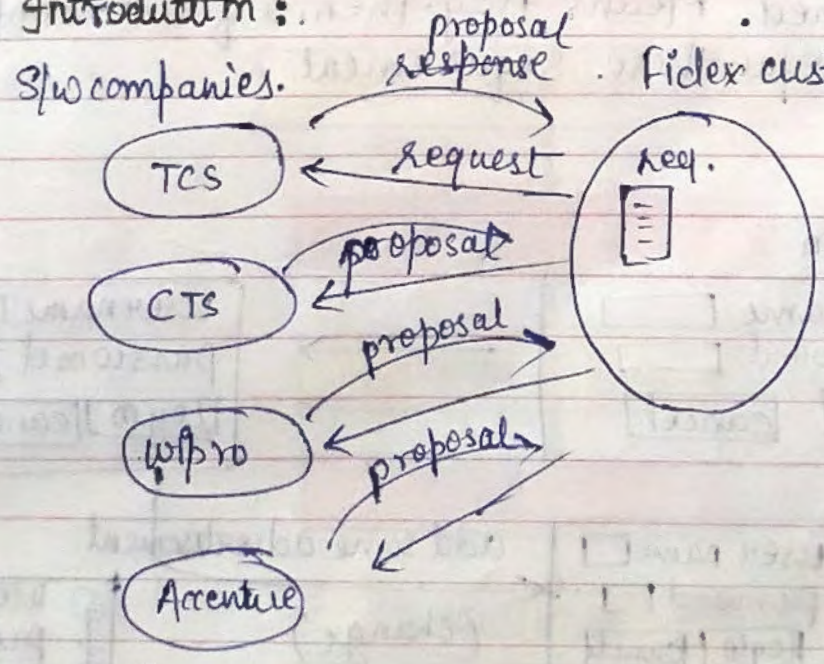
1). When ever the customer are expecting very high quality of product within a tight schedule.

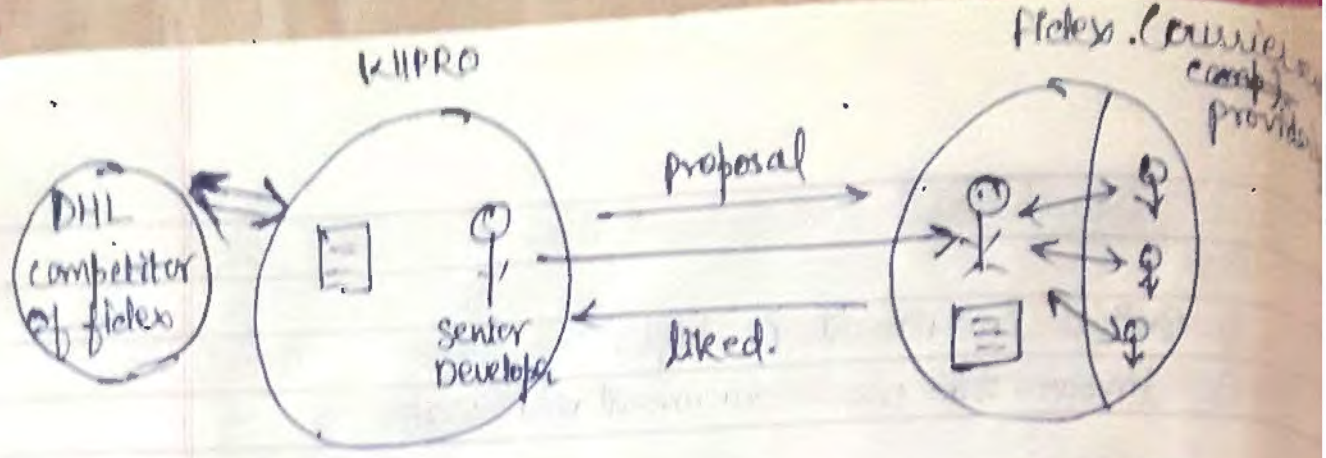
2). When ever we are creating a complex projects

3). When ever we are developing long term projects.

Prototype Model

Introduction:



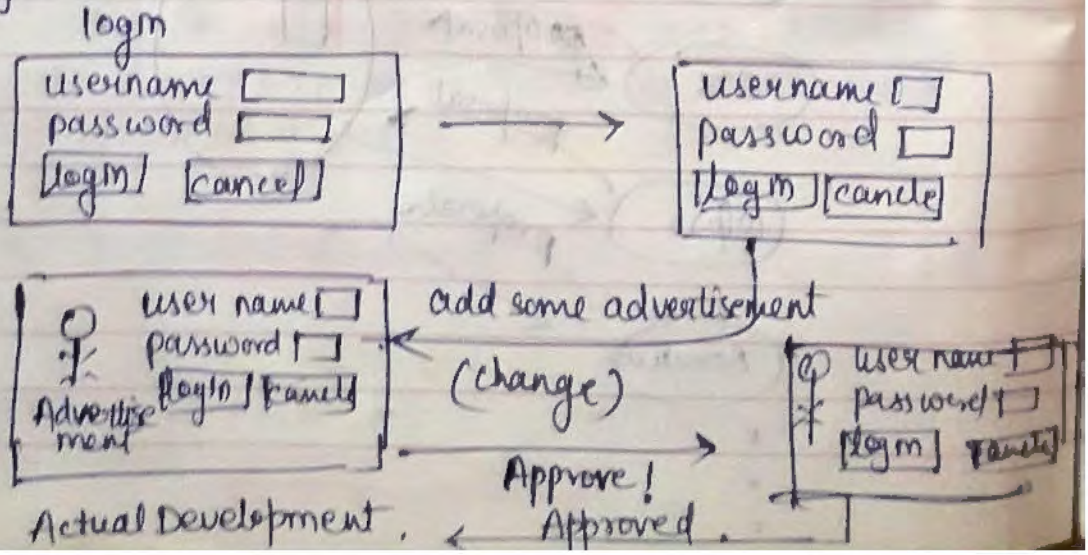


fidex liked wipro's proposal (as wipro has throug-thrown the bisket that we have just made the project for your compititor & we can develop some thing more better for you).

Now senior develop of wipro will go to fidex & try collecting the requirement but some employ there, say the & suppose he is collecting requirement for cancillation of courier.

Now some of the employ says 50% of the amount should be refuned some says 25% of the amount should be refuned some says no charge should be taken if its not dispatched; some says 100% should be charge if its dispatched. Means they themself are confused with regar there requirement.

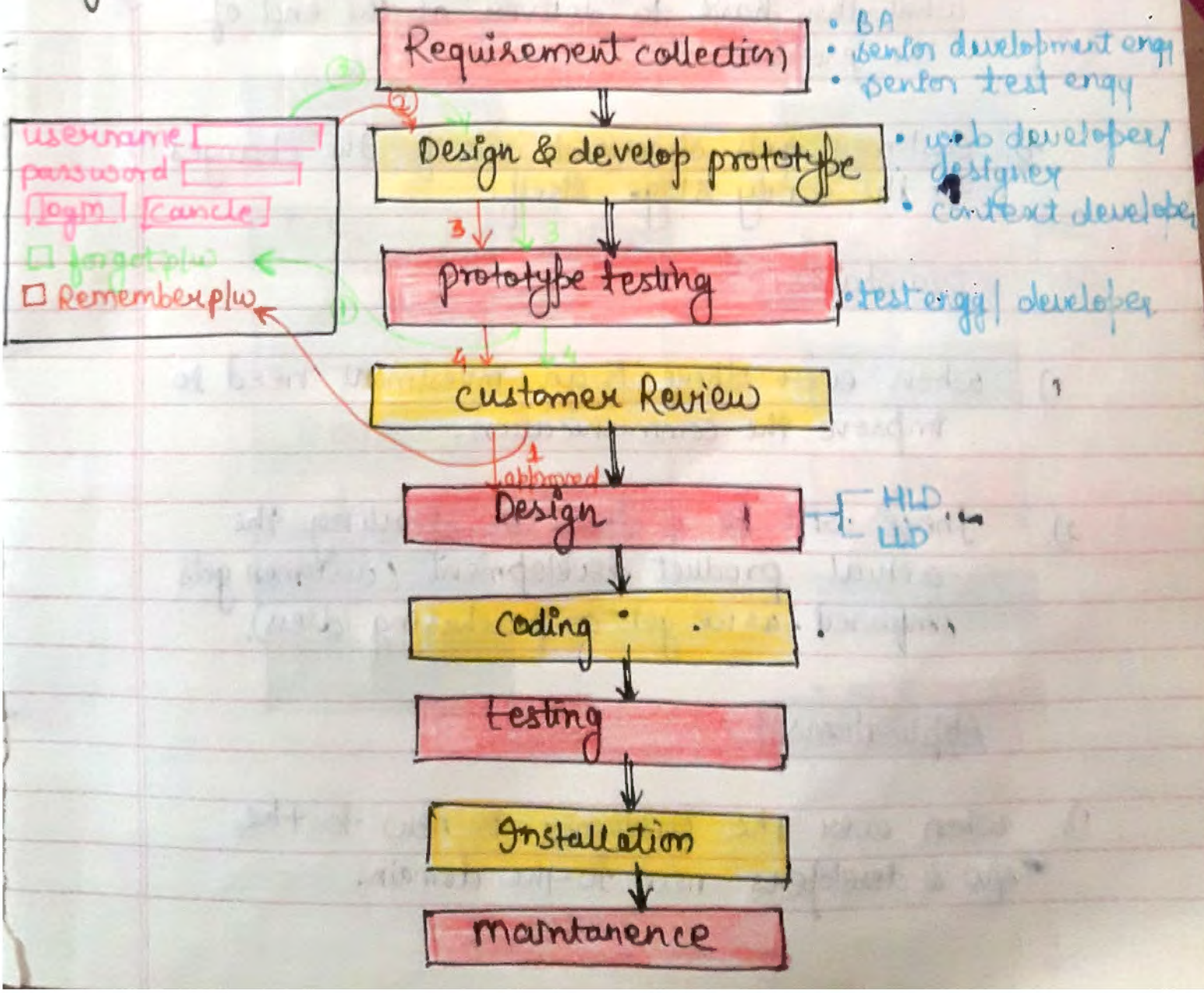
eg2.



prototype model is used where user the customer has huge expectations but does not have clearity in the requirement.

thus first prototype (template) is send to the customer, if it gets approved by him then only the real develop design, coding & testing is doen. And a customer can even denay later that what we developed is not what he wants. So its better to get approved by showing him the prototype of the project so make him clear, that how is this his project is going to look later.

Stages:



In prototype testing we check whether all the components are present according to requirement & look & feel of the prototype.

Advantages:

1). There is a improved communication b/w customer & s/w company.

2). In the beginning only customer is known what he gonna get at the end of the proj.

3). In the beginning only developer will be knowing what they have to deliver at the end of the project.

4). Customer gets chance to ask for the changes in the early stage itself.

Drawbacks:

1). When ever there is an investment need to improve the communication.

2). There will be a delay in starting the actual product development (customer gets confussed - as we get on purchasing dress).

Applications

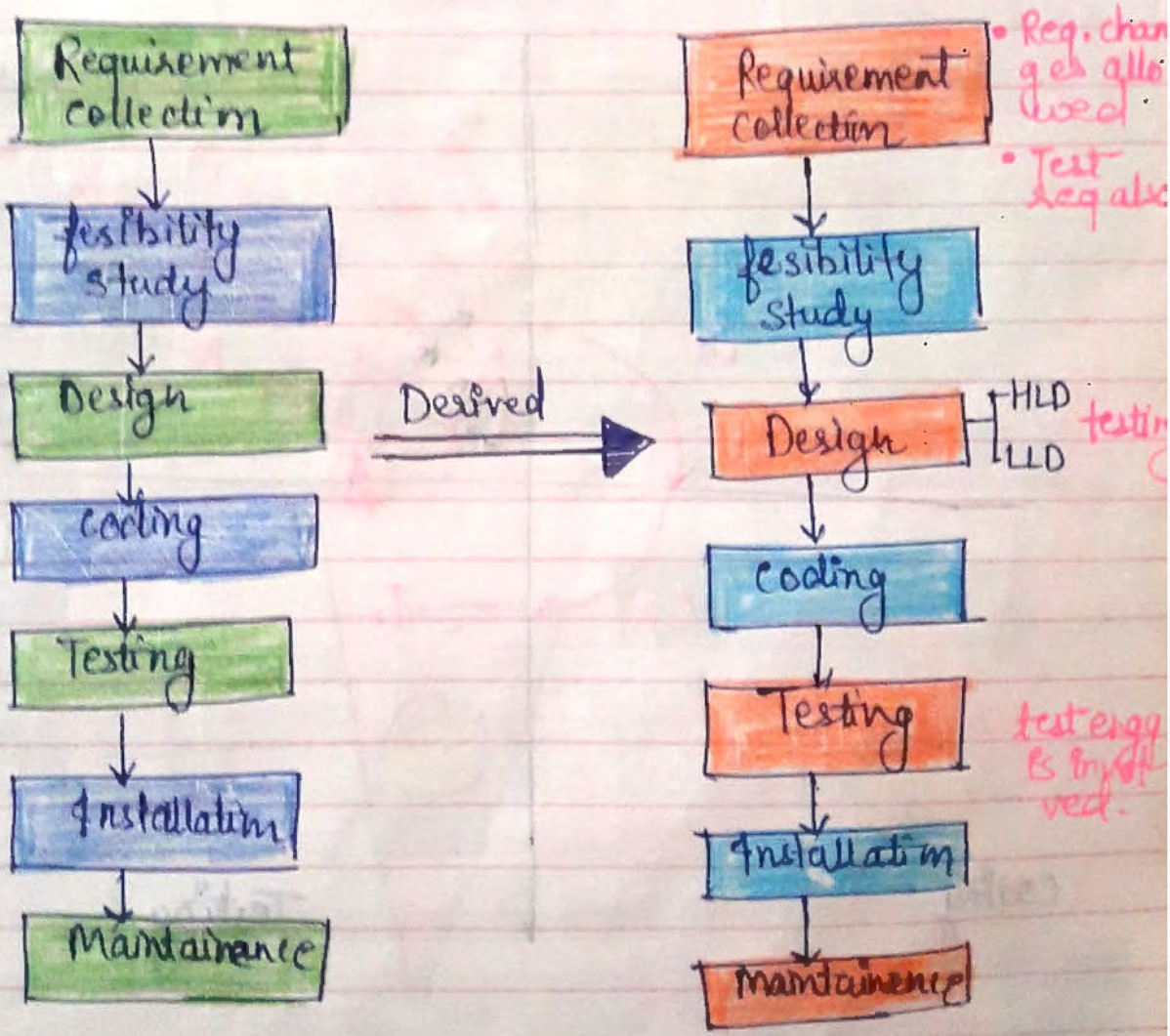
1). When ever the customer is new to the s/w & developers new to the domain.

1). when ever the customer is not clear about his own requirement.

Derived Model

* Here we take basic model & change it according to bussiness & com. requirement & company standards, it is called derived model.

eg.

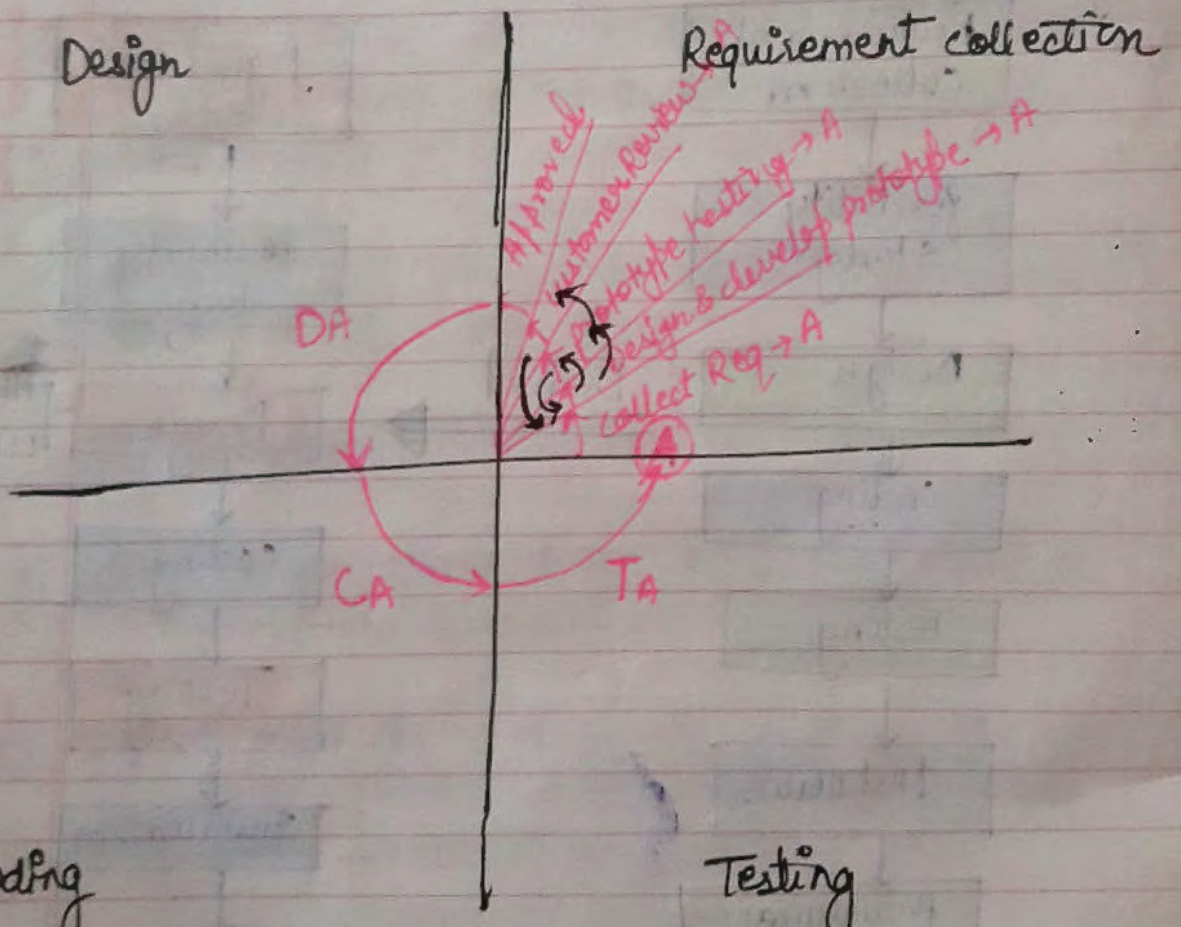


Hybrid Model

* The process of combining or merging more than 1 model into a single model is called as hybrid model.

* Will get the advantage of both the models we have combined together.

Eg: Spiral & prototype model

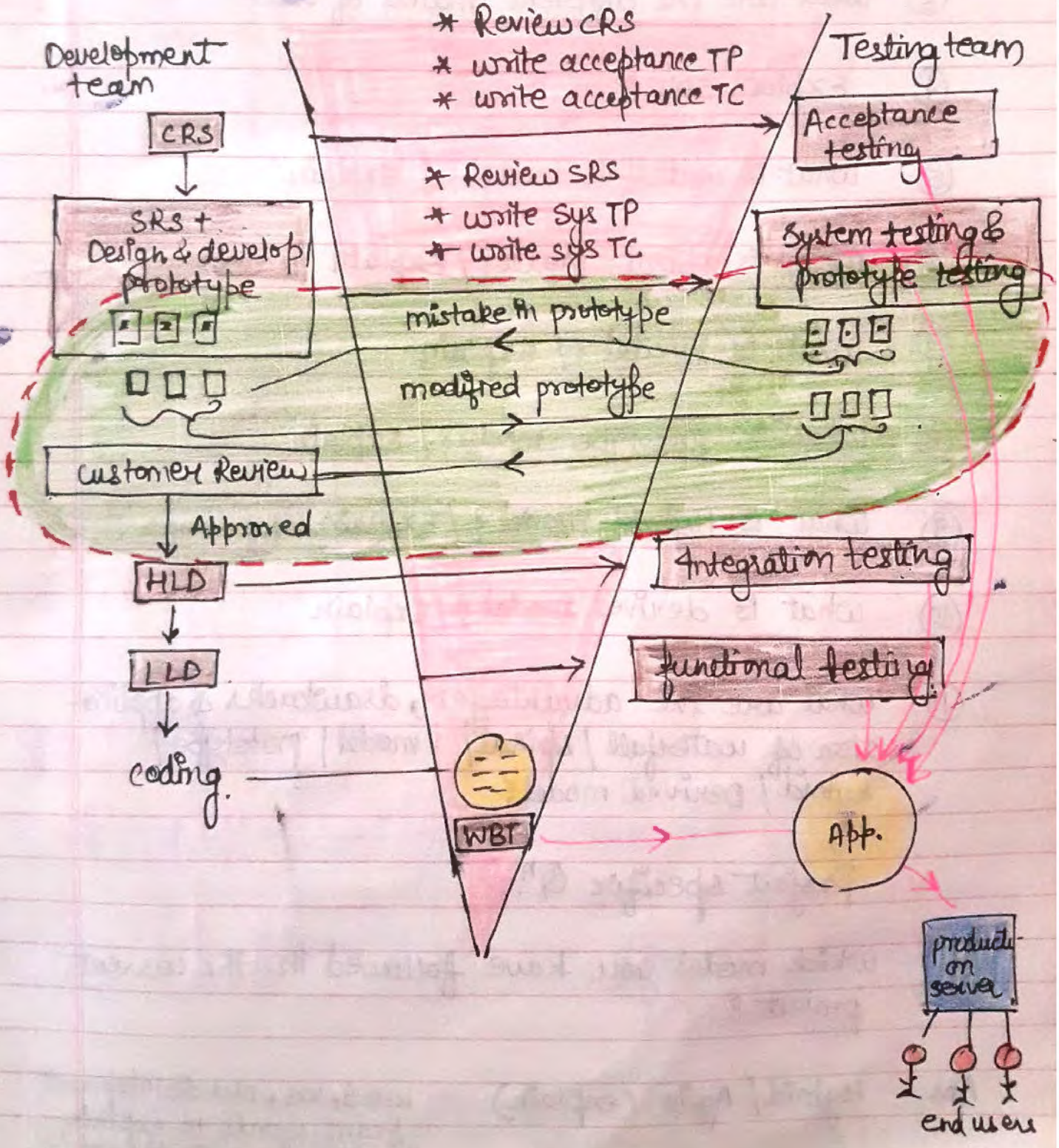


Application:

When ever there is dependency between the ~~models~~ modules & when ever the customer

Is not clear about his own requirement, when customer gives the requirement in stages we go for the combination of spiral & prototype model.

Eg: V & prototype model:



generic Qn

Generic Qn

1. What is SDLC?
2. What are the different stages of SDLC?
3. What are the different models of SDLC?
4. Explain SDLC?
5. What is waterfall model? / Explain
6. What is spiral model? / Explain
7. What is Vmodel? / Explain
8. What is prototype model? / Explain
9. What is Hybrid model? / Explain
10. What is derived model? / Explain
11. What are the advantages, drawbacks & application of waterfall / spiral / Vmodel / prototype / hybrid / Derived model?

Project specific Qn.

1. Which model you have followed in the current project?

Ans. Hybrid / Agile (explain) use I, we, our development team words to explain the same & create the environment.

Software Testing

It a easy job but serious.

eg 1. 96pider student

concept 1
2 x (not attended)
3 x
4
: x

got placed in NOKIA

Tested N8 phone

user

Dinga

Dingi

Call receiving button not working

Switched off the phone

getting call from Dingi

got breakup

Reputation of N8 ↓es

eg 2. Health care app.

♀ → appointment → consultation

♀=♀

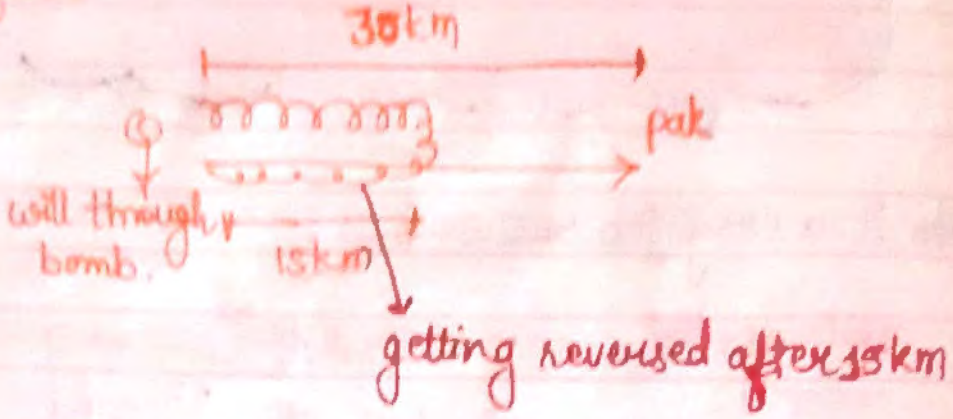
Dolo 650mg 1-1-1 6day (added by ♀)

[Submit] [Cancel]

pharmacy

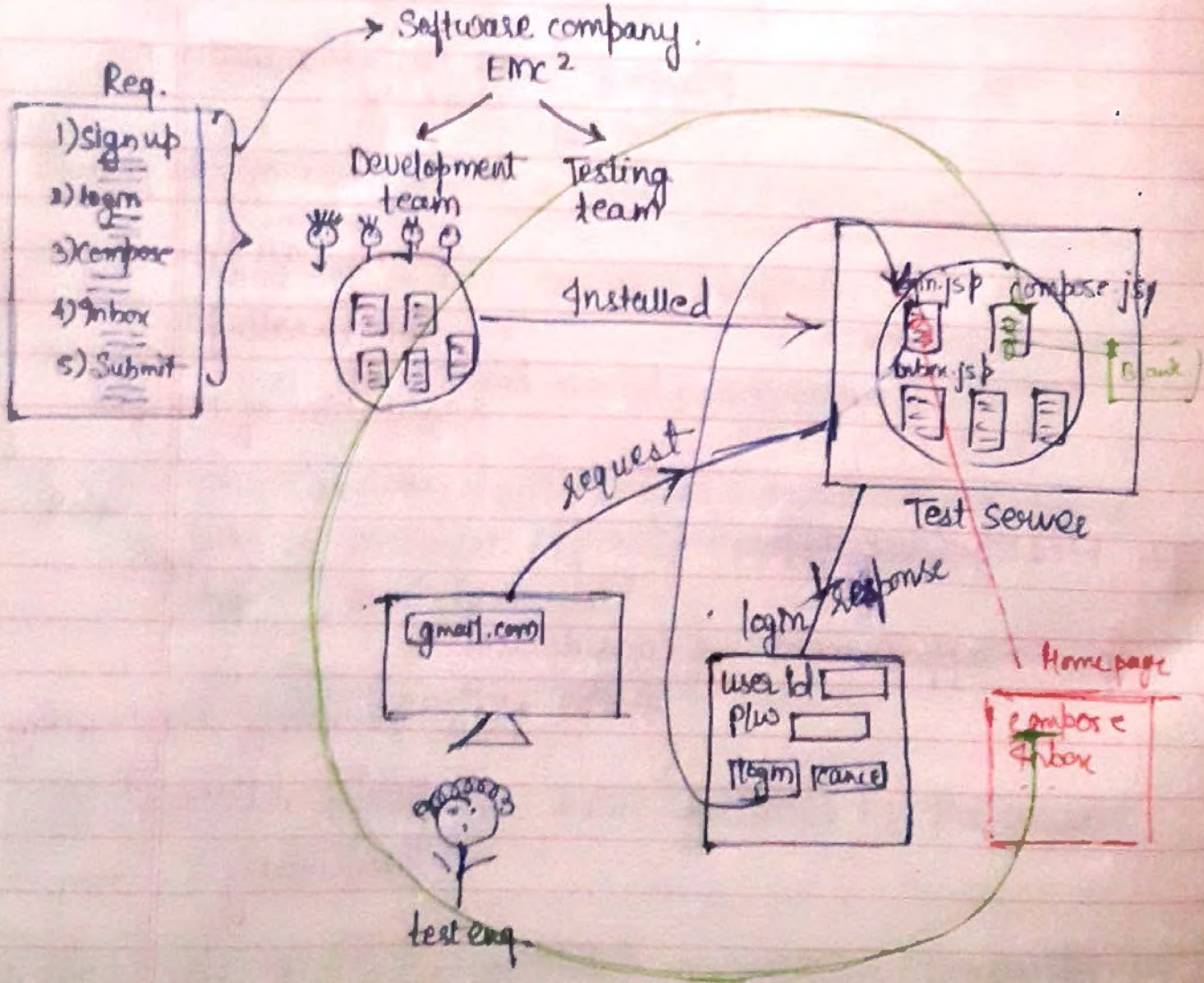
* manual mistake can affect only 1, but form application many people will be affected if its wrong has error

Eg 3. Defence:



Definition 1:

The process of finding the defects in the software is called software testing.



Definition 2:
The process of executing the program with intension of finding the defects is called software testing.

Definition 3:
Checking the functionality of an application according to the customer requirement specification is called software testing.

Types of Software testing

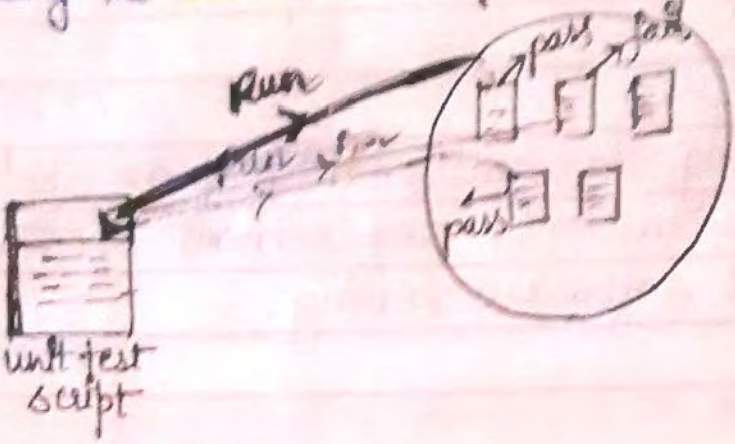
- White Box testing
  - (or) unit testing
  - (or) structural testing
  - (or) transparent testing
  - (or) openbox testing
  - (or) glass box testing
- Gray Box testing
- Black Box testing
  - (or) closed box testing
  - (or) functional testing.
  - (or) Behavioral testing.

White Box testing

* Testing each & every line of the program is called white box testing
* It is generally doen by the developers before giving the software to testing team.
* It is also called open box testing because source code is visible.

* It can be done manually & automatically also.

* We can do white box testing automatically by executing the unit test script.



(If fails, check → the correct prob in respective program → fix)

Black Box testing

* Testing the functionality/ behaviour of the application against customer requirement specification is called black box testing.

* Black box testing can also be done manually as well as automatically by using some automation tools like Selenium, QTP/UFT, Test complete etc.

Suppose developer have developed an application through which 1000 mails can be sent at a time, so in such cases doing manual testing will be tedious & there we go for automation testing.

Have you done white box testing?
In my project I know how developers did white box testing but I haven't done it, as I was busy with in doing black box testing.

* We can do black box testing automatically by executing automation test scripts with the help of automation tools.

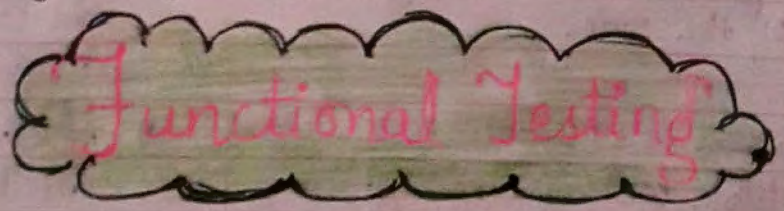
WHITE BOX TESTING	BLACK BOX TESTING
* Testing each & every line of the program is called white box testing	* testing the functionality of the application against requirement specification is called black box testing.
* It is generally done by developers & rarely done by test engineers, if they are good at programming.	* It is doen by test engineers.
* In order to do WBT, we should be good at programming	* In order to do black box testing we need not to know programming.
* To do white box testing we should have knowledge of internal design of the code.	* In order to do black box testing we need not to know internal design of the code.
* Since source code is visible white box testing is also called as open box testing.	* Since source is not visible in black box testing it is also called closed box testing.

Black Box Testing

Testing the functionality of the application against the requirement specification is called black box testing

Types of Black Box testing :

1. Functional testing
2. Integration testing
3. System testing
4. Acceptance testing
5. Smoke testing
6. Exploratory testing
7. Globalization testing
   - → L10N testing
   - → I18N testing
8. Compatability testing
9. Regression testing
10. Performance testing
11. Usability testing
12. Accesability testing



→ (GUI components)

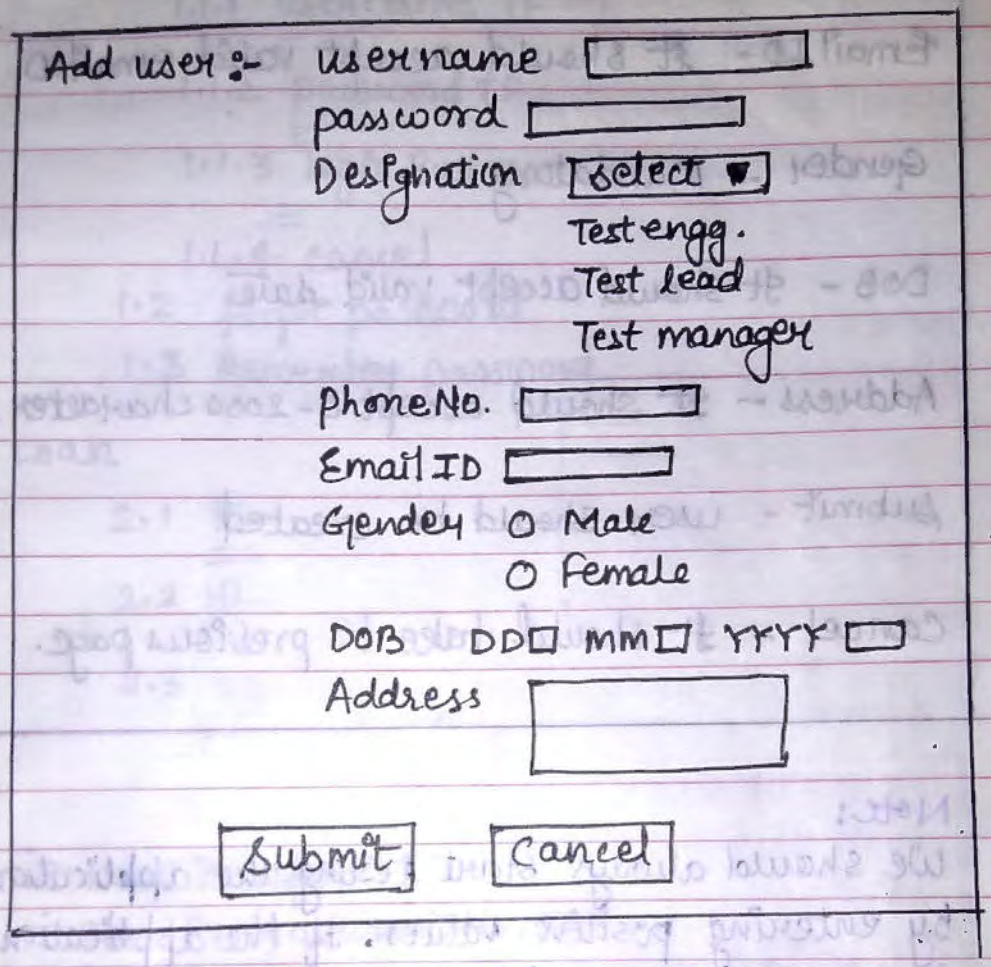
Defination: Testing each & every component of an application throughly against requirement specification is called functional testing

Assignment 1

List GUI components for web based applications.
List GUI components for mobile applications.
List GUI components for stand-alone applications
↓
ms office, excel.

Assignment 2

↓ login as admin



SRS

Add user : on click Add user link, Add user page should be displayed & the following components should be displayed username text field, password textfield ...

Username TF : - It should accept only 4-16 character
- special character should not accept.

Password TF : - It should accept only 8-24 character
- It can accept digits & special character

Designation - Mandatory

phone No - It should accept valid 10 digit No.

Email ID - It should accept valid email ID

Gender - Mandatory

DOB - It should accept valid date

Address - It should accept 0-2000 character.

Submit - user should be created

Cancel - It should take to previous page.

Note:
We should always start testing the application by entering positive values. If the application is working for valid data than only we should test for invalid data. If the application is not working for one of the invalid data still we can test with some more invalid data.

CBO - SRS

1. Welcome
    - 1.1 login
        - 1.1.1 username TF
        - 1.1.2 password TF
        - 1.1.3 login Button
        - 1.1.4 cancel
    - 1.2 forgot password
    - 1.3 Remember password

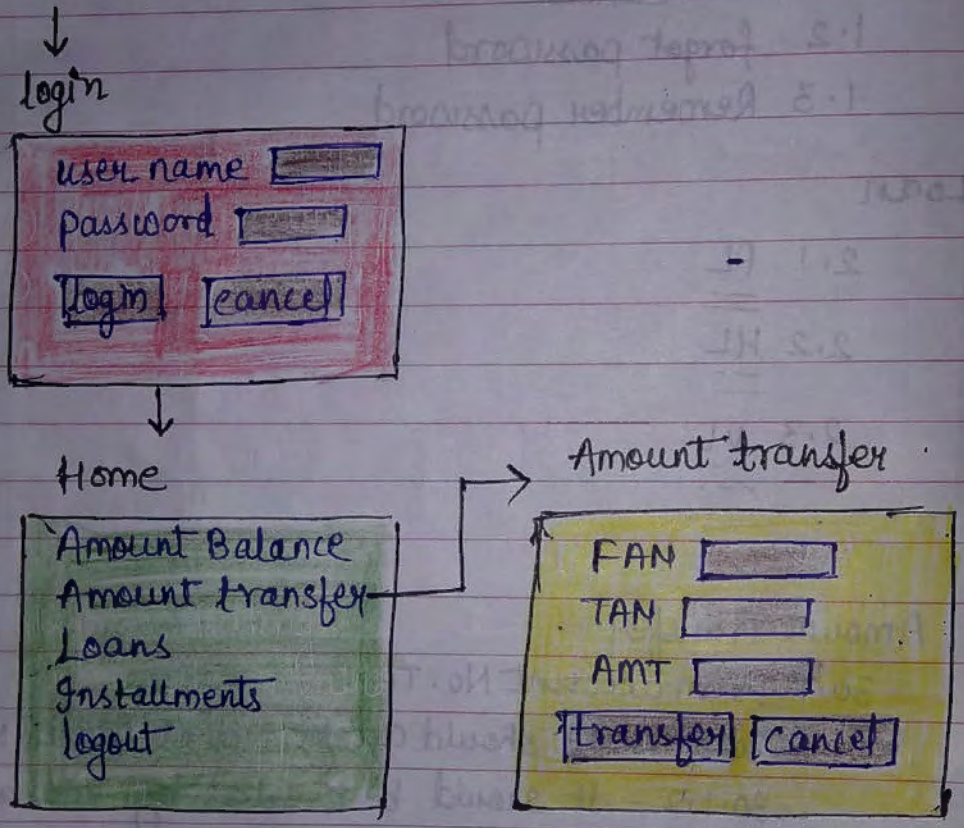
2. Loan
    - 2.1 PL
    - 2.2 HL
    - 2.3 VL

...

30 Amount Transfer
- 30.1 From account No. TF
    - 30.1.1 - It should accept 10 digit Integer No.
    - 30.1.2 - It should be created by manager
- 30.2 ~~TA~~ To account No TF
    - 30.2.1 -
    - 30.2.2 -
- 30.3 Amount TF :
    - 30.3.1 It should accept only +ve integer b/w 100 - 50,000
    - 30.3.2 It should not accept more than the balance.
- 30.4 Transfer Button
- 30.5 Cancel Button

Why the requirement document should be numbered?

- There will be clearity in the requirement & easy to understand.
- It becomes measurable.
- Easy to communicate with development team & customer.



Test senerious:

① FAN
- 10 digits ✓
- 9 digits ✗
- 11 digits ✗
- alphanumeric ✗
- blank ✗
- space ✗
- decimal ✗
- -ve no. ✗
- special character ✗

2). TAN - same as FAN
both account should be different.

3). AMT - 100 ✓
90 ✗
60,000 ✗
more than balance ✗
-value ✗
decimal ✗
blank ✗
space ✗
0 ✗
50,000 ✓
special character ✗
1001 ✗
10001 ✗
99 ✗
5000+100 ✗
0100 ✓
60,000 - 1000 ✗
1000/2 ✗
alphanumeric ✗

wrong scenarios

no special characters

Any kind of testing we can do in 3 ways.
① over testing
② under testing
③ optimised testing.

Over Testing:
Testing the application with those scenerios which doesnot make sense is called over testing. By doing over testing we loose lot of testing time.

Under Testing:

Testing the application with insufficient sets of data is called as under testing. By doing undertesting we miss lot of defects.

Optimised Testing:

Testing the application with those scenarios which really make sense is called optimised testing.

We have two types of testing

1. Positive testing
2. Negative testing.

Positive testing:

Enter the valid or expected data & test the application is called as positive testing.

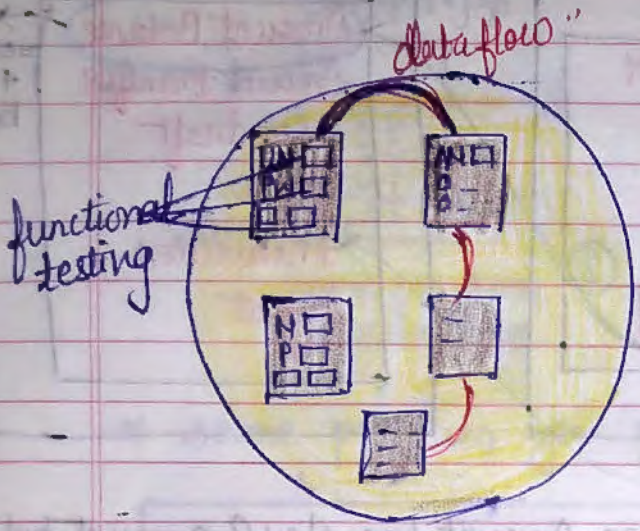
Negative testing:

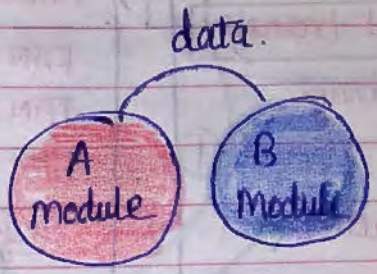
Enter the invalid or unexpected data & test the application is called as negative testing.

Note:

As a test engg. we should not assume or propose the requirement.

Integration Testing





Defination:
Testing the data flow between two dependent modules or features is called Integration testing.

Example:

1). Login as a user A in facebook
2). Sent request to user B
3). Logout as A & login as user B
4). check weather request of A has come or not
5). Accept the request & logout B.
6) login as A & check weather friend request accept notification has come or not.

If these messages are received in both the accounts means there is data flow happening & this is just a small eg. for integration testing.

[Yes pass
No fail

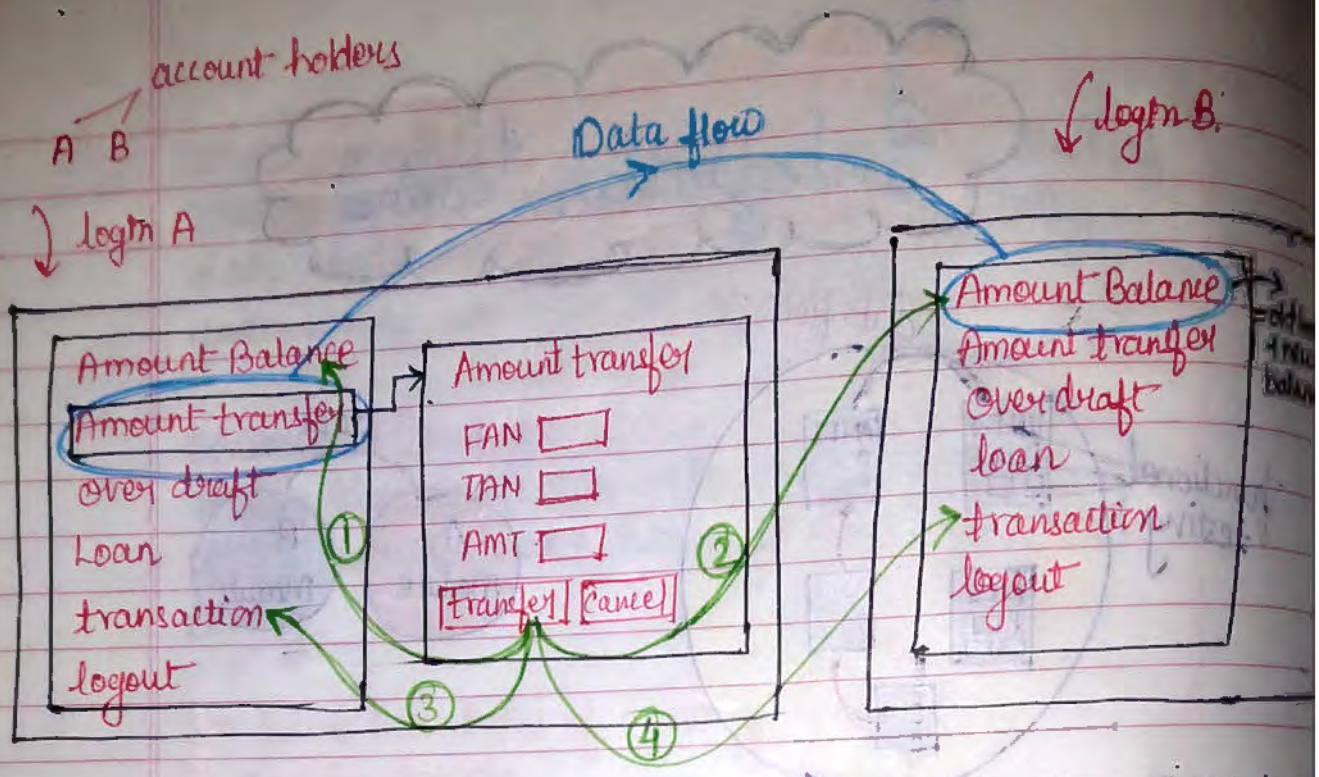


fig: Data flow between Amount transfer & amount balance

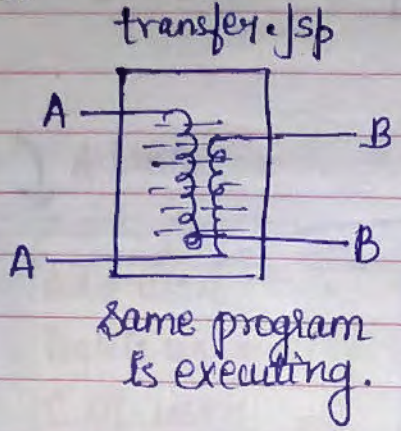
Senerious
A transfers to B
1). check B's balance
2). check A's balance
3). check transaction details of B
4). check transaction details of A.

2 Approaches

I	II (Best) → In less time, all senerious covered.
i) login as A & do transfer.	i) login as A & do transfer
ii) logout as A	ii) check A's balance & B's transaction details
iii). login as B & check B's balance & transaction details.	iii) logout as A & login as B
iv) logout as B & login as A.	iv) check B's balance & B's transaction details.
v) check A's balance & transaction detail.	

Is there any need to check that, If B transfer the amount to A, Is it working fine?

- No.



Same program is executing.

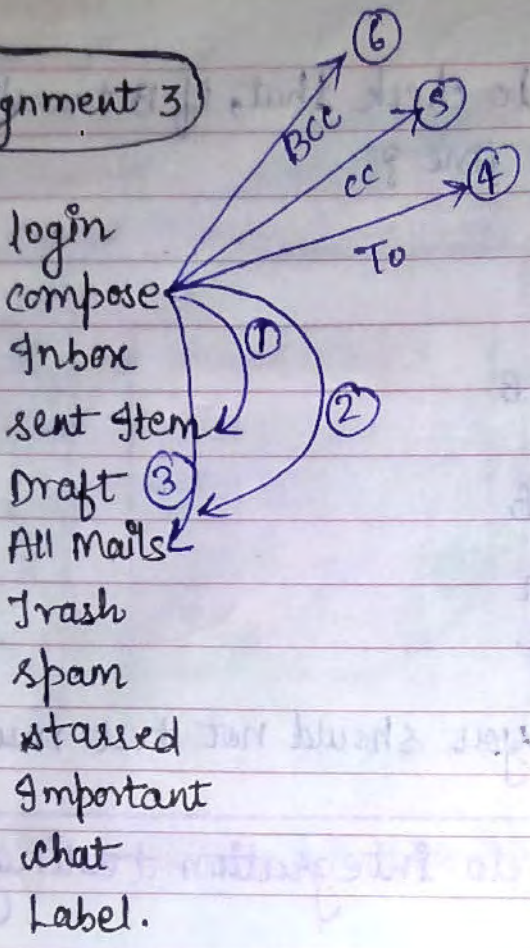
* same program you should not test multiple times.

Procedure to do Integration testing

1. Understanding the application is very important. we should understand how each & every feature works in depth.
2. we should also understand, how each & every features are related.
3. Identify all possible senerious.
4. Prioritise the senerious.
5. Document the senerious.
6. Test the application by executing identified senerious.

If the application is not working according to the senerious I will be considering it as a defect & report to developers.

Assignment 3



List all the senerious related to compose feature.

Compose - sent Item

1. Login to gmail as user A
2. click on compose link
3. Enter valid data into all the fields
4. click on sent button
5. click on sent Item link

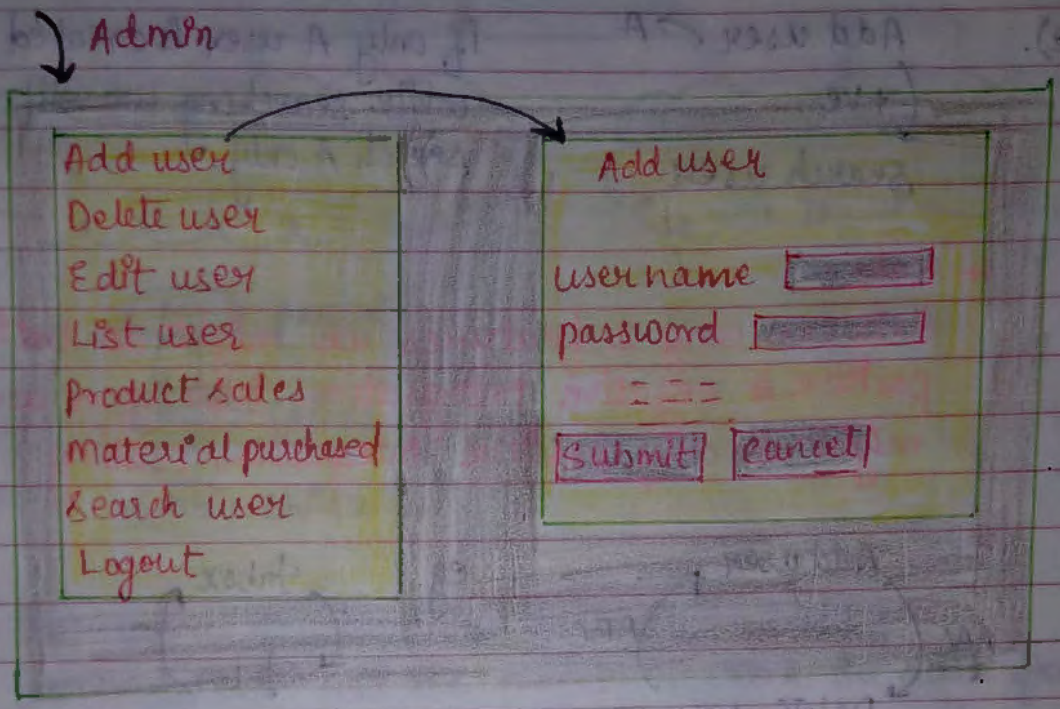
Expected Result.

Sent mail should be listed in sent Item page.

Actual result.

Assignment 4

- List 100 senerious for gmail.

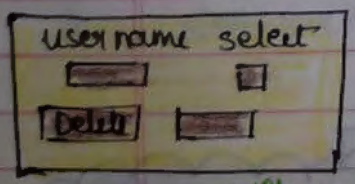


Senerious

Add user →
- Delete user
- Edit user
- list user
- search user

List user →
- Search user ✗
  No data How is happening.

Delete user →
- list user
- edit user
- search user
- add user



Edit user →
- list user
- Search user
- Delete user
- Add user

ABC → xyz (username edited.)
2nd 1st
-ive IT +ve IT

* Don't jump from feature to feature. Stay on 1 find all the senerious than move to next.

ABC → xyz

1). Edit user

+ive IT ← xyz / ABC → -ive IT

search user

2). Add user → A

+ve

search user

If only A user is created, while searching we will search A only.

* Note:

Between any 2 features we might do both positive & negative integration testing or we might do only positive integration testing.

Add user

ABC ← / ABC

Delete user

eg. Inbox

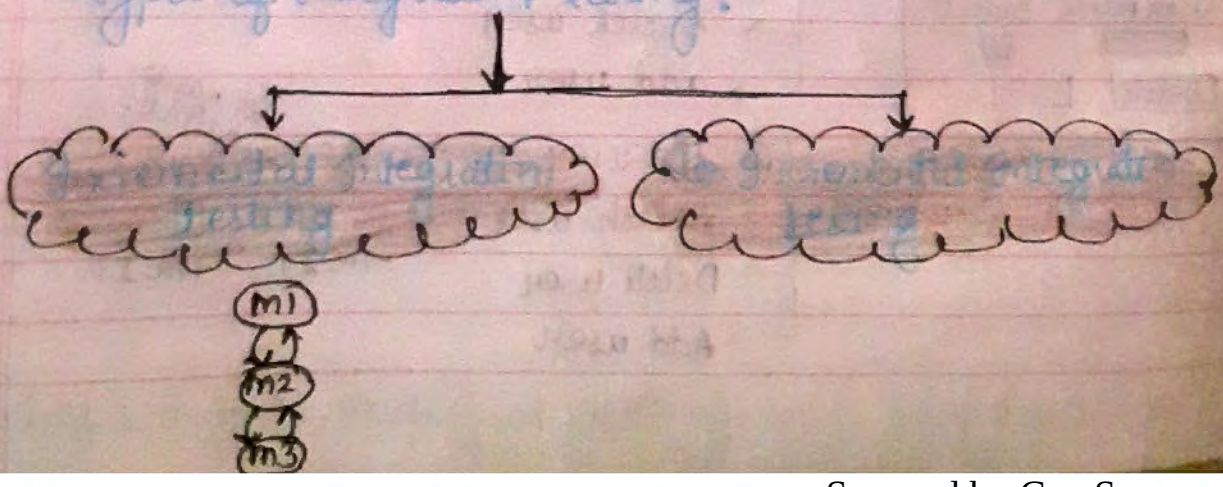
spam

compose

sent Item

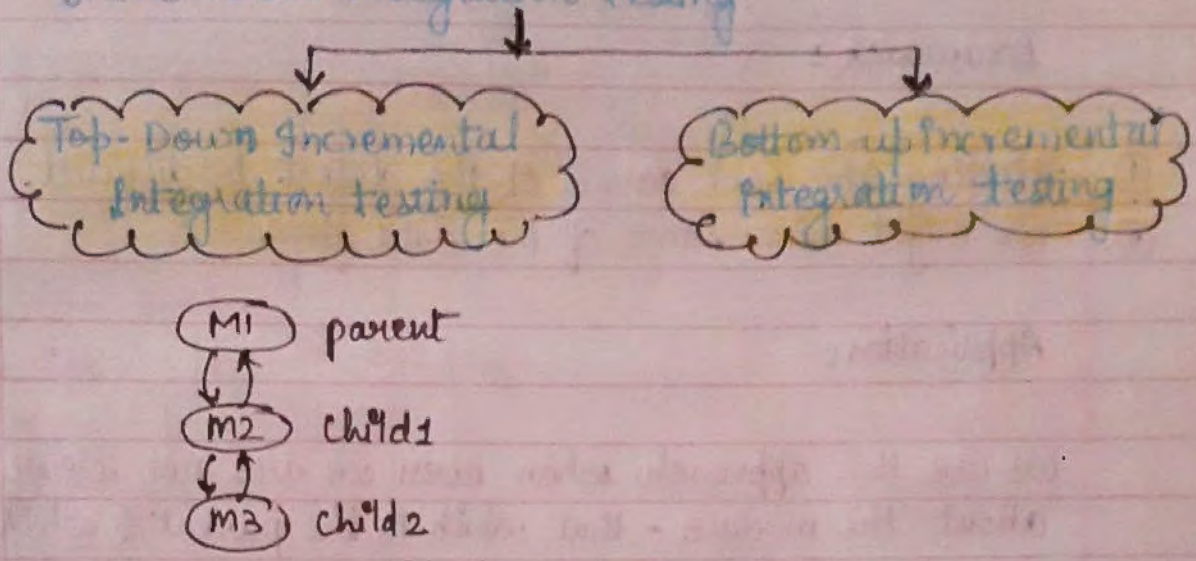
Between two features we might test data flow in both the way or only 1 way, It varies from features to features.

Types of Integration testing:



1. Incremental Integration testing:

Incrementally add the modules & test data flow between the modules is called Incremental Integration testing.



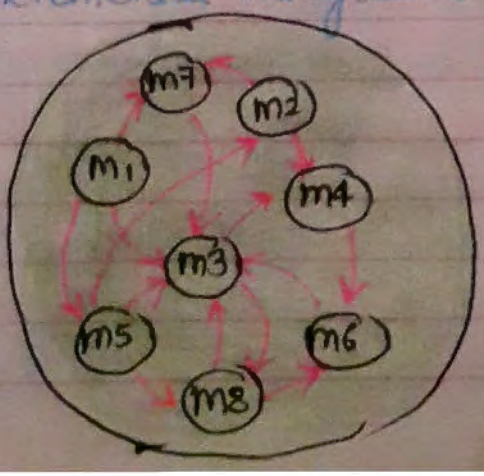
(i) Top-Down Incremental Integration testing:

Incrementally add the modules, check the data flow between the modules but make sure that newly added module is the child of previous module.

(ii) Bottom-up Incremental Integration testing:

Incrementally add the modules, check the data flow between the modules but make sure that module you have added is the parent of previous one.

2. Non-Incremental Integration testing: [Big Bang approach]



* Here we combine all the modules at a sout & check the data flow between the modules.

* It is also called big bang approach.

Drawback:

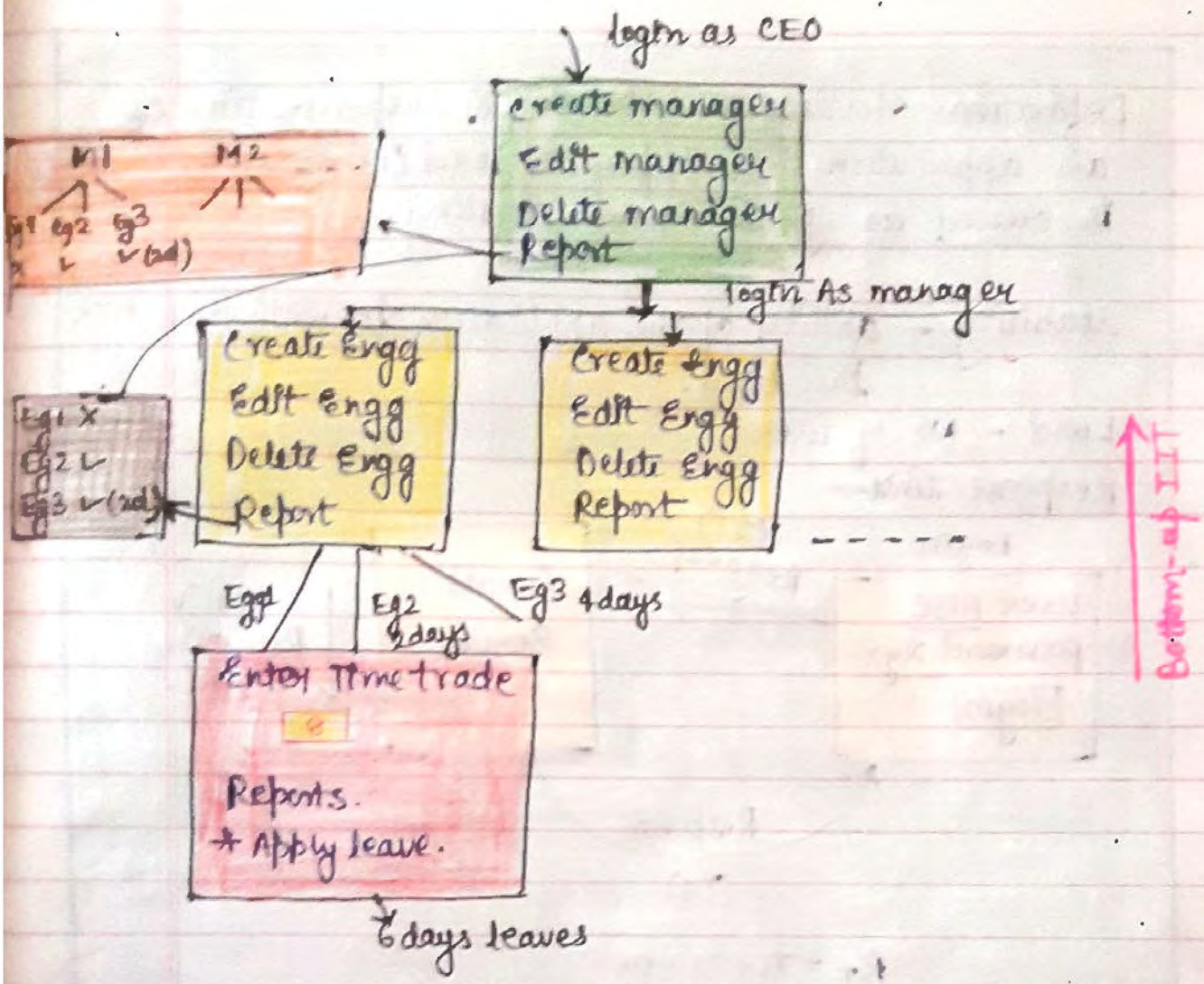
① Finding the root cause of the defect is difficult.
② We might miss some of the data flow.

Application:

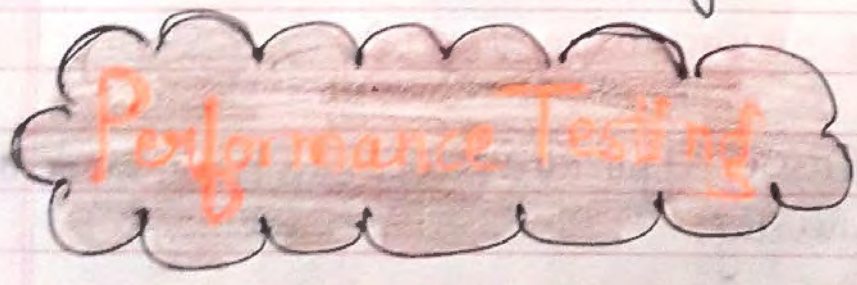
We use this approach when ever we are not clear about the module - that which is the parent & which is the child.

Ans. → Defination IT
→ Face book eg
→ Procedure
→ Types.

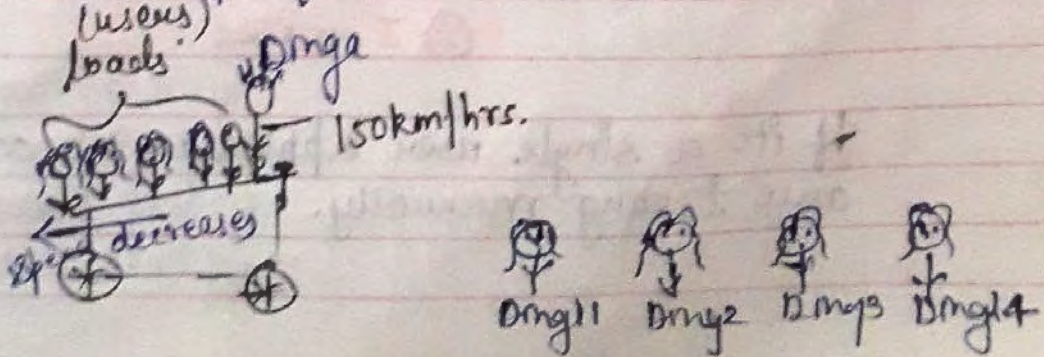
* when we can do both top-down as well as bottom-up testing it is called sandwitch Incremental Integration testing.



Eg: For Top-down, Bottom-up & sandwitch Incremental Integration testing



* Non functional / Scalability testing - because here Requirement Specification is not needed.

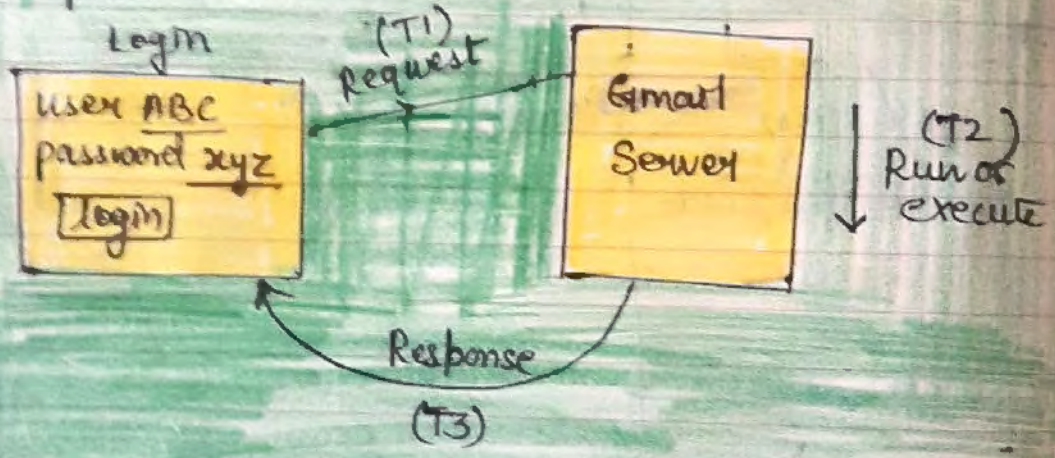


Definition: Testing the stability & response time of an application by applying load (no. of users) is called as performance testing.

Stability - ability of the application to withstand the load.

Load - No. of users

Response time -



$R_T = T_1 + T_2 + T_3$

How to do performance testing?

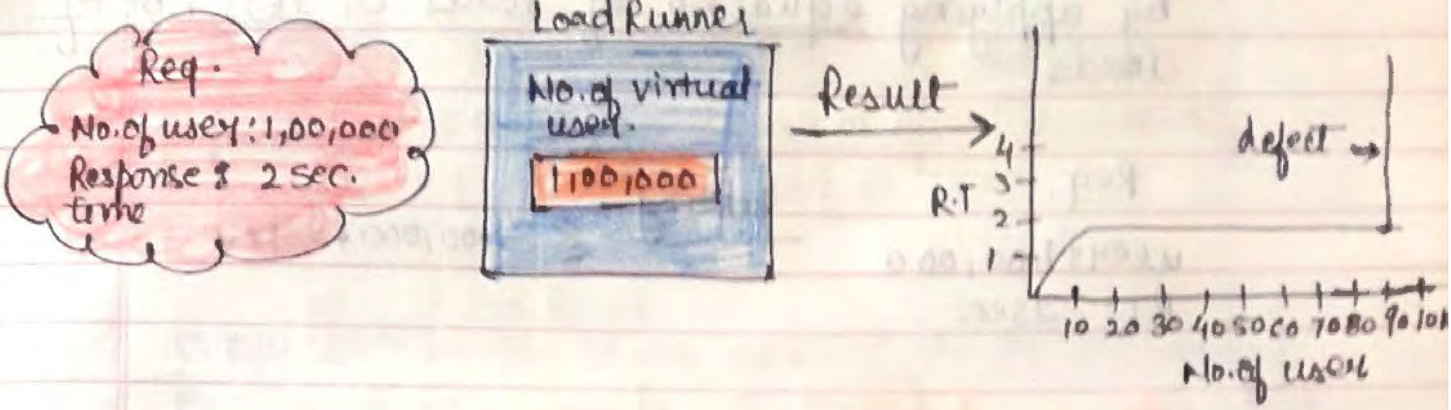
For a multiuser application we can't do performance testing manually.

We always do Performance testing by using some automation tools - ① load runner
② J-meter

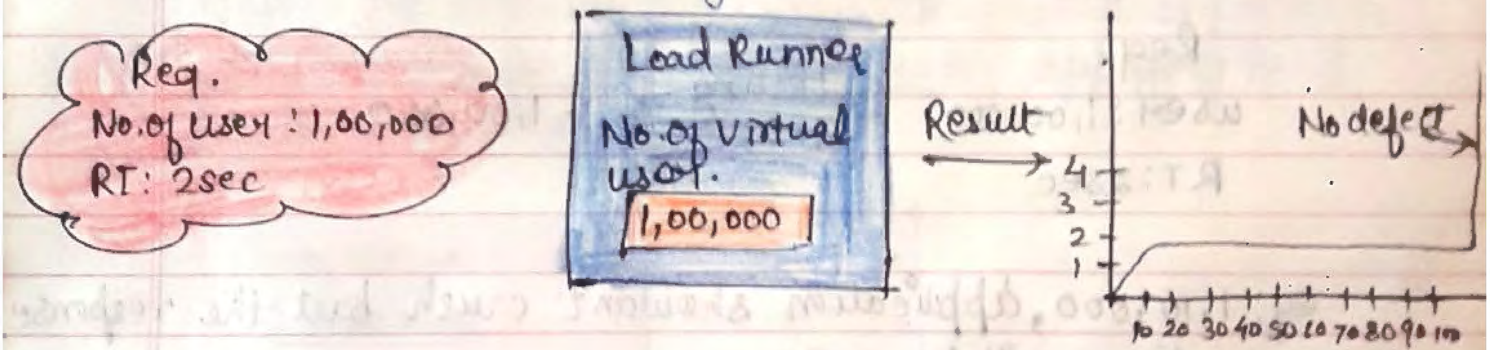
If its a single user application we can do performance testing manually.

How to do performance testing using load runner?

If its a single user applicati



at 80,000 users the threshold value increased that means the response time of the application is high even before 1 lakh users. Hence it is a ~~defect~~ defect & we raise it as bug to Development team.



The expected result/actual result has to be with in 1 lakh user & response time has to be in 2sec. After 1 Lakh users if the Response time yes it is not a defect

Types of performance Testing

- → Load testing
- → stress testing
- → volume testing
- → Soak testing

Load testing:

Testing the stability & Response time of an application by applying equal no. of loads or less no. of loads

Req.
users: 1,00,000
RT: 2sec.

testing → ≤ 1,00,000, 2sec..

Stress testing:

Testing the stability & Response time of an application by applying more no. of users is called as stress testing.

Req.
user: 1,00,000
RT: 2sec

testing → > 1,00,000

eg: 1,10,000, application shouldn't crush but the response time will be more.

eg: 2,00,000, application shouldn't crust it will through an error.

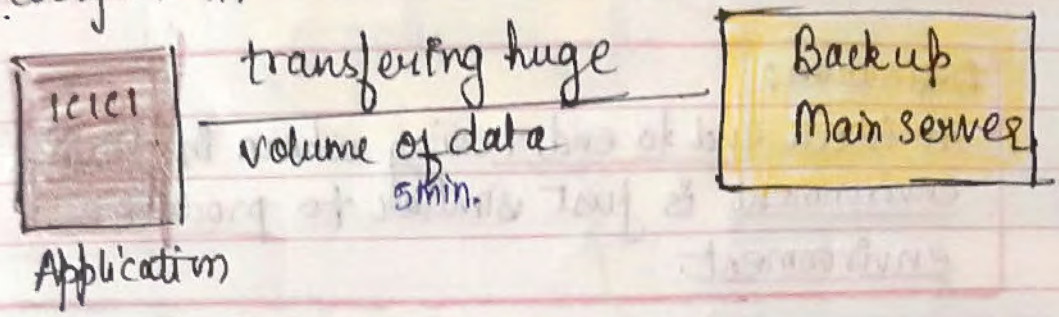
The application should not crush by applying more no. of users & if it crush it will be raised as a defect.

Volume testing:

Testing the stability & response time of an application by transfering huge volume of data

from 1 place to another is called as volume testing.

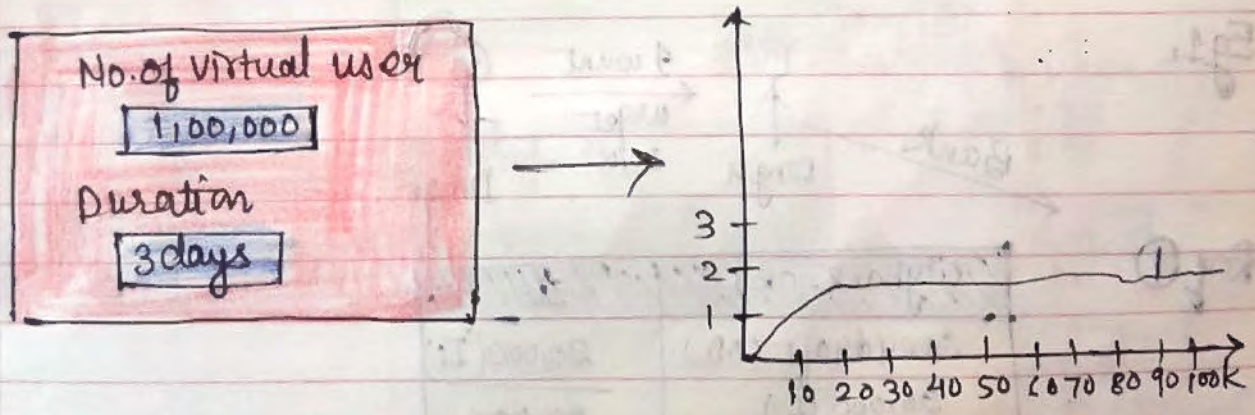
every 24th hr



every 24th hr. the data needed to be backedup it should take 5mins if it takes more than that, then it's a defect.

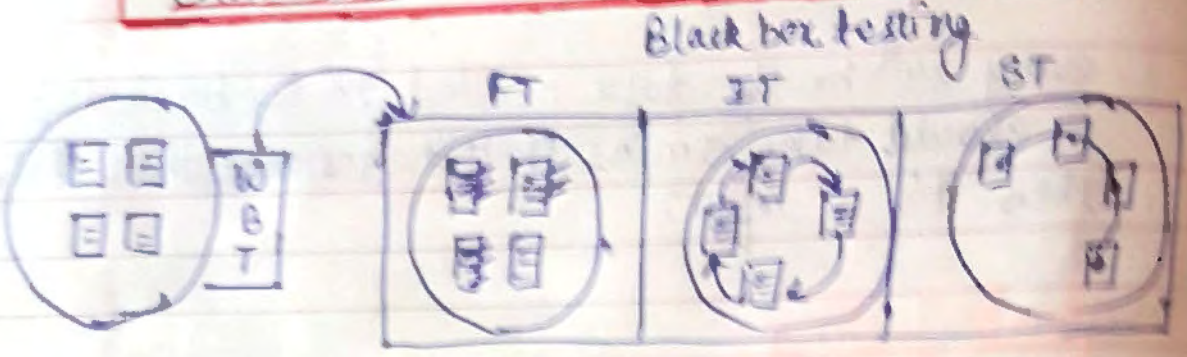
Soak Testing:

Testing the stability & Response time of an application continuously for a period of time by applying load is called Soak testing.



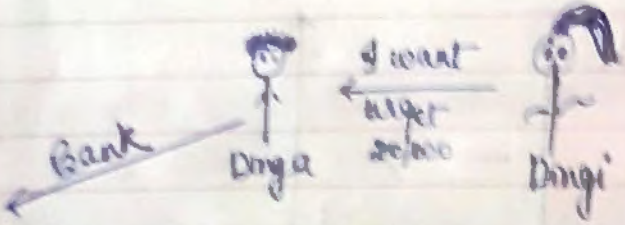
System Testing

Definition:
* It is an end to end testing where in testing environment is just similar to production environment



End-to-End testing:
Navigate through all the features & check whether end feature is working or not.

Eg:



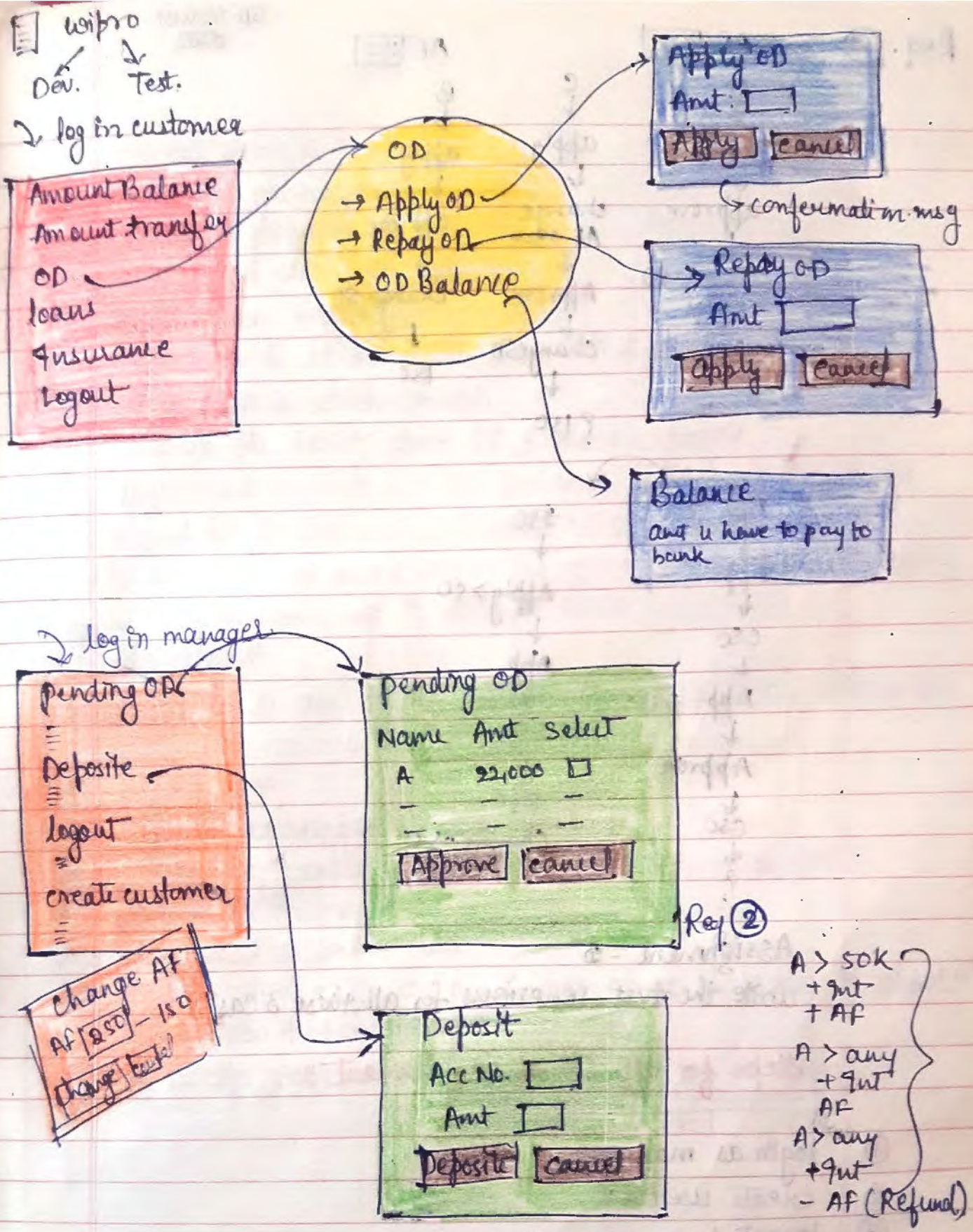
Req ①

citybank - cc	sbi
Overdraft (OD)	20,000 (IT)
20,000 (I)	20,000
20,000	400
400 (2%)	(2%)
250 (AF)	
20,650	
-20,650	
00,000	

for 1 month

OD manager → Approve

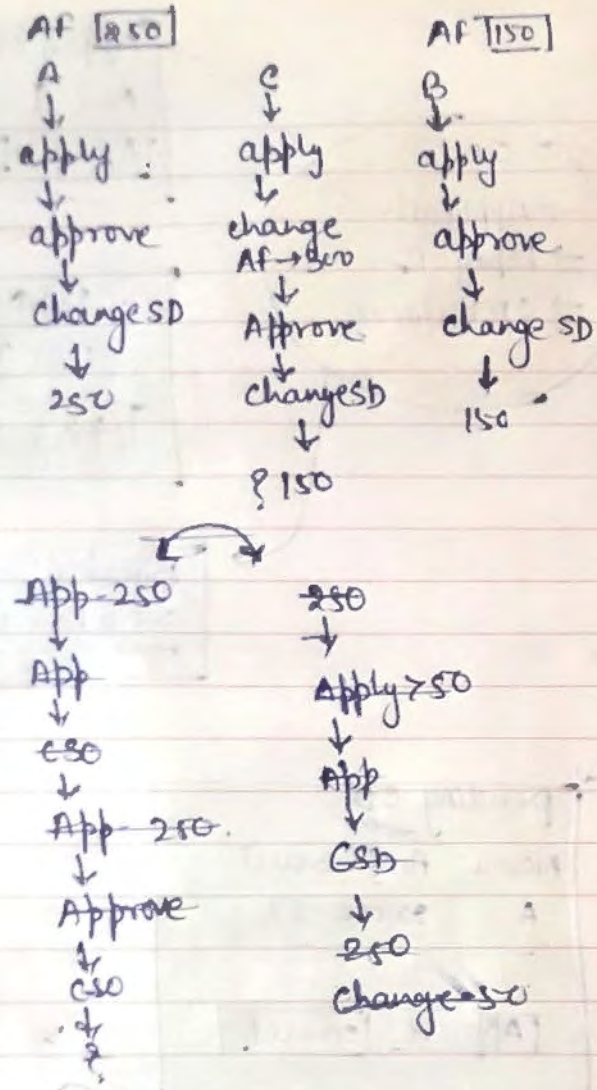
* AF: activation fee.



Name of Req:

Change activation fee :- when ever the OD manager changes the activation fee, from that moment onwards if anybody apply for OD the latest activation fee should be charged.

Req. (3)



Assignment - 5

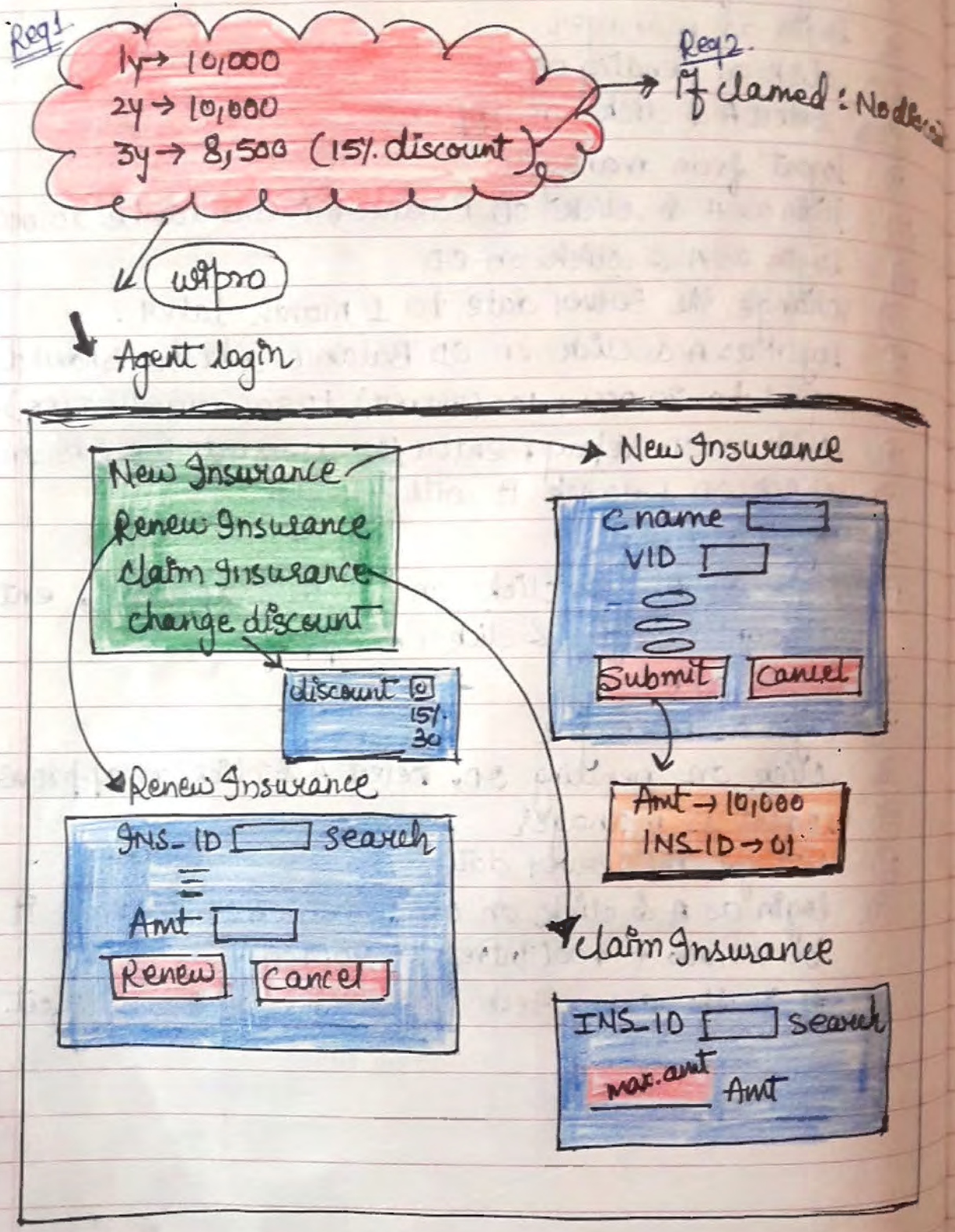
write the test seneriors for all these 3 cases

Steps for the 1st requirement:

1. login as manager.
2. create user A
3. logout from manager.
4. login as customer A
5. click on OD & then apply for OD by entering amt = 20,000 & click on apply button
6. conformation message should be received, that you have successfully applyed for the OD of amt = 20,000

(7). logout from A
(8). login as manager
(9) click on pending OD
(9)(10) Select A & click on approve
(11) logout from manager
(12) login as A & click ~~OD Balance, It should be 20,000~~
(12) ~~login as A & click on OD~~
(12) change the server date to 1 month later.
(13). login as A & click on OD Balance which should be equal to 20,000 + 400 (Intrest) + 250 (activation fee) = 20,650.
(14). click on OD repay, enter the amount & click on apply.
(15) check OD balance it will be zero.
(16) logout as A
(17) login as A and click on OD then apply OD, enter the amount = 20,000 & click on apply.
(18) logout as A
(19) login as manager
(20) click on 'pending OD, Select A & click on approve.
(21) logout as manager
(22) change the server date
(23). login as A & click on OD & then on OD Balance it should be 20,000 + 400 (Intrest) = 20,400/-
If its the case, test is passed else its a defect.

Eg2. ICICI - CRS



Scenerios for Req 1:

1. I will login as agent, click on new Insurance.
2. New Insurance page will be displayed where I will enter customer name in C name field & will fill all other details & click on submit button.
3. Clicking on submit button I must get the conformation message displaying amt = 10,000 & INS-ID = 01.
4. Now I will logout as agent & will change the server date to next 1 year.
5. I will again login as agent and click on Renew Insurance. option.
6. Renew Insurance page will be displayed where I will search INSID & it will display amt = 10,000 & I then I will click on Renew button, send logout as agent.
7. Now I will again change the server date to next 1 year.
8. and will login as agent again.
8. Now I will click on renew insurance.
9. Renew Insurance page will be display where I can search INS-ID as 01 & amt displayed after it will be 8,500 in amt field.

If this is the case, then test is passed else it has a bug.

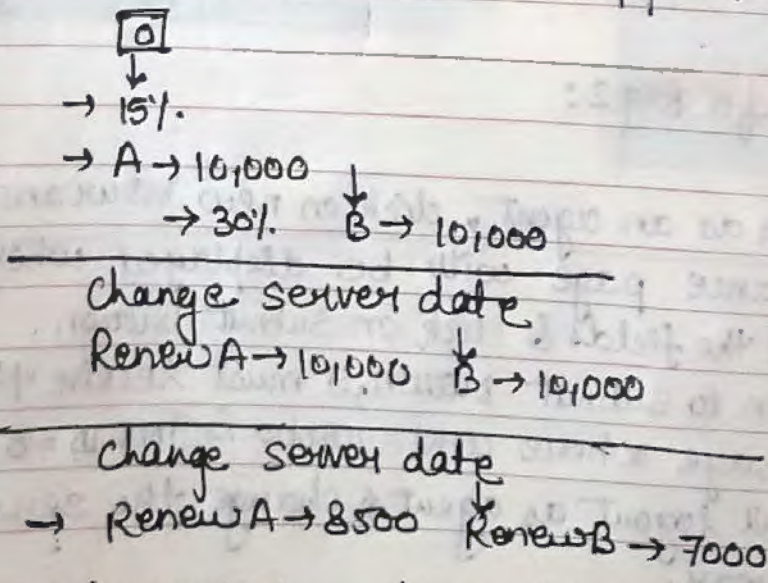
Scenerios for Req 2:

1. I will login as an agent, click on new Insurance.
2. New Insurance page will be displayed where I will enter all the fields & click on submit button.
3. Clicking on to submit button, I must receive the conformation message where amt = 10,000 & INS-ID = 01
4. Now I will logout as agent & change the server date to next 1 year.

5. Now I will again login as agent, will click on renew Insurance optin. ~~where~~ I will get the Renew Insurance page where I can search the INS_ID = 01 & amt ~~must be~~ displayed ~~equal to~~ must be equal to 10,000.
6. Now I will click on claim optin & I will get claim Insurance page, in which I can search Ins-ID as 01 and here maximum amount will be displayed.
7. Now I will logout as agent & change the server date to next 1 year.
8. And I will again login as agent and click on Renew Insurance optin. ~~where~~ I will get Renew Insurance page where I can search INS-ID as 01 and amt displayed in amt field must be equal to 10,000.

If this is the case then the test is passed else it's a bug & we must report it to the development team.

Requirement 3.
when ever the Insurance agent changes the discount from that moment onwards if anybody create's new Insurance for them latest discount should be applied.



* 2 senerious means 2 end-to-end testing.

- from 1 login to logout its 1 scenario.
- In interview give eg. of complex scenarios.

Scenarios for Req3:

1. I will login as a ~~manager~~ agent and will click on change discount option & there I will change discount = 15%. on next I will click on New Insurance, new Insurance page will be displayed where I will fill all the field for A & click on submit.
2. Now I will ~~change~~ click on change discount & will change it to 30% & will click on new Insurance, and fill all the field for B & click on submit.
3. Now I will logout as agent & change the server date.
4. I will again login as agent & click on renew Insurance where I will get renew Insurance page, there I will search INS_ID for A and amount that should be displayed in Amt field should be equal to 10,000 &
5. ~~I will again click~~ will click on Renew button. This will renew the insurance for A for the 2nd year.
5. I will again click on renew Insurance, ~~renew~~ option, renew Insurance page will be displayed, where I will search INS_ID for B and amount displayed in Amt field should be equal to 10,000 & then will click on renew optim. This will renew the B customer's Insurance for the 2nd year.
6. Now I will logout as agent & will change the server date to next 1 year.
7. I will login as agent & repeat the same process to renew the Insurance for A & B for the 3rd year where amt field must display Amt = 8500 for A & Amt = 7000 for B.

If this is the case then test is passed else failed.

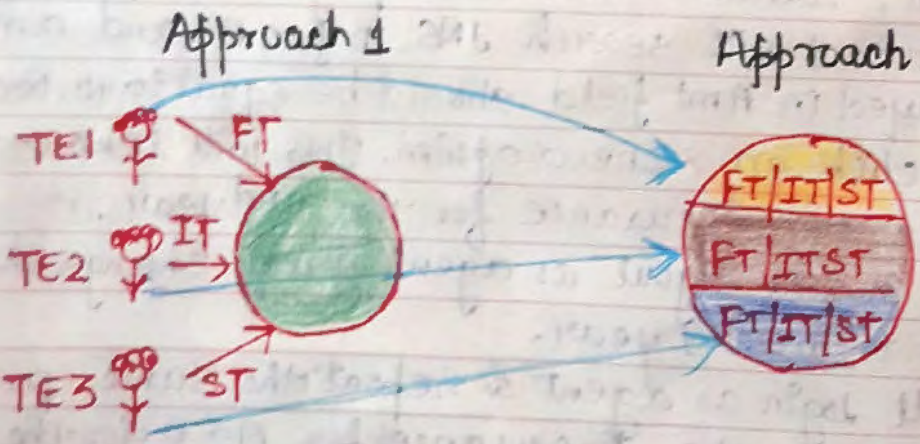
Assignment 6 write 1 eg. for system testing.

Work allocation

Suppose we have an application (eg. gmail) & it has following features.

Signup	login	compose
Inbox	All mail	sent Items
Spam	Draft	Trash
Important	Starred	Labels
Chats	Help	Settings
Outbox	logout	forgot P/w
Remember	profile	Contacts
Calender	folder	Themes
Archive	Social	promotion

Now I have to assign these features to different test engineers (suppose TE1, TE2 & TE3)

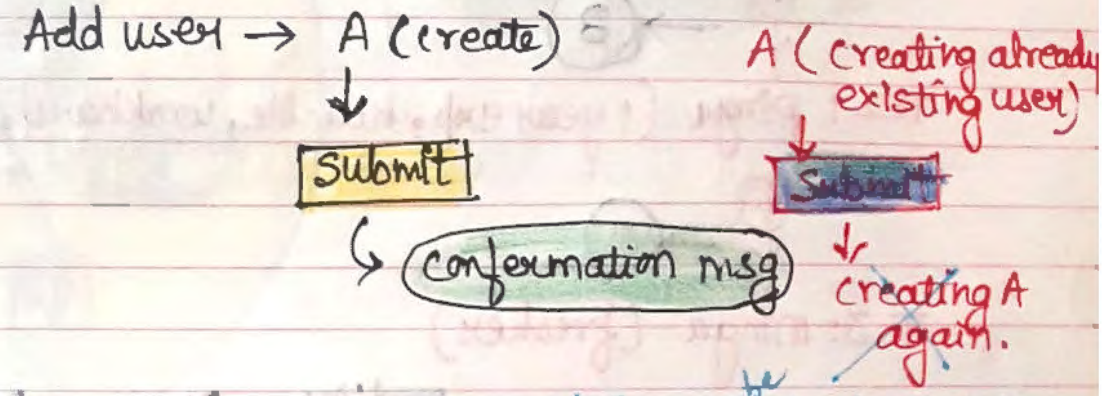


Which approach is best?
Ans. II

Drawback of I approach *
① Everybody has to understand the entire application

② TE1 may think it is a part of IT & will skip that part & TE2 may think it is a part of FT & may skip the same part. (confusing

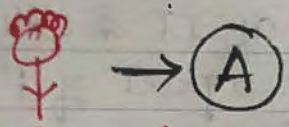
eg:



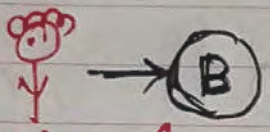
while allocating the work, similar modules interlinked modules will be grouped together.

A	B	C
Compose	Signup	Chat
Inbox	login	Calender
Sent Item	logout	Contacts
Draft	forgot p/w	help
All mails	remember p/w	theme
Spam	settings	folder
Starred	Logout	label
Important		social
Archieve		promotion.
more & main features.		not interdependent much

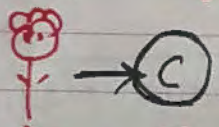
Dividing the work in this way, may also lead to more work for 1 group & less for another but still we will go this way.



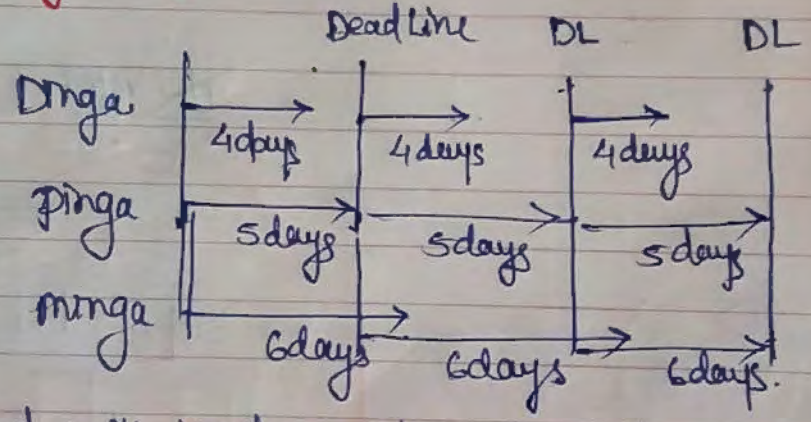
TE1: Dinga (4+years exp, talended (but he escapes ⇒ follow up is required)



TE2: Pinga (1 year exp, humble, workhard, trusted)



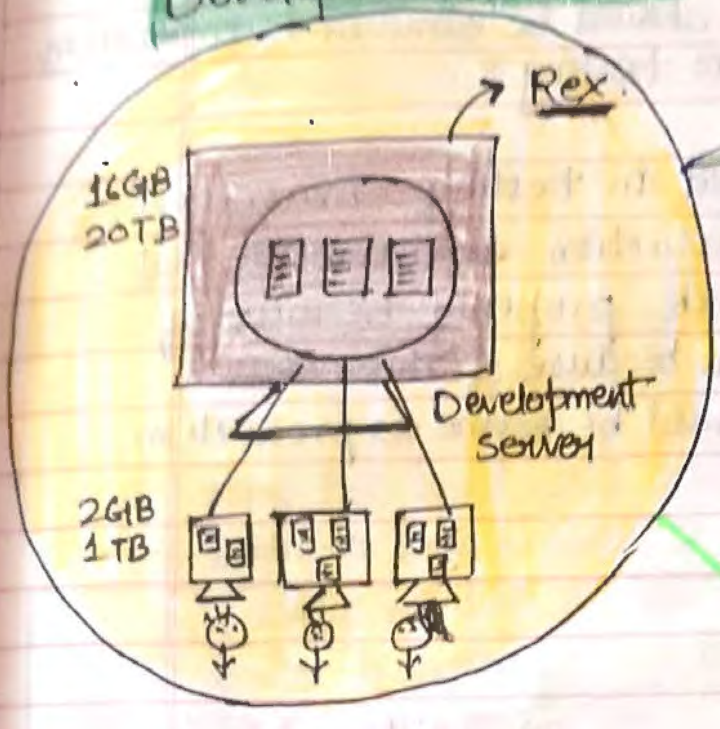
TE3: Minga (fresher)



Lead will keep on observing, he will not instantly react, when he see that minga gets late for the 3rd DL also ~~need~~ & he works regularly mean he he really has lot of work & then he will adjust for it.

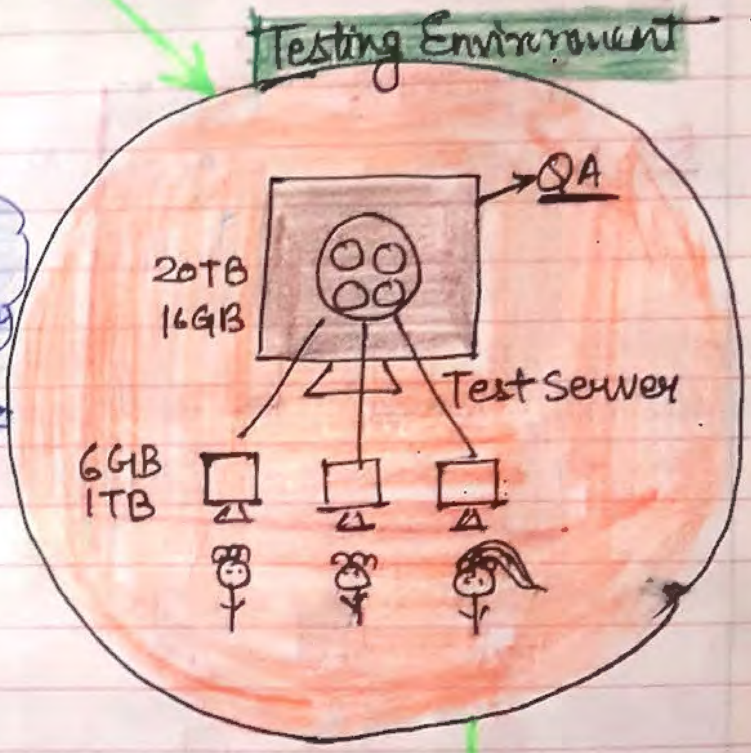
He comes to know that Pinga is really working superfine & is talented also he will as Dinga - that Dinga you please do pinga's work he is not working fine & will ~~give~~ ~~the~~ assign Dinga's work (critical section) to pinga as he don't require any followup.

Development Environment



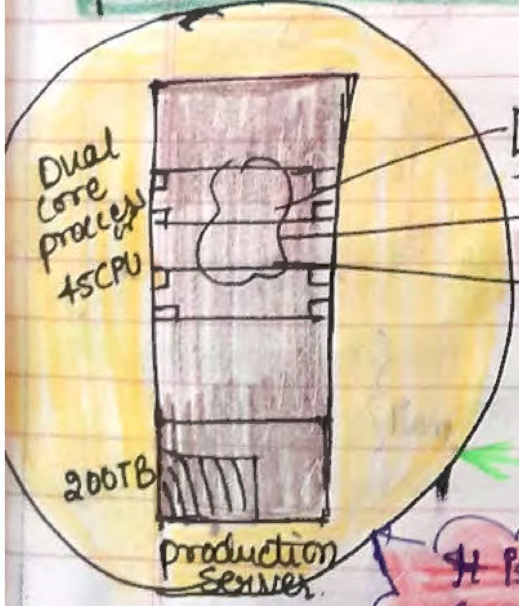
It is a setup used to develop the software, which consist of s/w, H/w & network.

Testing Environment

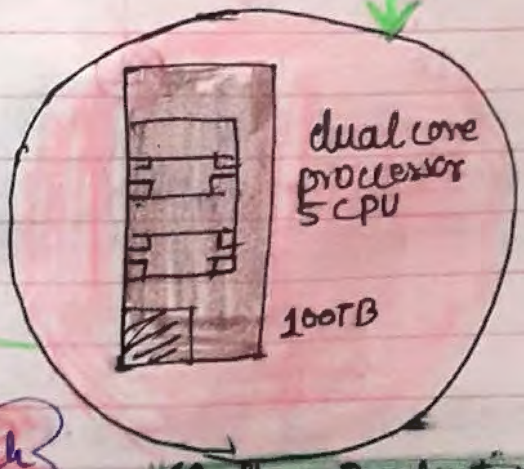


It is a setup to test the s/w, which consist of s/w, H/w & N/w

Production Environment



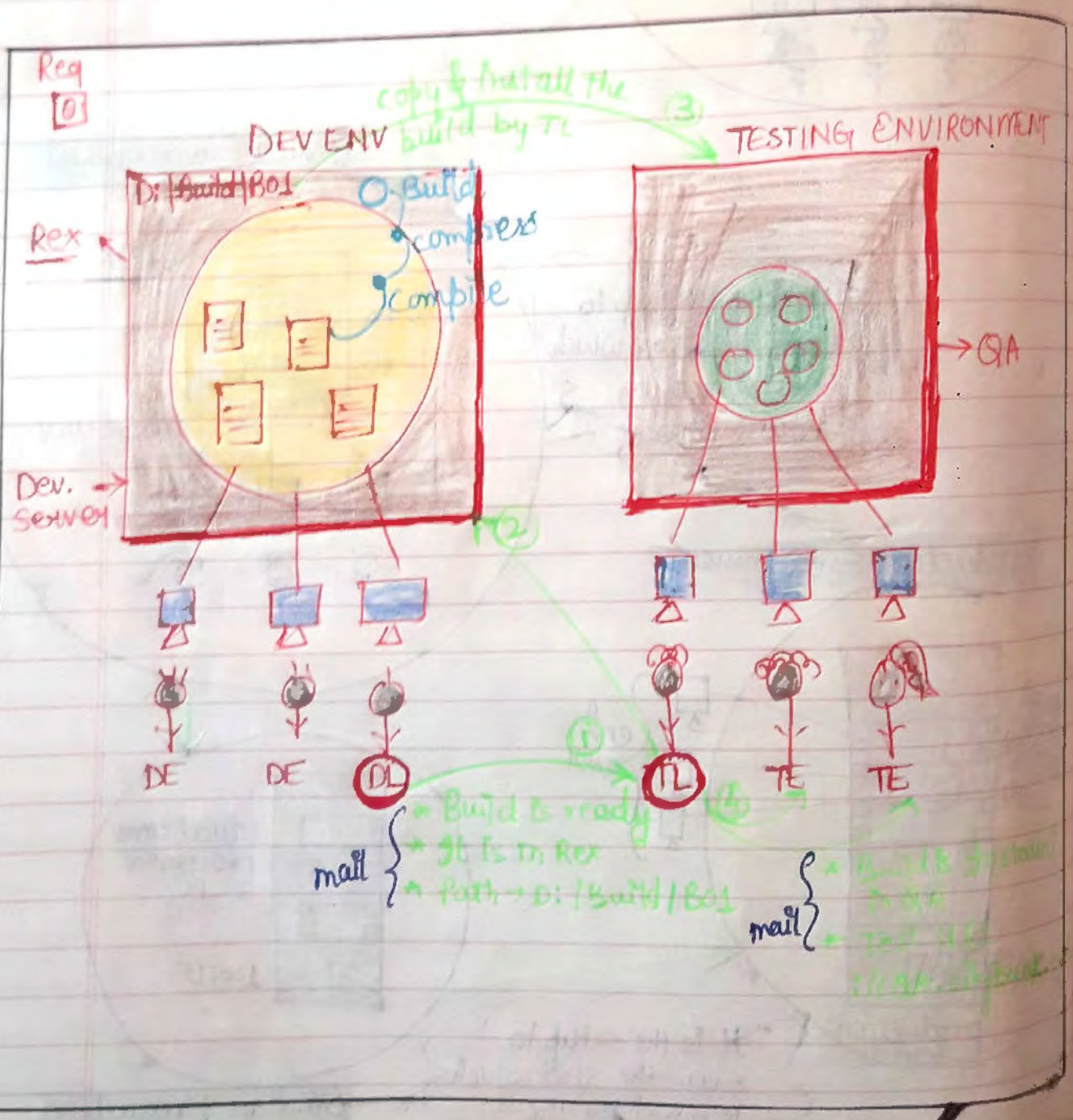
It is the setup to run the s/w which is used for the busstnels



Similar Production Environment

Why testing environment should be similar to production environment in system testing?

The software we tested in testing environment & installed in the production environment that software may not work properly because of different configuration because of this reason testing environment should be same as production environment.



* Developers will develop the program & will compile & compress it to create the build.

* Once the build is created it will be stored in some drive & Development lead will informe to the Test lead that the build is ready & stored in Rex(sewer name) in some drive & link for its path will be given in a mail.

* Now test lead will click on the path given & will login to Rex sewer & copy the build & install to testing sewer & will inform to Test engineers that build is ready & will provide the link ~~of~~ for path to them through mail.

* Now TE's will test the build according to the modules assigned to them.

* If test ex. finds the defect he will inform to Development lead & then he will inform to the related development engineer & the DE will fix it.

* As soon as you get the defect, you must immediately raise it.
(if not - u may forgot it, anybody else may take the credit)., developers our time will be wasted if report is given at EOD)

* As a test lead if you get the new build then, you must uninstall the previous one & install the new build in your test sewer for further testing the application.

* Suppose 100 defects has been raised by the testing

team & development team have fixed 60 defects out of it & have sent the new build, then we will have to retest only the fixed part & will never retest unfixed part.

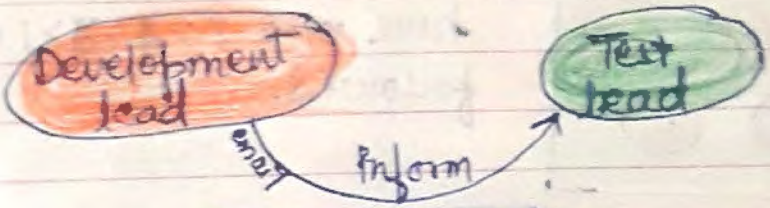
* When ever we find the defect we make a report through some tools & a no. is also given to that bug which give ease in identification & understanding.

* Defect can be raised by any one (any tester) but the one who raised the defect, will only be doing the retesting for it.

* Author of the feature should test the feature, author of the defect should retest the defect.

* There will be some features like login, sign in which will be used by all but author will be only one.

defect in login was raised by Dingi, because that was common. but the features those are not common will not be raised defect by others because they will be bussy in there own modules only & others can take revenge of it by doing the same. But anybody can raise the defect any time.



* Buil ts ready
* path → D:/Builds/B01
B02
* module A is developed
(OR) * module B is developed
60 defects of A is fixed.

D1 -- } Details about
D2 -- } fixed defects.
D3 -- }

* Build: B_01.

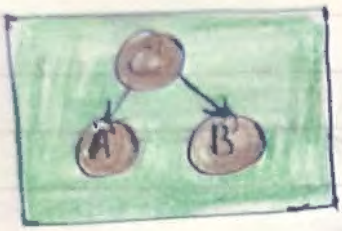
Build: B_02

Build02 will be ~~having~~ containing the new module as well as the previous one also.

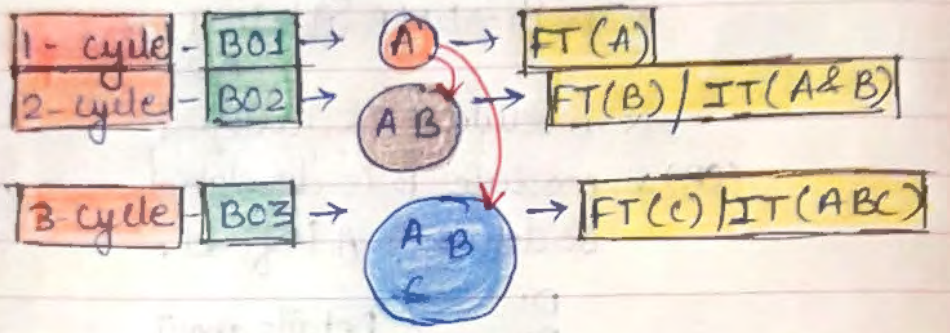
* ① Test old module A
② Test New module B
③ Defect fixed in A
④ Integration b/w A & B

2 > 4 > 3 > 1

If you follow this strategy, you can catch the defect soon, but its not the international standard to be followed.
coe will first test the new module, because manager will ask the status for the new modules first.

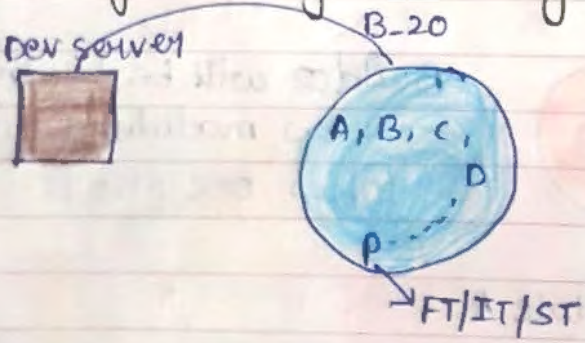


here also 2>4>3>1 will be followed.



Defects found in the first test cycle might be fixed in the second ~~of~~ test cycle or in third test cycle or in forth or at the end of the release depanding on the priority of the defect.

Q. where you do system testing?



① Minimum bunch of features should be ready.
② Testing environment should be similar to production environment, is ready.
③ ~~of~~ old / basic features should be stable

up till add to kart features its application is ready but its not sufficient to do online shopping because payment gateway is not added.

1 cycle
2 cycle
3 cycle
:
19 cycle
} FT/IT

20 cycle — FT/IT/ST

:

40 cycle — B-40 — A, B, C ... Z

all the modules requested by the customer are delivered.

41 cycle — A, B, C ... Z
42 cycle "
43 cycle "
44 cycle "
45 cycle "

last few cycles are for retesting, no new module is added in FT.

RETESTING
(testing weather the defects is fixed or not.)

46 cycle — No defect, all modules are beautifully working, all bussiness senerious are working fine, no block & critical defects, s/w is working fine in testing environment similar to production env

⇒ It's the time to release the s/w

S/w can have many release (eg. whatsapp ask for the update with its new release).

Q. When do we go for release?

(i) When all the modules requested by the customer are ready.
(ii) When there is no critical or blocker bugs.
(iii) When the s/w is tested in an environment which is similar to production env.
(iv) When all the end to end senerious are working fine.

Q. What is release?

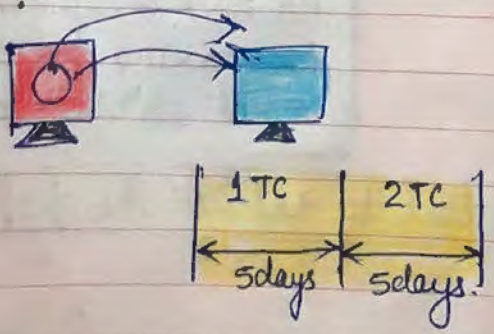
Starting from gathering requirement, develop the s/w & test it for so many cycles ~~w~~ & until you launch the s/w to production is called as one release.

Q. What is test cycle?

It is an effort/duration spend to start & complete the testing is called test cycle.

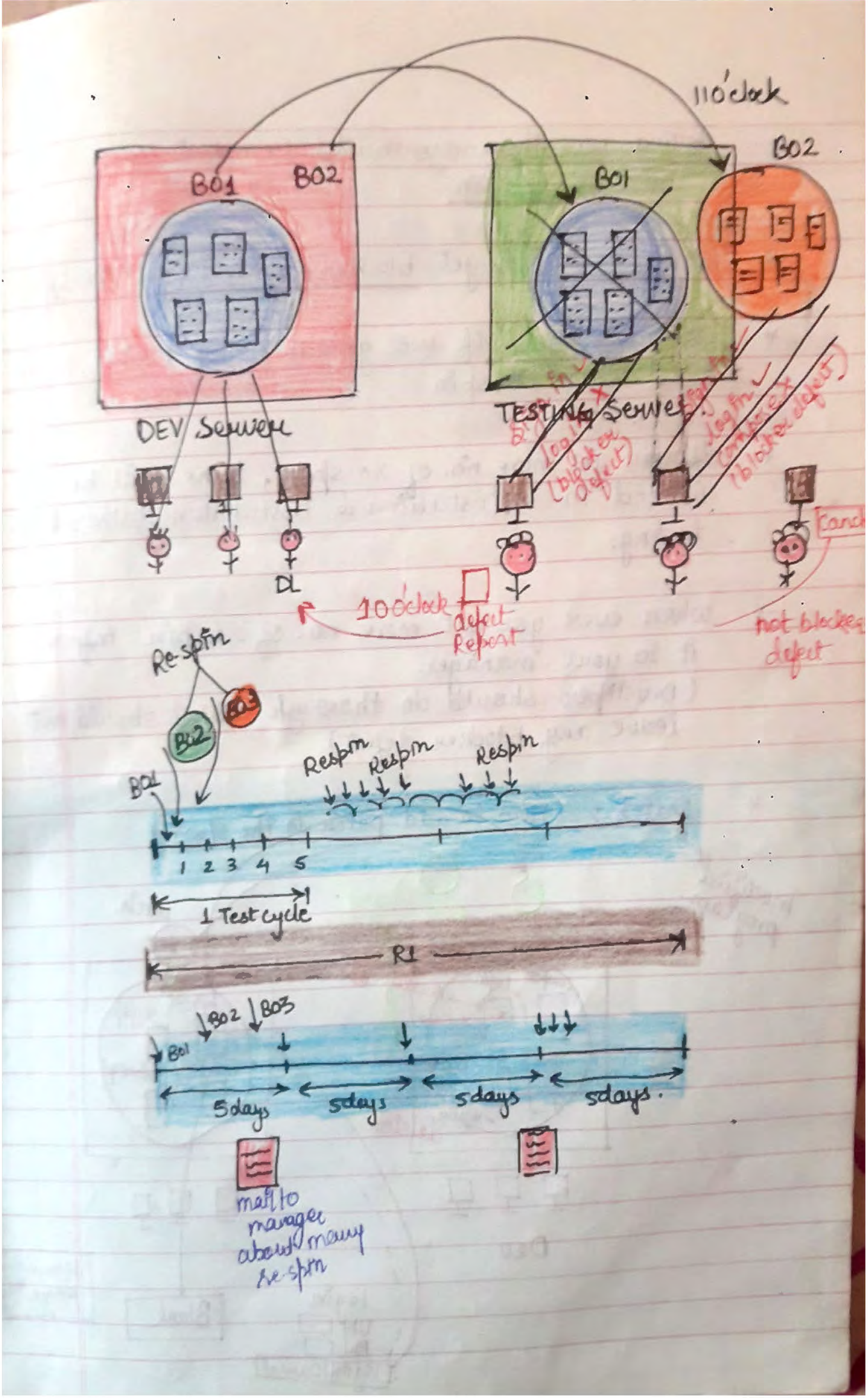
1 test cycle might take 3 days, 4days, 5days, 6days It depends on size of the application, complexity of the application & no. of test engineer.

Q. In the current project how many cycles you have tested?



1 year Experience

1 month = 4 cycles
1 year = 4×12
= 48 Test cycles.
(48-50 TC say)
• Show like you are thinking & answering.



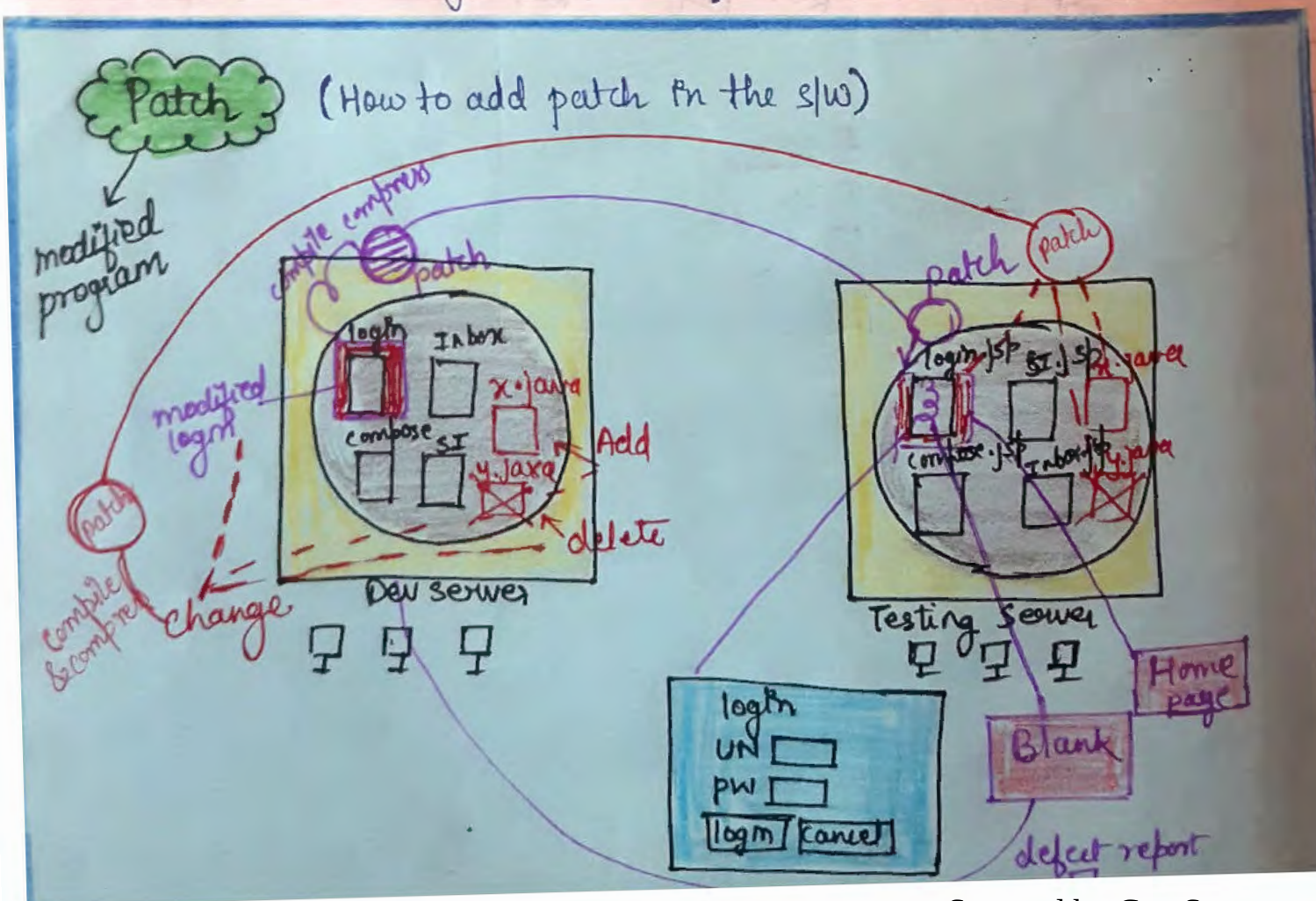
* Getting more than one build in a test cycle is called as Re-spin

* When ever we get blocker defect we get Re-spin

* In a test cycle if we get 'n' no. of builds also, it is called re-spin.

* If you get more no. of re-spin, time will be wasted in uninstallion & Installation rather then testing.

* when ever you get more no. of respin inform it to your manager.
(Developer should do thorough UBT & should not leave any blocker defect).

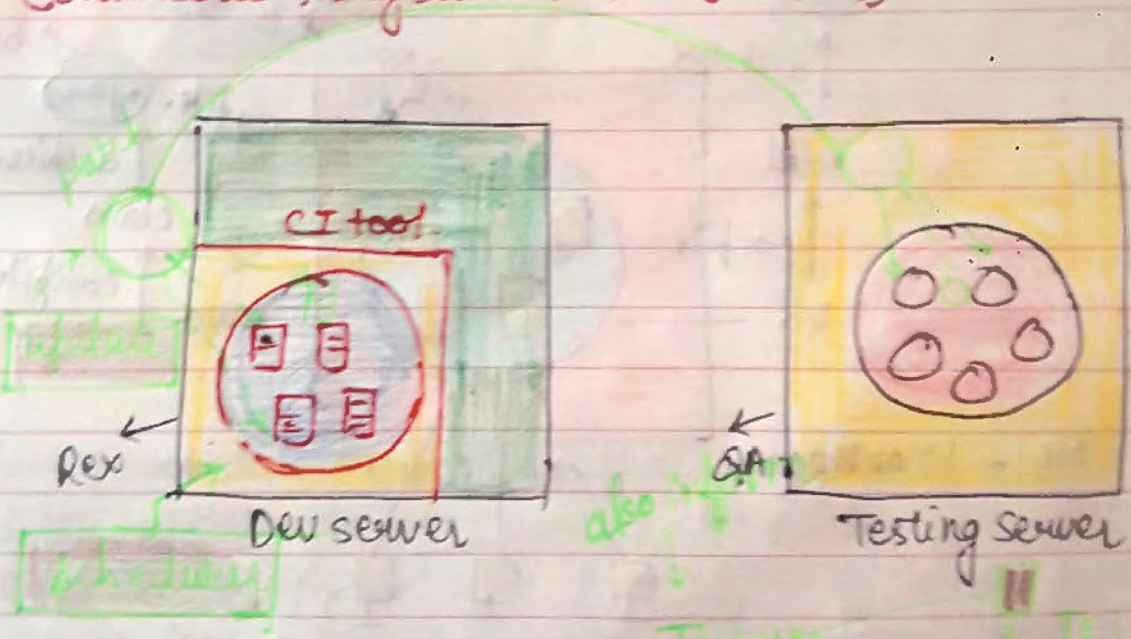


* Patch is a modified program, when ever you get patch you don't need to uninstall you just have to update. you will get patch when ever there is a small defect or blocker defect.

* Patch will be made only when there are some small changes else build is created, if you get new build you need to uninstall & reinstall it again.

* changes in program can be like -
  adding a new program
  modify the existing one
  or deleting the existing one.

* Continuous Integration tool: (CI tool)



CI tool is installed in the development environment

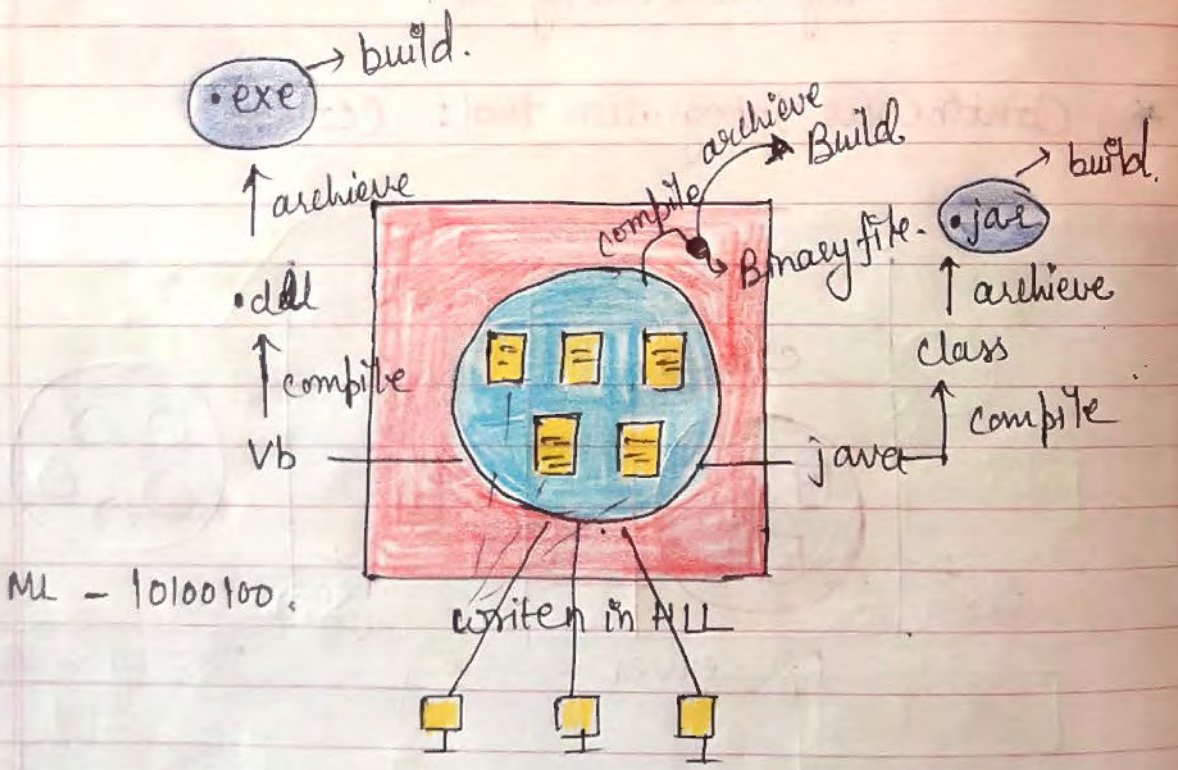
Update button is present in the CI tool. while developer click on update it automatically picks all the

updates & make a patch of it & sent to the test sever & install it there & also update (inform) the test engineers that the patch is ready along with test path.

* continuous integration is doen by CI tool between dev & test ~~sever~~ team, so it is called continuous integration tool.

* Scheduler feature is present in CI tool through which it automatically take update in a fixed interval of time even without clicking on update.

*



Compiler → High L'L → Machine LL

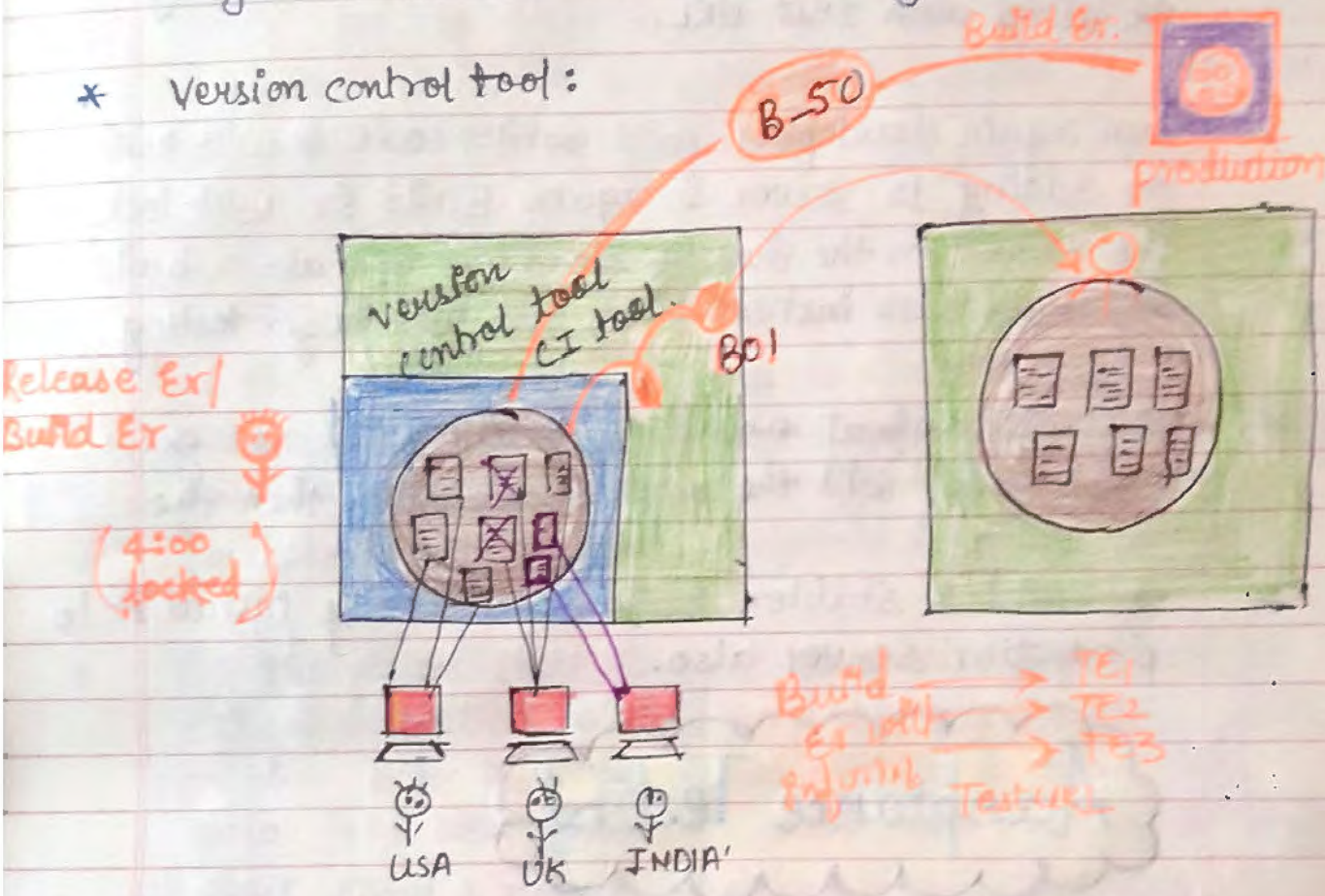
error → compile time error → syntax mistake

Execution → Runtime error → logical mistake.

Q. Who is involved in the installation of the s/w?

Anybody from testing team or anybody from development team or release engineer or a field engineer, might be involved in installating the s/w.

* Version control tool:



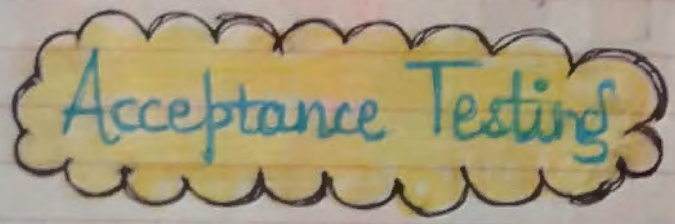
* ~~In CI tool~~ It may happen that some body might delete/update the programs of somebody else, without even taking permission from him or some body can do less amount of work & go, mean there is no record of work doen by everyone & no. information about the updates. so we use version control tool - which will keep track of who have performed what & all the changes made.

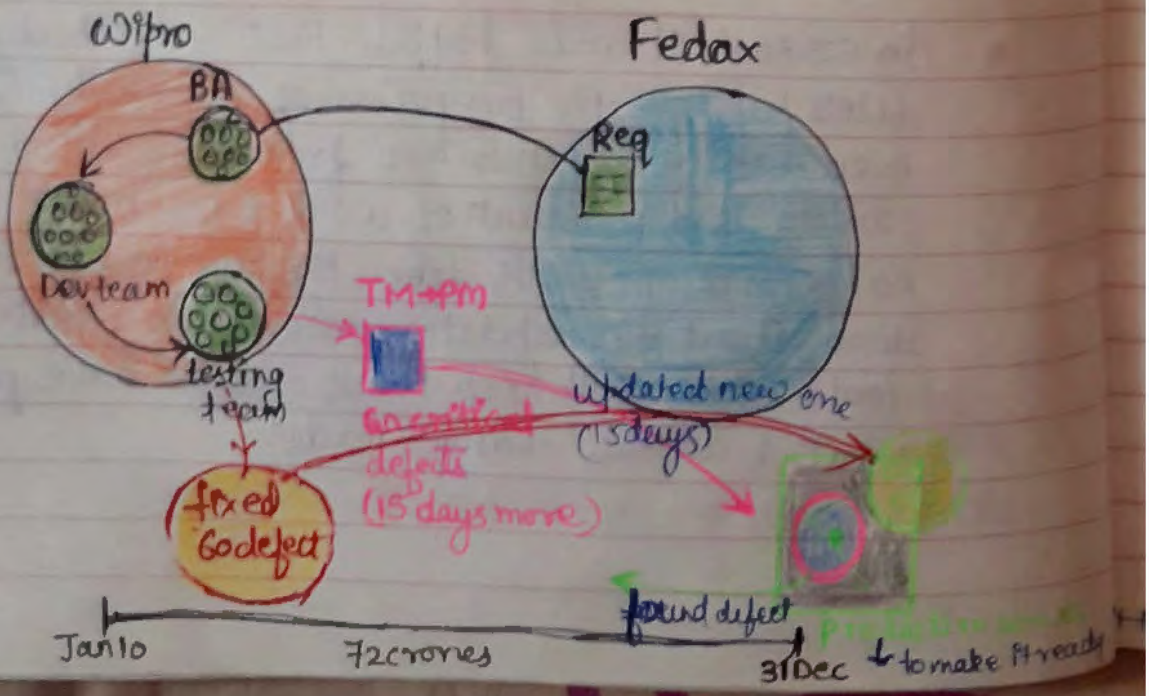
* We have build Er. or we call release Er. who informs to the development that at some hr. of the day he will lock the development server & then he will create the build & install it in testing server & will also inform all the testing Er. along with test URL.

* Then again developers will write code & will keep on adding to server & again Build Er. will lock the server on the day of dead line & make a build & install also install it on test server for testing.

* This will repeat under the build until we get a build which will the most stable one, then the

* If build is stable, Build Er. will only install it to production server also.





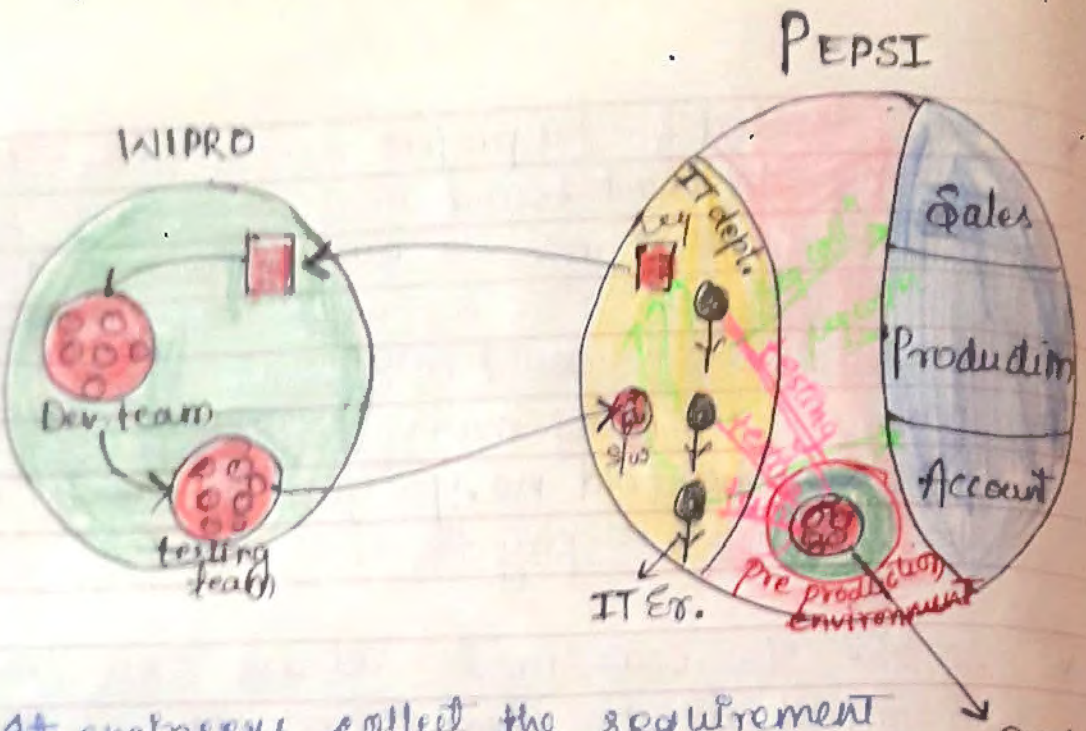
* time decided for the project was 31st dec, but after testing Test lead found at the end that s/w is running well if we see externally but internally it is left up with some 60 critical defect & to fix them it would take some 15 days. He informed to the Proj. manager & tak tried to convence him but he told no, you just launch it otherwise we will have to pay them.

* so the s/w was made release even with those 60 critical defects. Now at customer's place ~~it~~ they got the s/w & where trying to install production server which took them nearly 15 days. Now they have installed & where running the s/w & they found a defect & then informed to the wipro.

* By that time these bugger have also fixed all 60 critical defect & so they told ok we reactify your problem & its just the small issue & then they install the new build in which all the 60 se defects where fixed & now everything was running beautiful & even the ~~fe~~ fedex was ~~also~~ & also not made aware about 60 defects this ~~way~~ was the trick played by project manager & they escaped.

* But there are the companies ~~which~~ which might be older than wipro & much aware about s/w & even have there own IT depts. & there IT ~~engg~~ Er. come in role, in those sort of senerious these wipro people can't cheat them.

took 15 days.



* It engineers collect the requirement from different depts. & pass it to many s/w companies (diff. modules to diff companies) & ones the agreement is signed & s/w is developed & delevered to there company they will first test it in pre-production environment & if working fine then only the, will install it to production environment.

Why is IT branch?
All the s/w companies have sent the request to technical board that we want professionals who have technical knowledge of collecting the bussiness requirements & conveying it to s/w companies & get there s/w prepared & can also understand & test the s/w & know all about it. And then after the IT branch was introduced.

CS people are technically good they can develop what ever the req is given to them but they don't know about the actual senarious of how the particular business run's.

* IT engineer do acceptance testing, they can do all: FT, Integration testing, sys testing but generally they do end-to-end testing.

* Is there any difference b/w end-to-end testing ~~done~~ doen by test engineer & end-to-end testing doen by IT engine
→ yes, TE will do according to CRS, IT people will do it according to ~~practicle~~ practical need of the bussiness & customers.

* In 1st phase end-to-end testing doen by TE & IT er. will be 80% same & 20% (diff. doen by IT people) this gap ~~be~~ es as the no. of years with that s/w ~~fed~~.

* But TE's may leave the company if they get good experience - so IT companies are paying high for documen~~ta~~ting things, so that it can be ~~refe~~ refered later also in case of long term projects.

Q. What is Acceptance Testing ?

It is an end-to-end testing doen by IT Engg. sitting in the customers place where in they take real time bussiness senerious & check whether the s/w is capable of handling it or not.

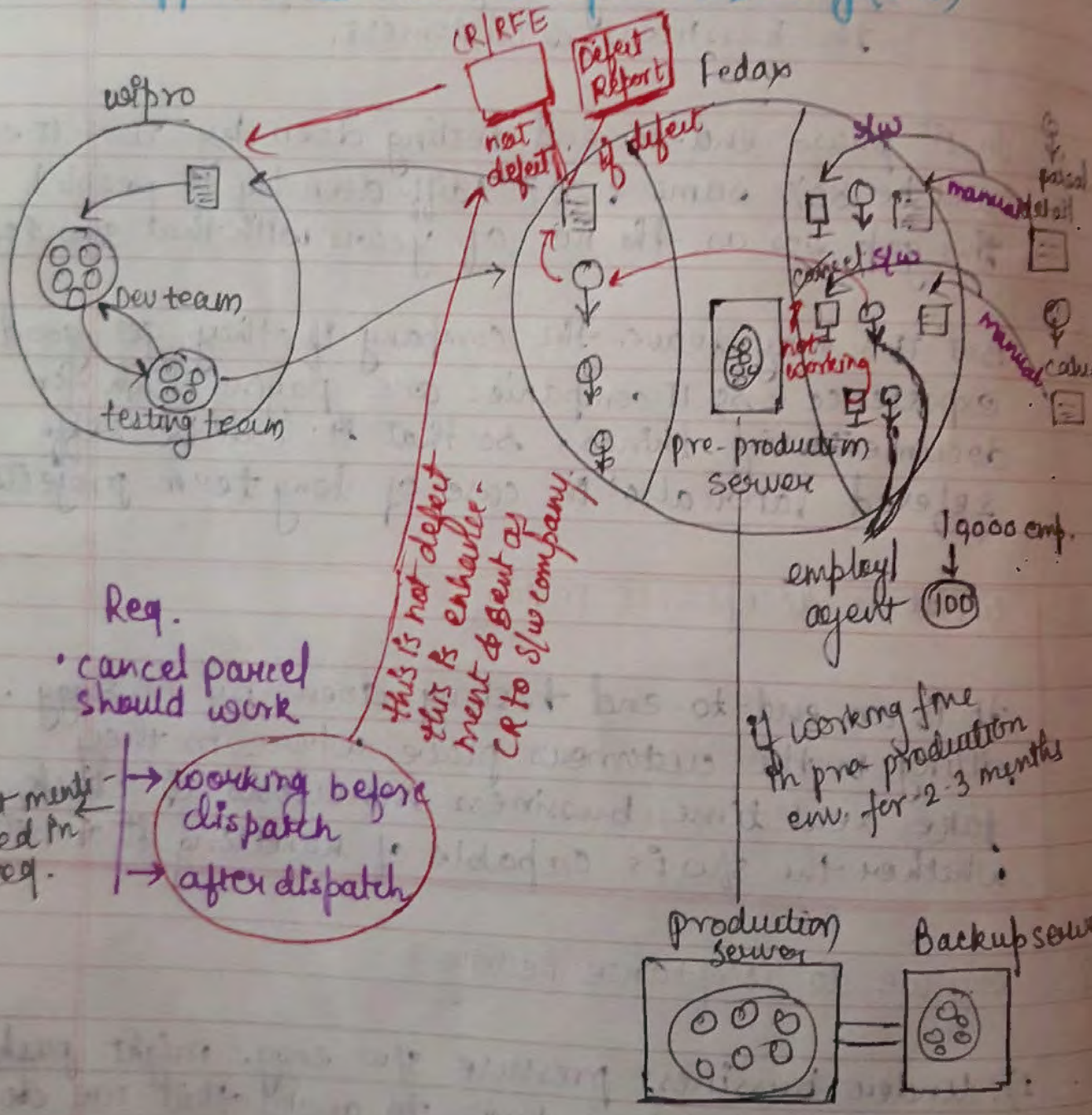
Q. Why we do acceptance testing ?

1). Under bussiness pressure s/w engg. might push the s/w with lots of bug, to avoid that we do acceptance testing.

2). With critical bugs if customer move the s/w to the production, he will be under severe loss, to avoid identify that we do AT.

3). Developers might misunderstand the requirements & developed wrong features, to identify that they do acceptance testing.

Approach - II User acceptance testing (UAT)



* CR (change request)
RFE (Req. for enhancement)

Q. If IT Er. approve the app then which report they sent back to s/w company?

As IT team of Fedax is small, so they will take some 100 employes/agents of fedax & involve them for testing, where these 100 people will test the s/w like they will manually keep the record of data (as they use to do) as well as use the s/w simultaneously to check whether its working fine or not, if they find any defect they will inform to IT Er. & IT people will cross verify it, if its a defect they will raise it to the s/w company & get it fixed & if its not the defect according to CRS then they have given to s/w company then IT Er. will document it as CR/RFE & sent it to s/w company to enhance the s/w. once the s/w works fine in pre production server for some duration (eg. 2 months) then it will be launched to actual production & now no manual work will be doen.

Defination: It is an end-to-end testing doen by end users when they use the s/w for the bussiness for a particular period of time & check whether its capable of handling real time bussiness senerious.

Q. Why the no. of acceptance test cycle ies?

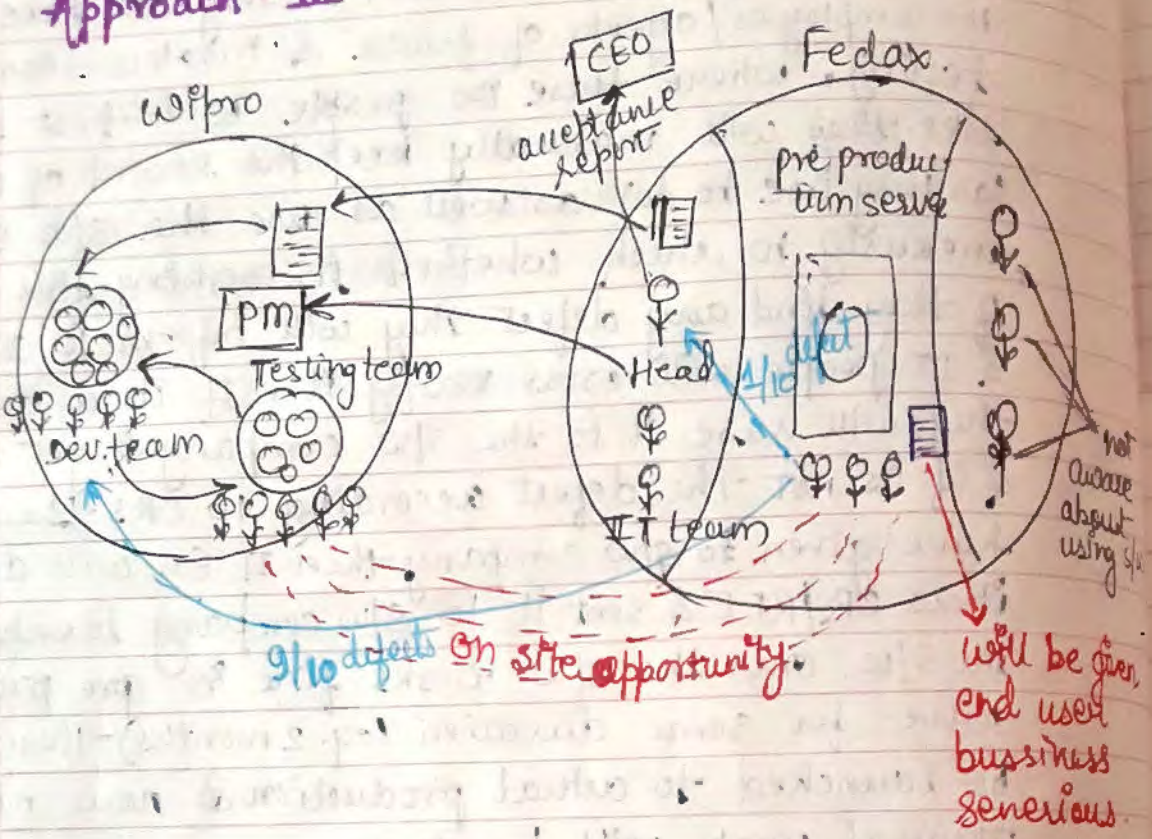
① After delievering the product, customer is getting lot of new ideas so he is asking for lot of changes.

② The requirement which was given in the beginng is not clear so it is leading lot of rework.

③ Testing team has not doen proper testing & they have missed lot of bugs.

Q. If change in req. happens in approach-III & IV then this report is called what?

Approach - III



* IT team is small & end users/employes are not much aware about the s/w usage, so IT head will ask the project manager to send some of there tester in there place to perform acceptance testing. Now this is called onsite opportunity for TE's and IT team will collect all the requirement & generate end user bussiness generious & ask there TE's to do acceptance testing refering to it

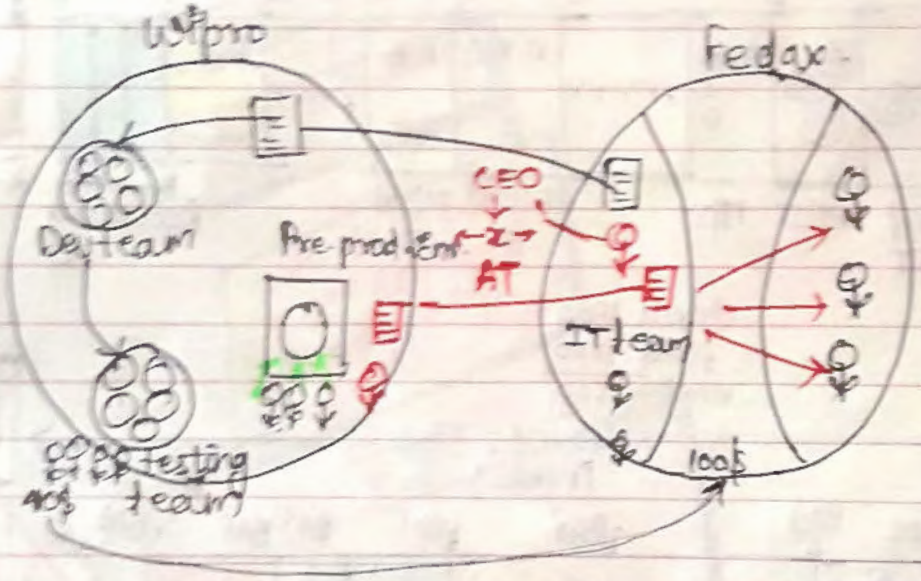
* Now TE's at customers place, if finds any defect will inform to IT team, but internally a game is happening that out of 10 defects 1 will be told to IT head & rest 9 will be directly told to development team & will be fixed in the name of that 1 defect.

* This story is known to IT head also, but he is

more focused on quality of s/w other than making money from bugs.

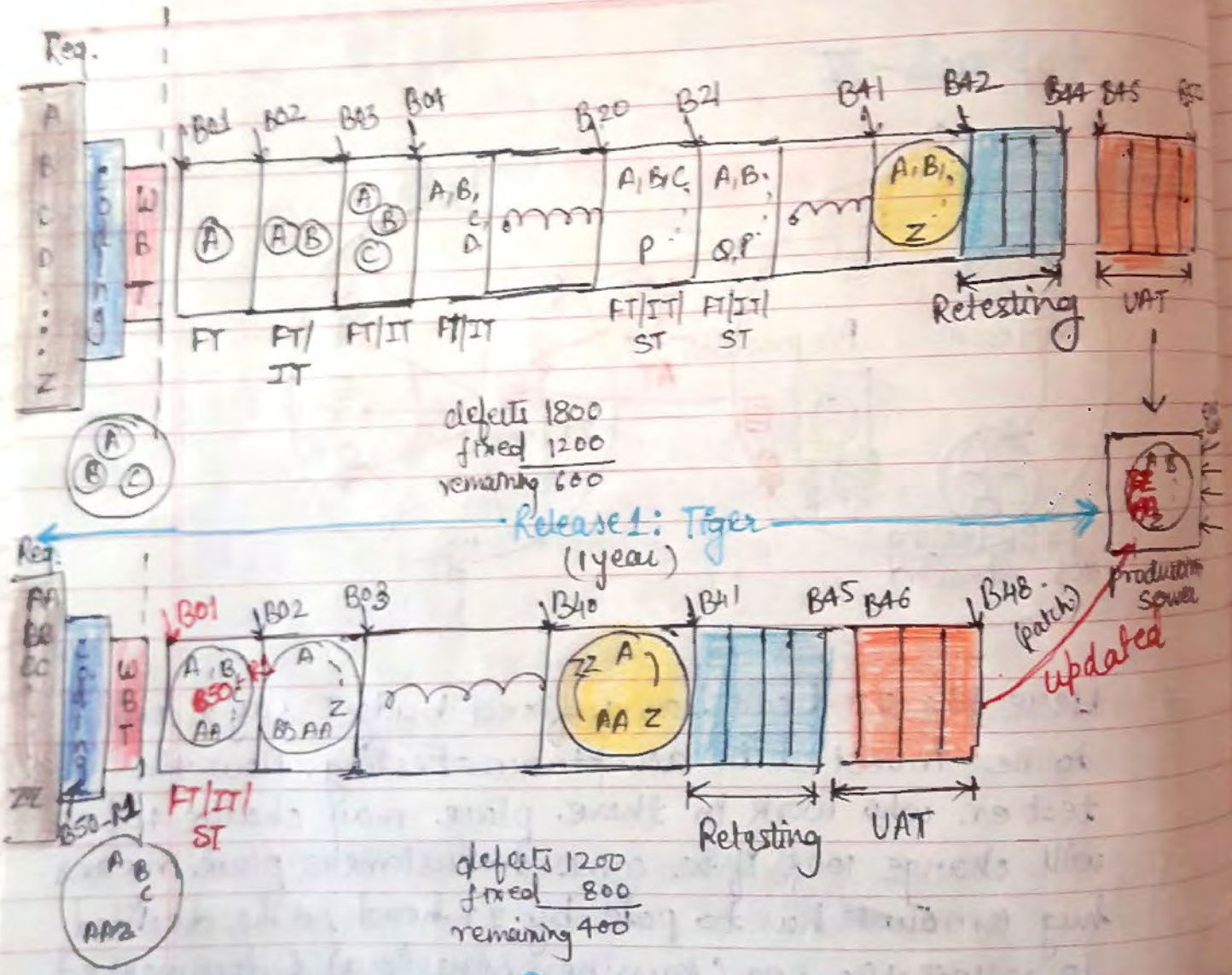
Definition:
It is an end-to-end testing which is doen by our own test engineers sitting at the customers place, where ~~it~~ in they refer end user bussiness senerious & they check whether s/w is capable to handle it.

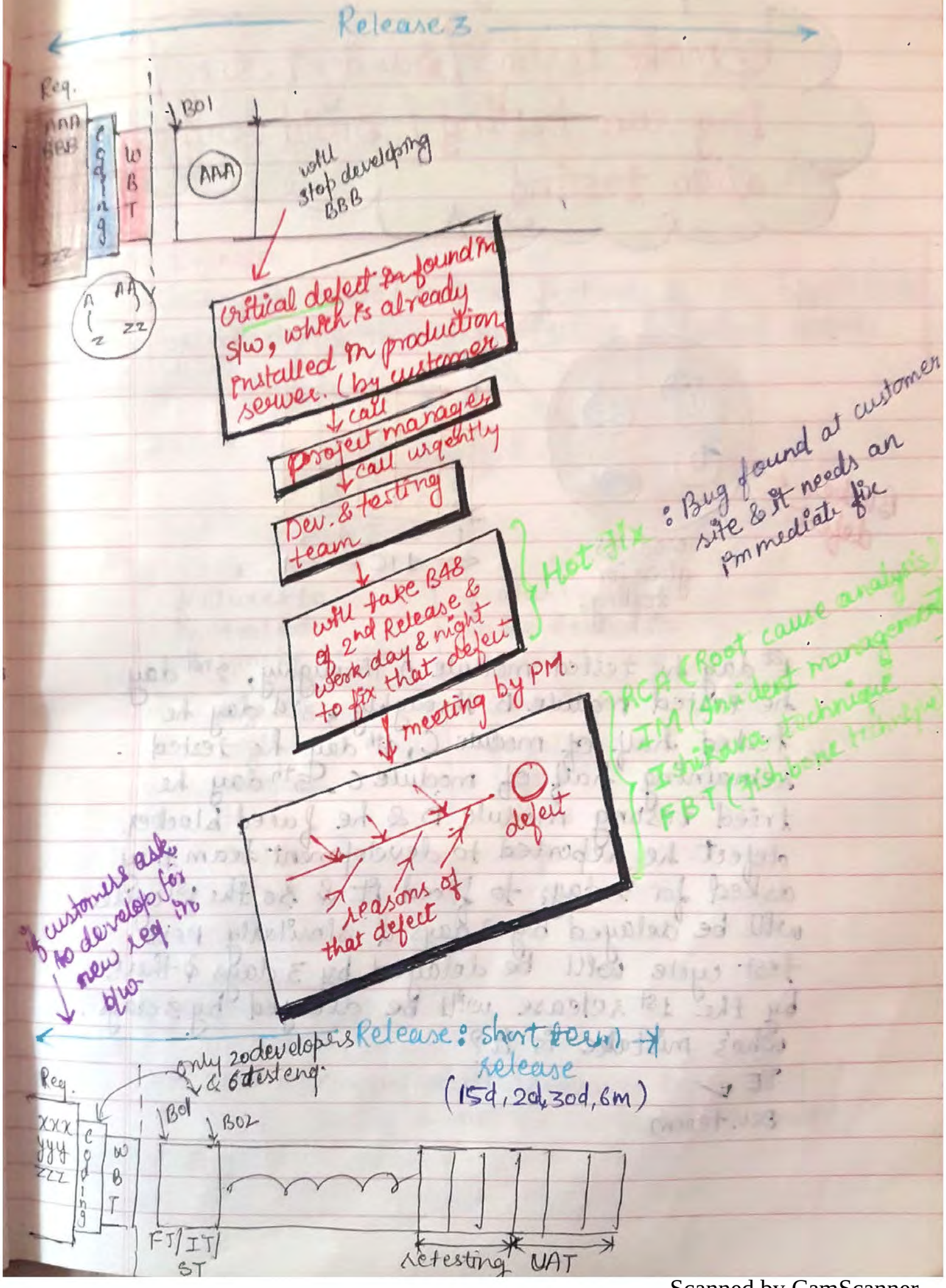
Approach-IV



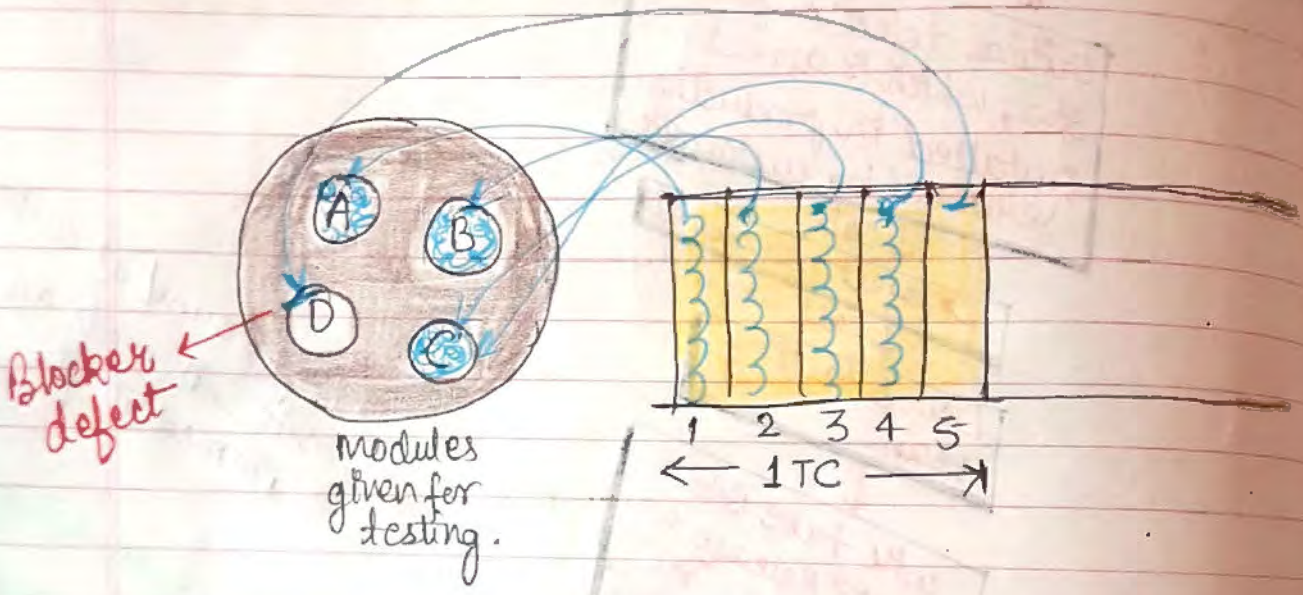
* Here the IT head has a fixed budget say x amt to be invested in acceptance testing. Now the tester who work in these place may charge 40$ will charge 100$ if he comes to customers place means huge amount has to paid by IT head so he decided to collect the req (bussiness senerious) & documented them & went to the customers place. Now here he takes some TE's & ~~pay~~ ask them to perform AT for these s/w. by reviewing ~~that~~ all those bussiness senerious.

Definition:
It is an end-to-end testing doen by our own TEs, sitting in their own place. they are given the real bussiness scenarios & they test whether the s/w is capable ~~to~~ ~~hand~~ of handling it or not





Smoke testing / Sanity testing / Dry-run testing / Build verification testing.

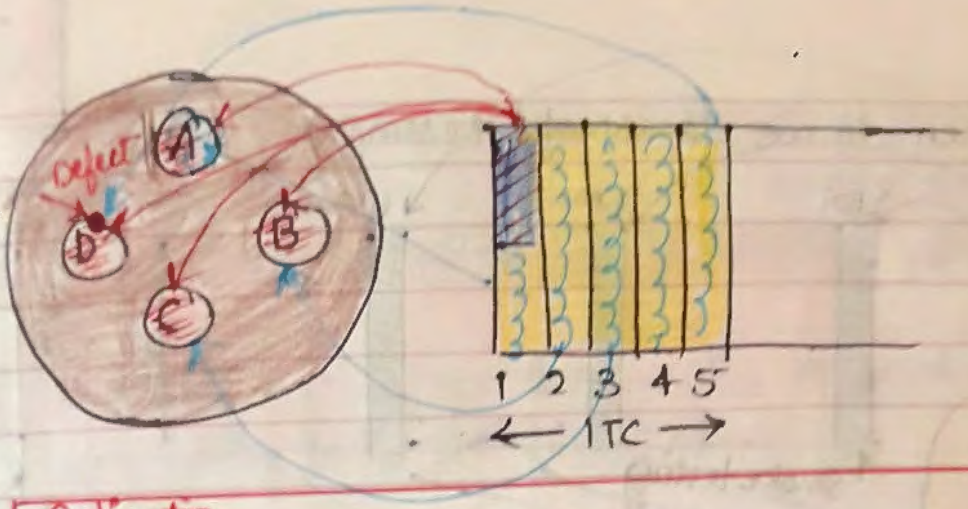


1st day he tested module A throughly, 2nd day he tested module B throughly, 3rd day he tested half of module C, 4th day he tested remaining half of module C. 5th day he tried testing module D & he faced blocker defect. he reported to development team they asked for 3 days to fixed it & so the 1st cycle will be delayed by 3 days & similarly next test cycle will be delayed by 3 days & there by the 1st release will be delayed by 3 days.

Who's mistake is it?

TE ✓

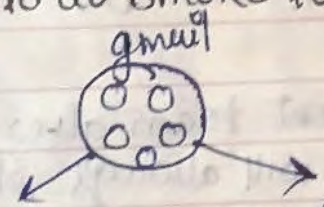
Dev. team



Defination.

Testing the basic / critical features of an application before doing throughty testing is called smoke testing.

How to do smoke testing?

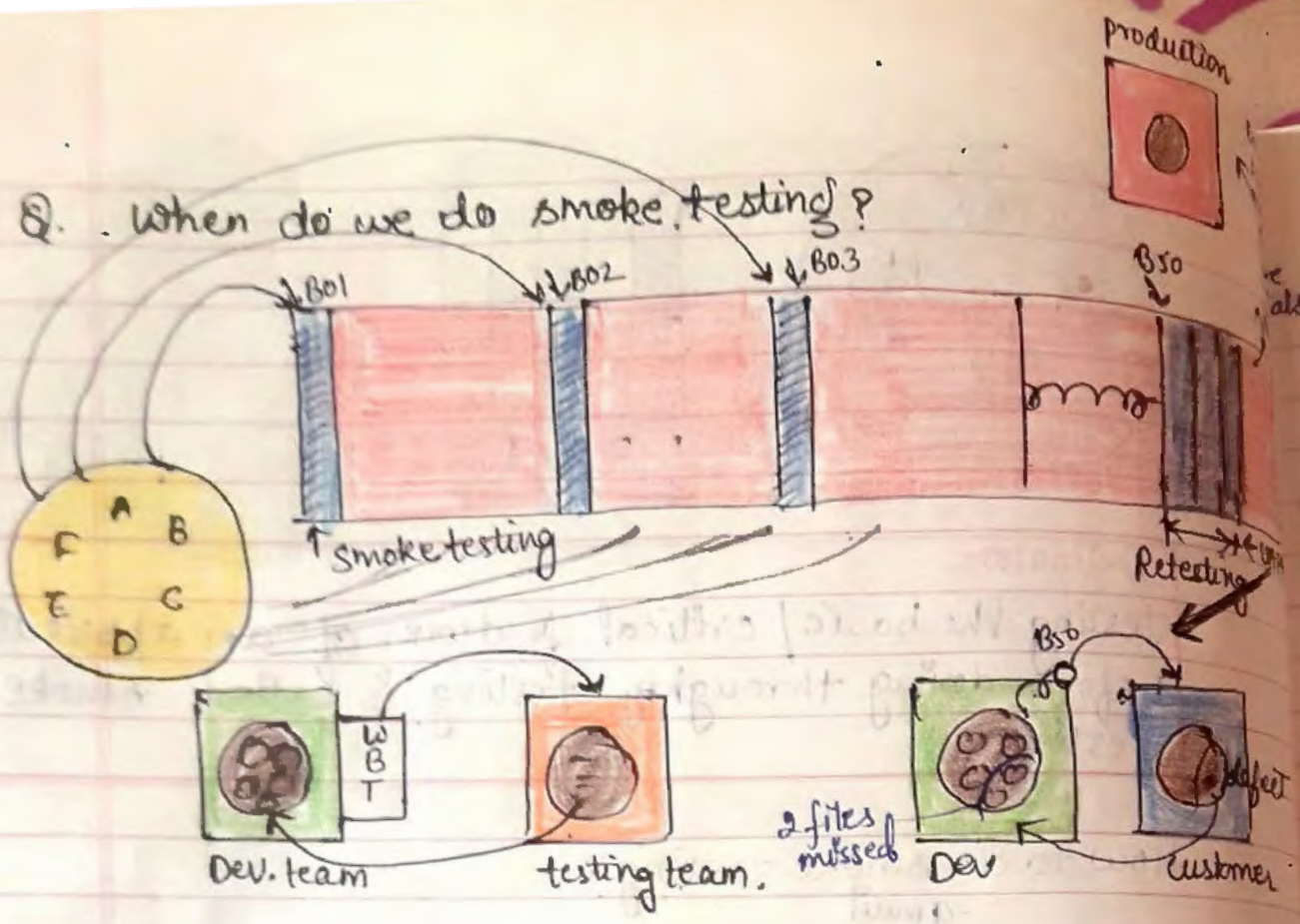


features to be tested as smoke testing	features not to be tested in smoke testing.
Sign up	Sent Item
login	forgot plw
compose	logout
Inbox	labels
~~Sent Item~~	help.
Draft	
All mails	

Assignment 7

For FB & whatsapp find the features to be tested in smoke testing & not to be tested in smoke testing.

Q. When do we do smoke testing?



When ever the development team gives build to the testing team, we will always start with smoke testing.

When ever the development team gives the build definetly they would have doen some changes & those changes may affect basic or critical features so we start with smoke testing.

While doing UAT at customer side also, 1st smoke testing should be doen because if they get defect later & get new build (after fixing the defect) they need to test it again from the begining & that will be the waste of time so smoke testing should be doen at the begining itself to know about blocker defects.

same

When ever the build goes for acceptance testing, customer should always starts with smoke testing.

In order to ensure that installation is doen properly we do a round of smoke testing after s/w is installed on production server.

Developer soon after doing WBT, do smoke testing before giving build to testing team.

Q. Why we do smoke testing?

① To check whether the build is testable or not.

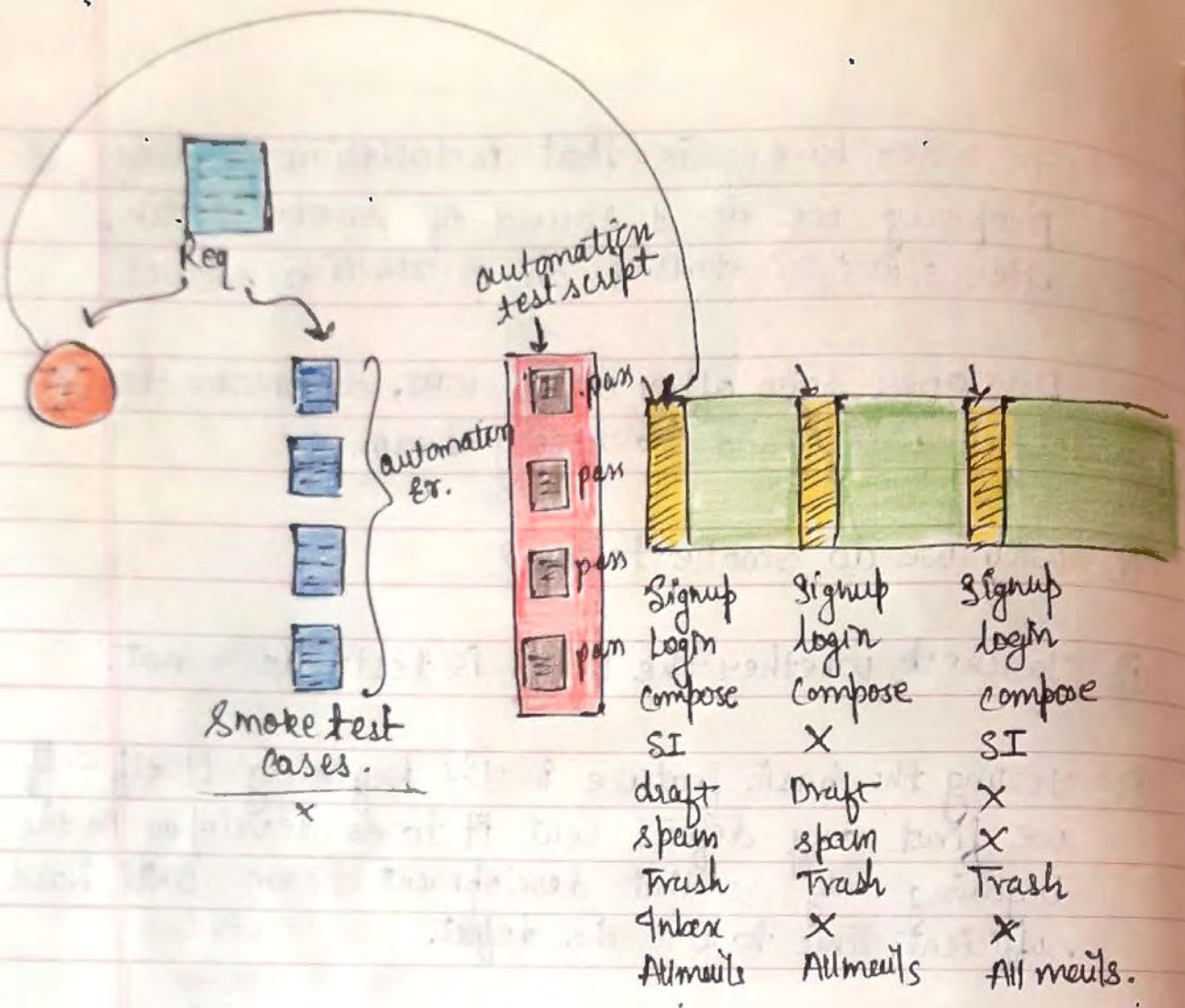
② Testing the basic feature in the beginning itself & if we find any defects sent it to ~~do~~ developer in the beginning only so that development team will have sufficient time to fix the defect.

③ Doing smoke testing indirectly conforms that the product is installed properly.

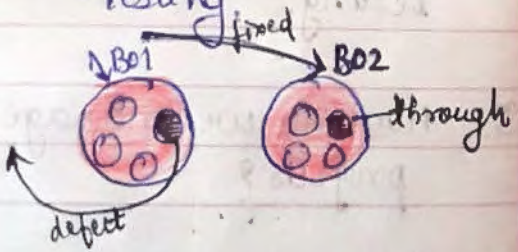
④ It is a kind of health check of a s/w where in we try to check, whether we received broken build from development team. It is also called as Build verification testing.

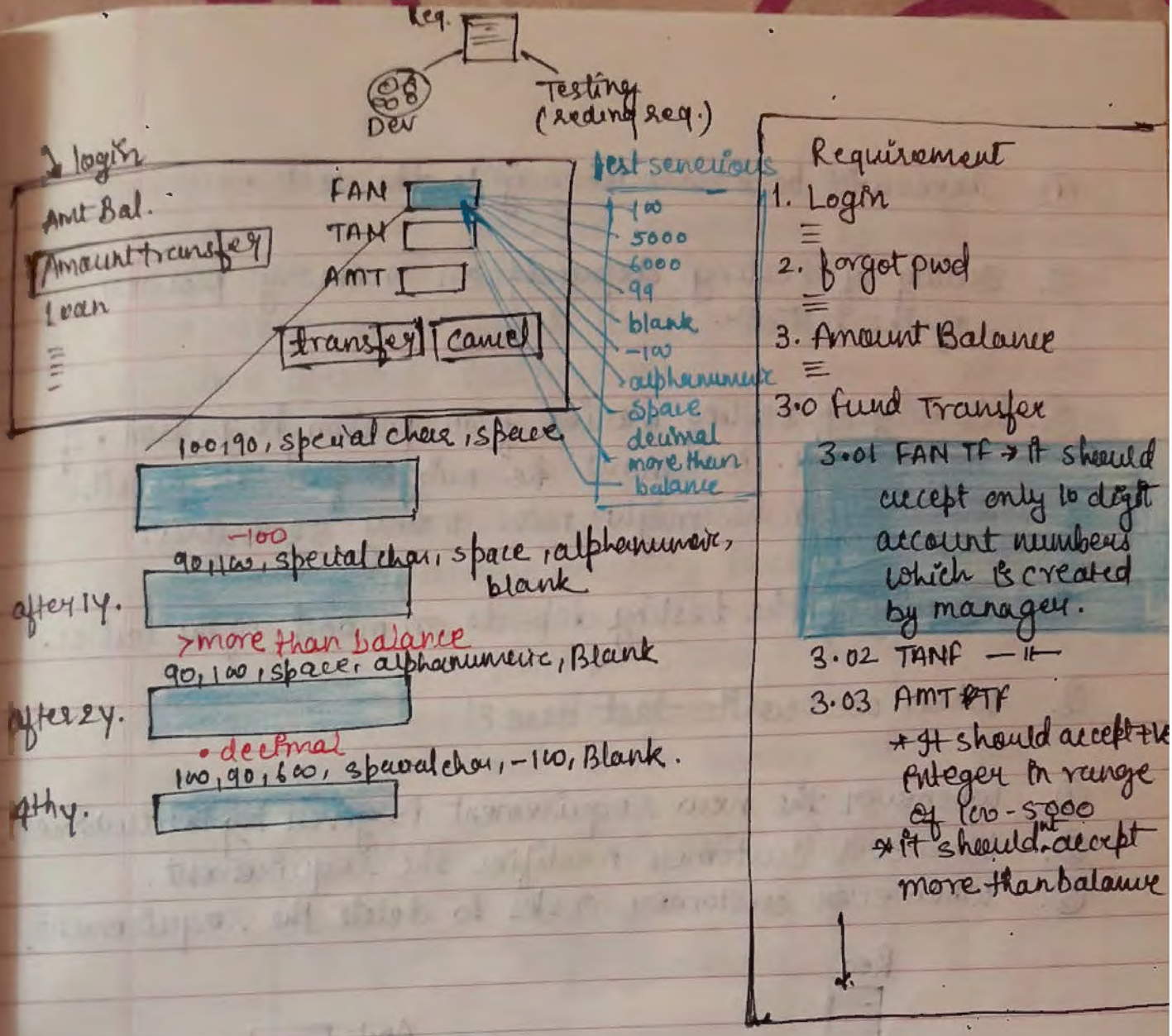
Q. How do we manage smoke testing in real time projects?

- By executing smoke test cases
(or)
By executing automation test suipt.



Difference between smoke testing & sanity testing.

Smoke Testing	Sanity testing.
→ wide & shallow testing. 	→ narrow & deep testing 
→ It is scripted → doen by dev. & test er.	→ It is unscripted → doen by tester.



Q. What is test case?

It is a document which consists of all possible senerios for a specific requirement.

It consist of following sections - step no., action or discription, Inputs, expected result, actual result, status & comments.

Q. what are the drawback if you test the application by directly referring the requirement.

(1) There will be no consistency in the test execution.

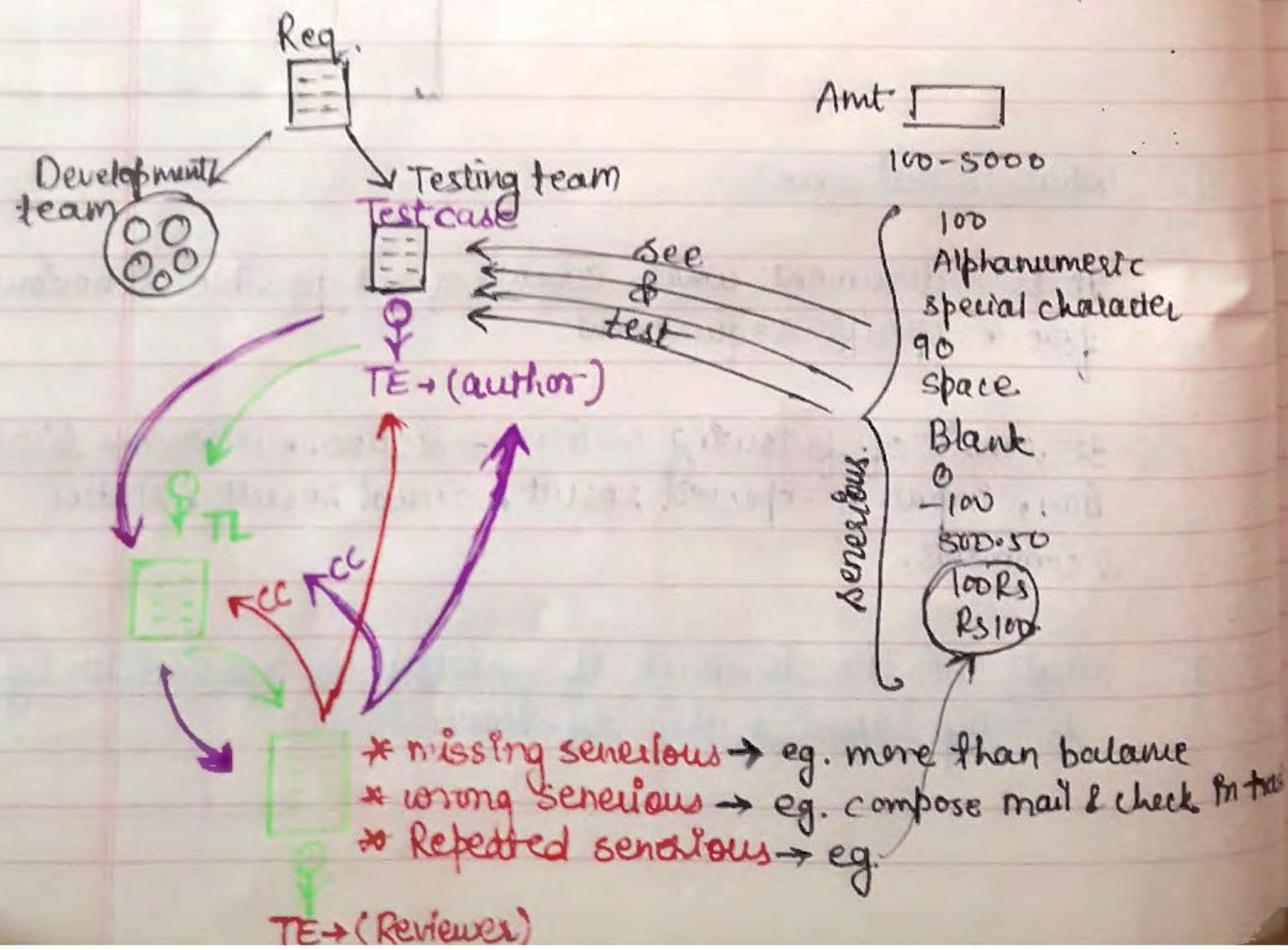
(2). Quality of testing depands on memory power of the test er.

(3). Quality of testing varies from person to person, if the test er. is smart he might find all possible senerious or he might miss many senerious.

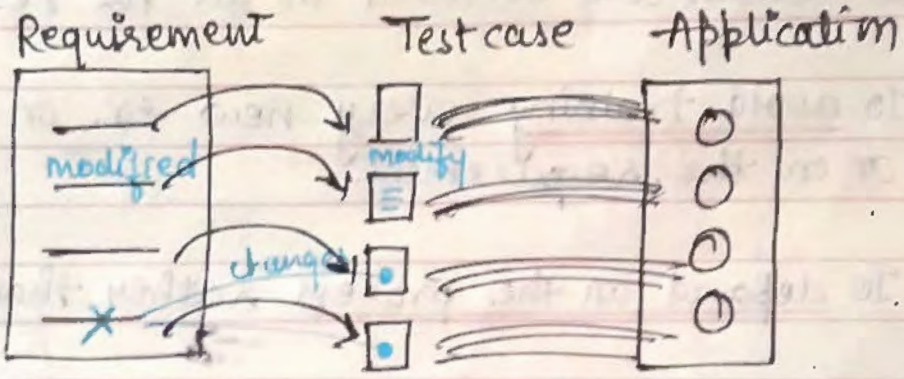
(4). Quality of the testing depends on mood of the test er.

Q. when we write test case?

(1). whenever the new requirement is given by the customer.
(2). whenever customer modifies the requirement.
(3). when ever customer asks to delete the requirement.



Tester who writes the scenarios is called author, he writes all possible scenarios by understanding the SRS & will document it which is then called as test case & sent it to test lead, he will sent it to some another tester called reviewer who will review the test cases, while reviewing he may find missing scenarios, wrong scenarios or repeated scenarios. Reviewer will sent these mistakes to author keeping team lead in cc. Now author will correct the document & sent it back to team lead as he don't know who is going to review it & then team lead will again sent to reviewinger. If no. of cycles are more mean TE is poor.



Testing the application by refering the test cases is called s/w testing. If there will be modification in requirement, test case will also be modified & application will be tested according to modified testcase. If some req. has been deleted by the customer, we will check if that deleted req. have affected to some other test cases, if so we will modify it & will re-check the application according to modified ~~Te~~. test case.

Q. Why we write Test case ?

① To have a better coverage -
when developers will be developing the product, the testing team will have a sufficient time, test er. should identify all possible scenerios & document it so that when ever the development team gives a built for testing, testing team will not miss any scenerios while test execution.

② To have a consistency in test execution -
means if you would have documented all possible scenerios you can ensure that all the scenerios are executed in all the test cycle.

③ To avoid training every new er. on the product or on the requirement.

④ To depend on the process rather than a person.

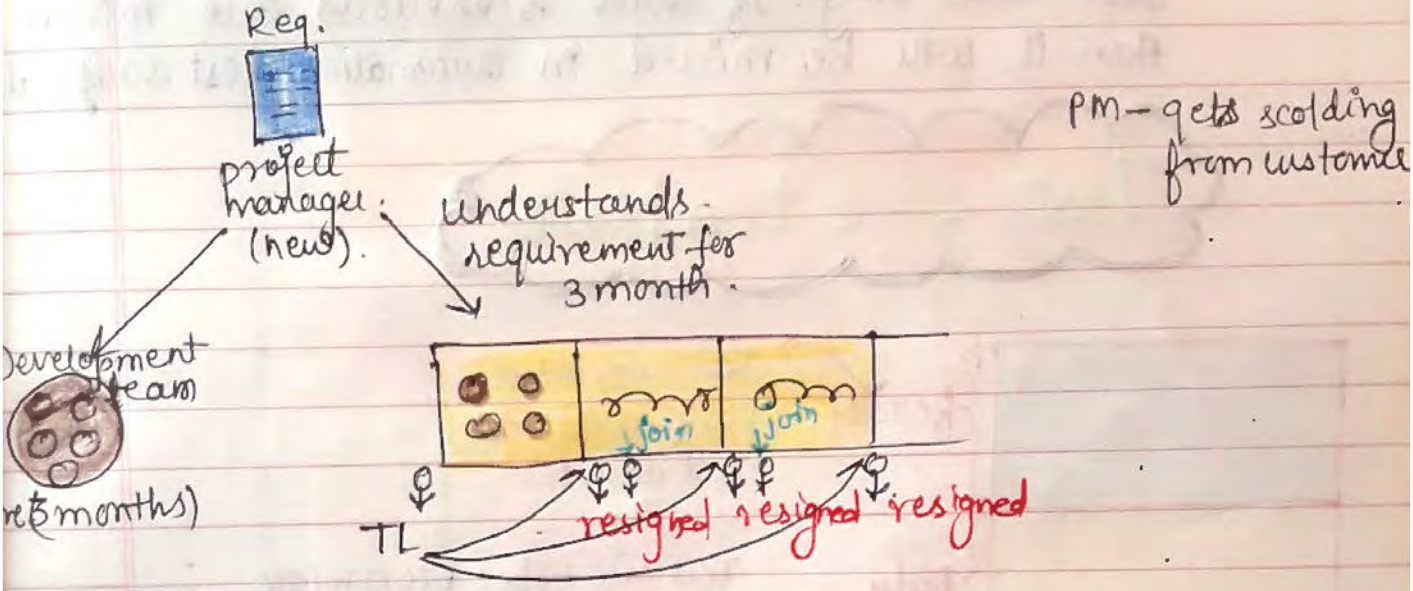
⑤ Test case is the only document which acts like a proof to developer, customer, project manager & management that they have covered all possible scenerios.

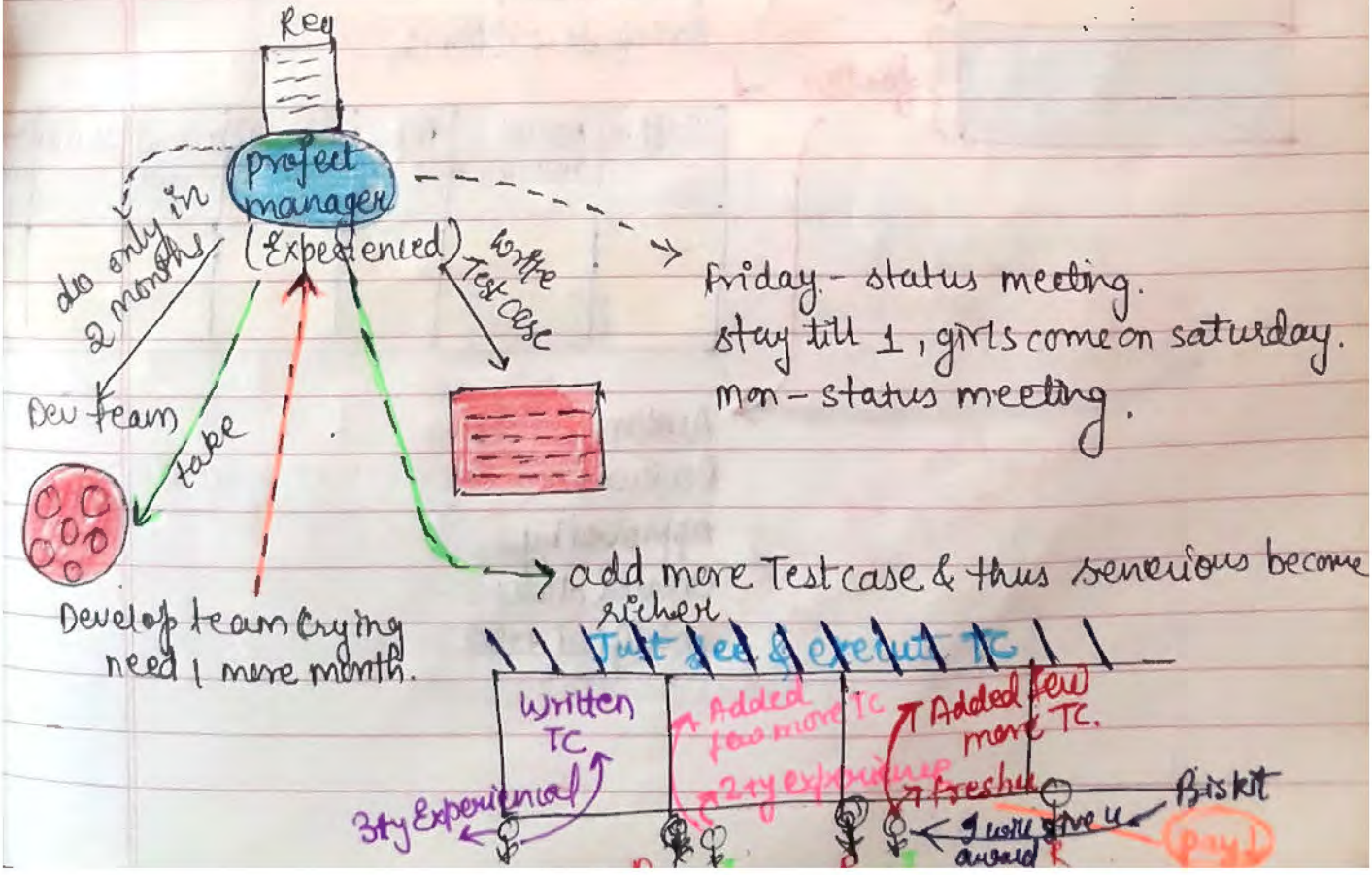
⑥ Test case, it acts like a base document to write automation scripts - if you write automation script by refering Test cases you can ensure that same coverage is maintained in automation script as well.

⑦. If all the scenerios are documented in very organised way it takes very less time for test execution.

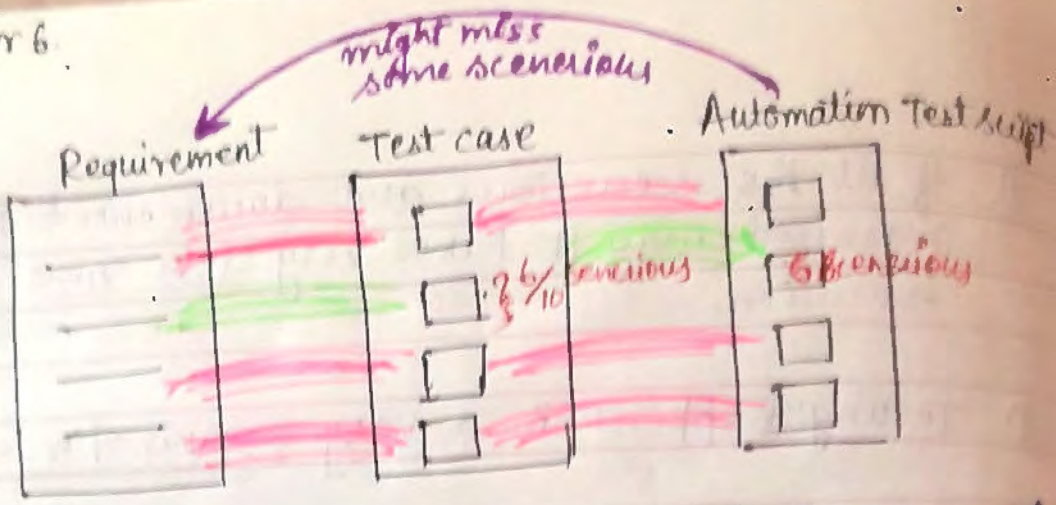
⑧. Testing happens in a very organised manner.

Explaination for 3 & 4.





Explanation for 6



test script will be written by refering manual test cases only. If some scenerious are missed then it will be missed in automation test script also

Test case template

Test case Name:
Requirement no:
Test data:
severity:
Test case type: FTC/ITC/SIC
Precondition:
Brief description:

Step No.	Action/ description	Input	expected result	actual result	Status	com...

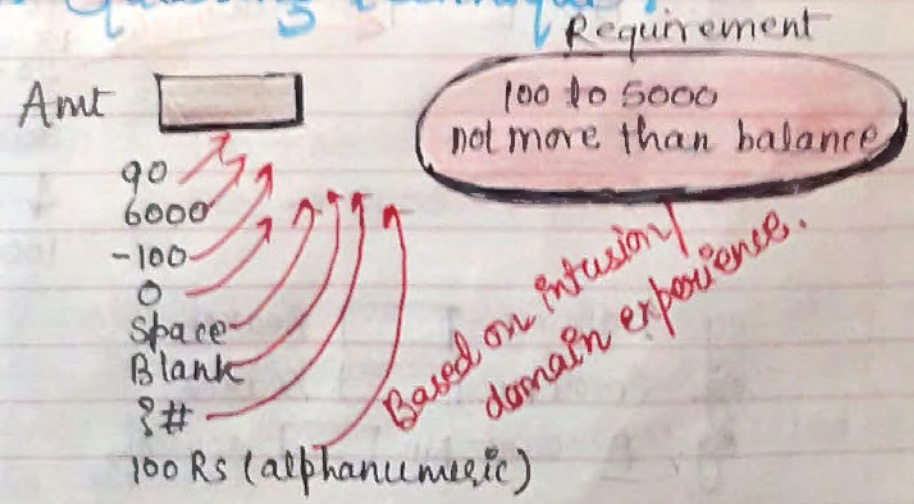
Author:
Reviewer:
Approved by:
Created date:
modified date:

If we randomly derive the scenarios for a specific requirement we can't ensure that we have covered maximum possible scenarios to overcome this we should follow testcase design technique while deriving the scenarios.

Test case design technique

- Error guessing technique
- Equivalence partioning
  - Pressmen
  - Practice
- Boundary value Analysis

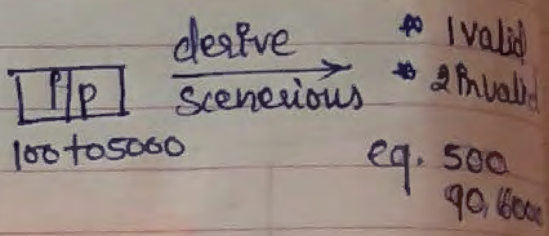
Error Guessing technique :



Defination: When we guess all possible error based on intusion or based on passed experience it's error guessing technique.

Equivalence Partioning technique:

- pressmen
- Practice



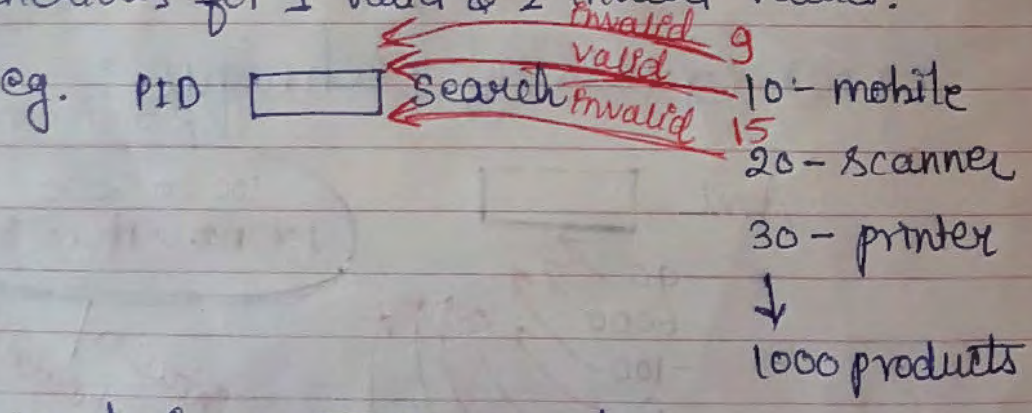
Pressmen technique:

According to him:

1. If the given I/p is the range of value then derive the scenerious for 1 valid & 2 invalid values.

   eg. age
   temp
   marks.

2. If the given I/p is the set of values then derive the scenerious for 1 valid & 2 invalid values.



eg. train no. [ ] search
eg. Enroll No. [ ] search
eg. Account No. [ ] search

③ If the I/p is boolean then derive the scenerious for both true & false value.

eg. gender ○ male
○ female

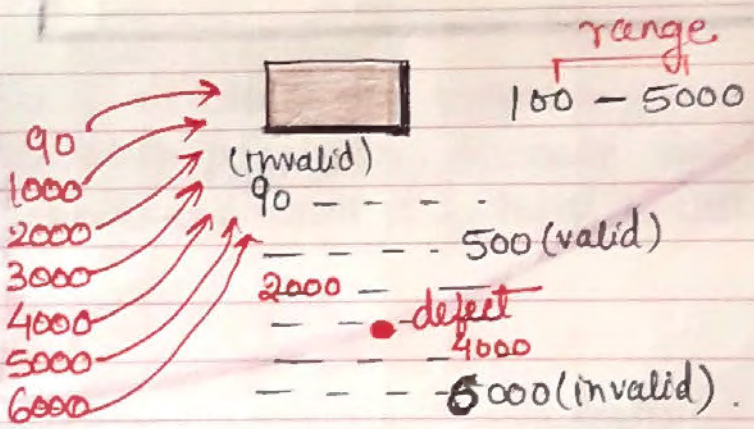
Submit

select – True
unselect & click on submit – false.

eg. ☐ I Agree.

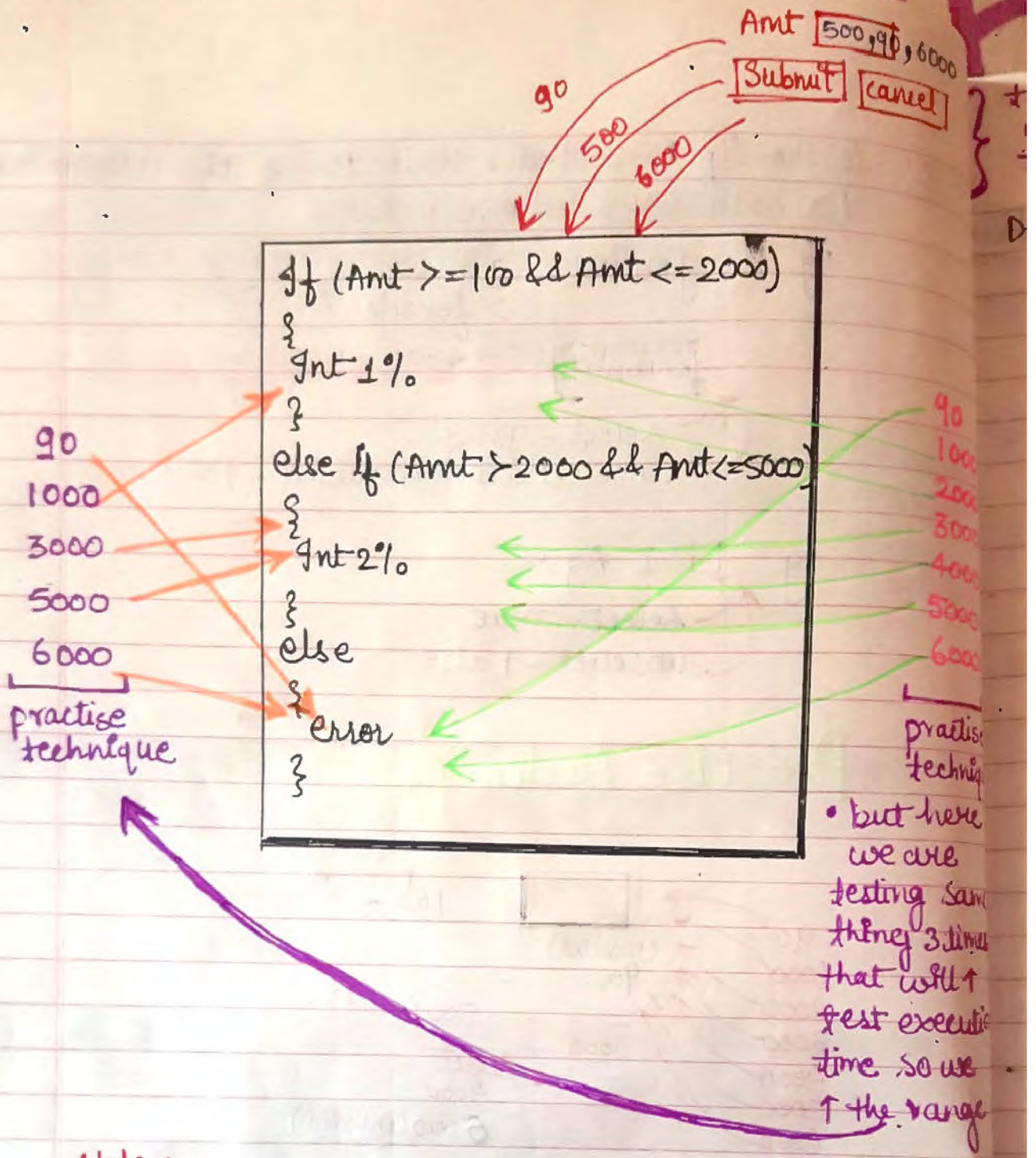
Select – True
unselect – False

Practise technique



Deviation in value { 100 → 2000 → 1% Intrest
2001 → 5000 → 2% Intrest

Defination: If the given Input is the range of value then divide the range into equivalent parts, test for all the values but make sure that you have tested atleast 2 Invalid values.



Note :

When ever there is deviation in the requirement go for practise.

Boundary Value Analysis (BVA) technique

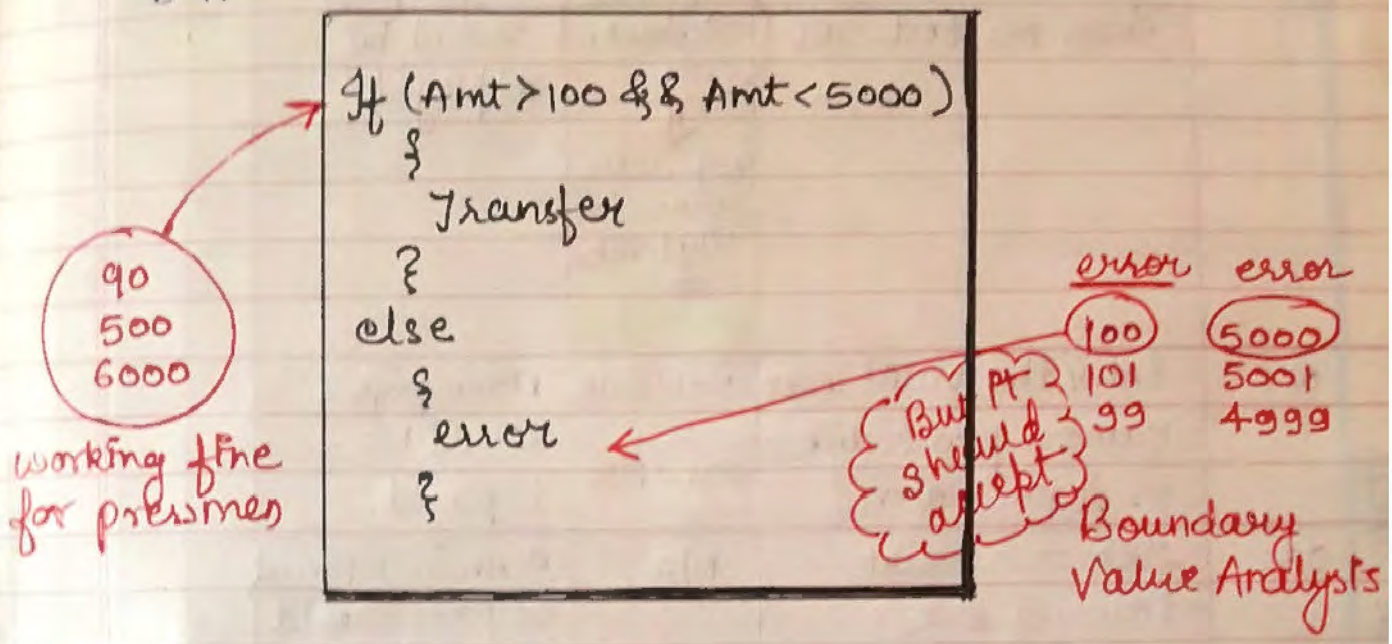
A-1 ↑ A ↓ A+1 to B-1 ↑ B ↓ B+1

eg. 100 – 5000

99	4999
100	5000
101	5001

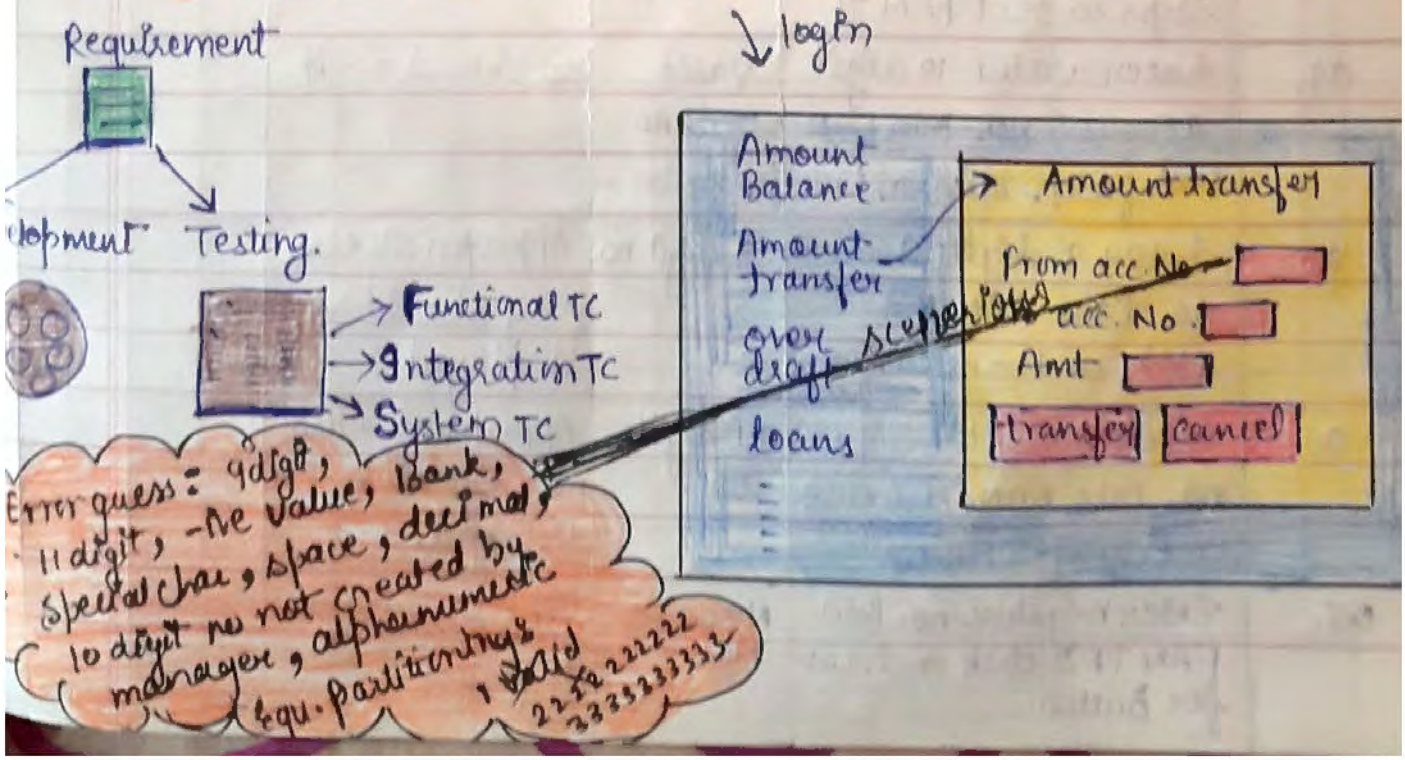
Testing by using pressmen technique:
we haven't tested else part of the program.

Defination: If the I/p is range of values between A to B, then derive the scenarios for A, A+1, A-1 & B, B+1, B-1.



Note:
When ever we go for boundary value analysis we can skip pressmen, because ~~BA~~ BVA itself have checked 1 valid & 2 invalid values.

Functional test case:



eg: functional test for amount transfer feature.

QA track Test ☐ ☐ ☐

Test Server

Step No.	Action / Description	Inputs	Expected Result	Actual Result	Status
01.	open the browser enter the test_URL	Hardcoded http://QA city bank.com ⇓ http://<server name> city bank.com (generic)	login page should be displayed.		
02.	Enter the valid user name & p/w & click on login button.	valid UN & valid p/w	Home page should be displayed.		
03.	click on amount transfer link	N/A	① Amount transfer page should be displayed. ② The following components should be displayed. * FAN TF * TAN TF * Amt TF * transfer button * cancel button.		
	following are the steps to test FAN TF				
04.	Enter valid 10 digit account no. into FAN text field & click on transfer	valid acc. no.	It should accept		
05.	Enter 9 digit account no, into FAN TF & click on transfer Button	9 digit no.	Appropriate error msg should be displayed		
06.	Enter 11 digit account no. into FAN TF & click on transfer Button	11 digit no.	Appropriate error msg should be displayed.		
07.	Enter negative no. into FAN TF & click on transfer Button	Negative No.	Appropriate error msg should be displayed.		

Navigation steps

Step No.	Action / Description	Inputs	Expected Result	Actual Result	Status	Comment
08.	without entering any value into FANTF click on transfer button.	Blank	Appropriate error msg should be displayed.			
09.	Enter special character into FANTF & click on transfer Button.	special character	Appropriate error msg should be displayed.			
10.	Enter the ~~no.~~ space in b/w the no. into FANTF & click on transfer Button.	space in between no.	Appropriate error msg should be displayed.			
11.	Enter alphanumeric no. into FANTF & click on transfer button.	alpha numeric no.	Appropriate error msg should be displayed.			
12.	Enter decimal no. into FANTF & click on transfer button.	decimal no.	Appropriate error msg should be displayed.			
13.	Enter 10 digit no but it is not created by manager.	Invalid acc. no.	Appropriate error msg should be displayed.			
	following are the steps to test TANTF					
14-26	same as for FANTF just change FAN→TAN					
27.	Enter the same account no. in TANTF as FANTF	~~valid~~ same acc. no	Should not accept			
	following are the steps to test Amt TF					
28.	Enter an integer no. with in the range 100-5000 into amt TF	positive integer no. b/w 100-5000	Should accept			
29.	Enter a no. less than 100 than ~~5000~~ in amt TF	~~→100~~ 100 ~~→5000~~	Should accept			
30.	Enter 5000 in amt TF	5000	Should accept			
31.	Enter no. less than 100 in amt TF	~~no<100~~ 99	Should not accept & appropriate error msg should be displayed			
32.	Enter no. greater than 5000 into amt TF	~~no>5000~~ 5001	Should not accept & appropriate error msg should be displayed			
33	Enter any negative no. into Amt TF	negative no.	Should not accept & appropriate error msg should be displayed.			

Amt No. → BVA → 99 ✓ 1999 ✓
100 ✗ 5000 ✗
101 ✓ 5001 ✓

Step No.	Action / Description	Inputs	Expected Result	Actual Result	Status
34.	Enter a decimal no. into amt No. text field	decimal no	Should no. accept decim & appropriate error msg should be displayed.		
35.	Enter a alphanumeric no. into amt No. text field	alphanum eric no.	Should no accept & appropriate error msg should be displayed		
36.	Enter special character into amt No text field	special character	Should not accept & appropriate error msg should be displayed.		
37.	~~Enter~~ Don't enter any thing in amt No. TF	blank	Should not accept & appropriate error msg should be displayed.		
38.	Enter space in b/w no. in amt No. TF	space in b/w no.	Should not accept & appropriate error msg should be displayed.		
39.	Enter 101 in amt No TF	101	Should accept		
40.	(boundary value analysis) Enter 4999 in amt No. TF	4999	Should accept		
	following are the steps to test transfer button & cancel button.				
41.	If all the fields - TAN, FAN & amt TF is ~~filled~~ entered with valid ~~data~~ value, click on to transfer. (mandatory fields should Before click be filled)	FAN, TAN, amt TF	Should navigate to next page & conformation msg should be displayed.		
42.	If delayed in entering the clicking transfer button.	N/A	Time out msg should be displayed.		
43.	click on cancel button.		Reconfirmation msg should be displayed.		
44.	click on cancel button after reconfirmation.		Should navigate to next page.		

Note:

* While writing test cases we should always start with navigation steps.
* we should use words like "should be displaye" / "must be displayed."
* we should not hard code the test case we should write generic TC.

eg: Integration test case for amount transfer between 2 account.

Step No.	Action/Description	Inputs	Expected Result	Actual Result	Status	Comm
1.	Open the browser, enter the test URL	https://<server name> city bank. com	login page should be displayed.			
2.	enter valid username & password of user A & click on login button	valid UN & PWD for A.	Home page should be displayed.			
3.	click on amt transfer link	N/A	amt transfer page should be displayed.			
4.	Enter valid data into FAN, TAN & amt TF & click on		confirmation msg should be displayed.			
5.	click on amt balance link		The transfer amt should be deducted.			
6.	Click on transaction link		The transaction details should be displayed.			
7.	Click on logout link		logm page should be displayed			
8.	Enter valid username & pwd of user B & click on logm Button	valid UN, pwd of B	Home page should be displayed.			

9	click on Amt Balance link		The transfer amt should be added in B account.		
10.	click on transaction link		The transaction details should be displayed.		

Header :-

Test case Name : CityBank_Amounttransfer_checkAB&transaction
(or) CityBank_Amounttransfer_IntegrationTC
(or) CityBank_Amounttransfer_all possible scenerios

Req.No: CBO_SRS 30.1.1

severity: critical/major/minor.

Test data: username, p/w & account No. of user A & user B.

Pre-condition: - main balance should be there in A's account
- user A's & B's acc. no. should be created

Testcase type: Integration TC (+Ve & -Ve)

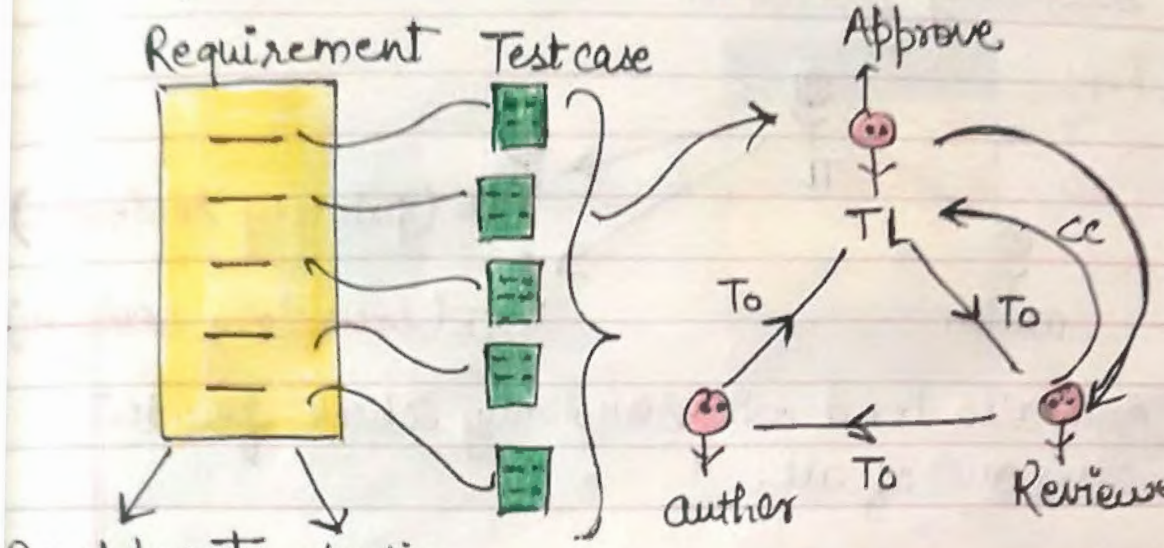
Brief description: This test case consists of all possible Integration scenerious on amt. transfer feature.

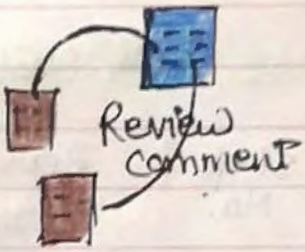
Assignment-8

1). Write 5 test cases for gmail.
2). Write system test case on OD feature.

Best test case should have following
① easy to understand
② cover all possible scenerious
③ It should take very less time for execution.
④ should not have repeated & wrong scenerious.

Test case review process :



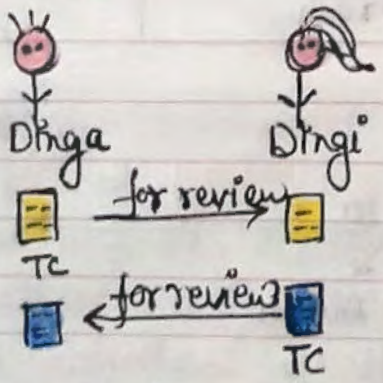


* missing scenerios
* wrong scenerios
* Repeated scenerio

Q. How TL will allocate the test case for reviewing ?

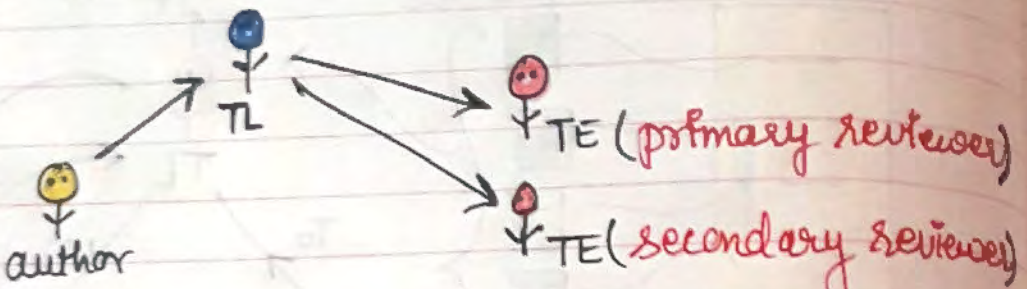
- Based on domain knowledge of TE
- one who works on similar module (home loan - personal loan)
- assign the test case to one who takes responsibility.

Q. How do you ensure that test Er. will review test cases properly ?



both r frnds so they may not review

Ans. assign same testcase to both primary & secondary reviewer.



And also lead will randomly check few test cases out of all.

Test case Review template

S No.	Test case Name	step No.	Reviewer Comments	severity	Authors status	comment
01	Gmail_compose	01	test data is missing	high	No fixed	Test data is not re.
02	Gmail_Compose	03	checking sent mail in SI scenerious is missed	critical	fixed	required
03.	Gmail-compose	08	wrong scenerious checking sent mail in trash	high	fixed	

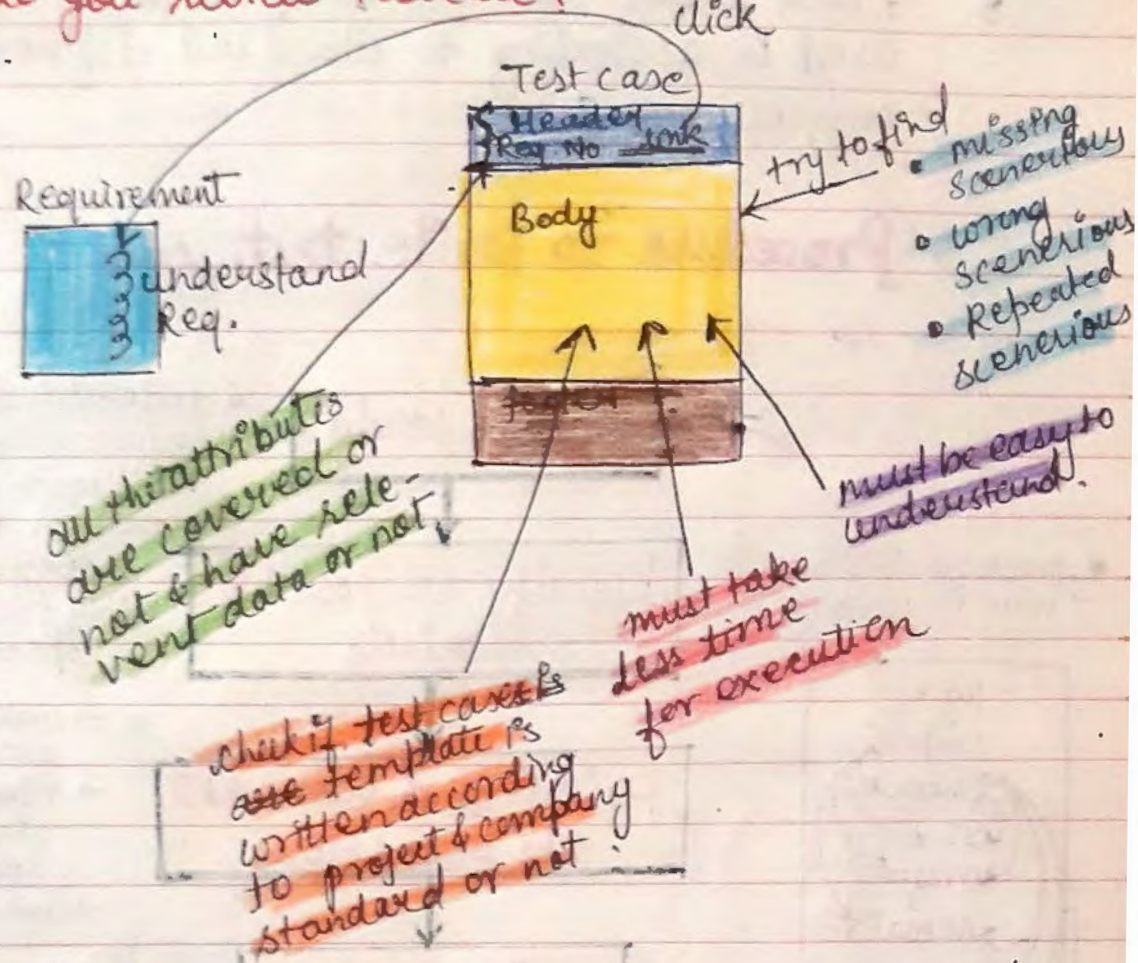
After fixing the issue

Sent to

Reviewer / Author

- Must be approved by TL

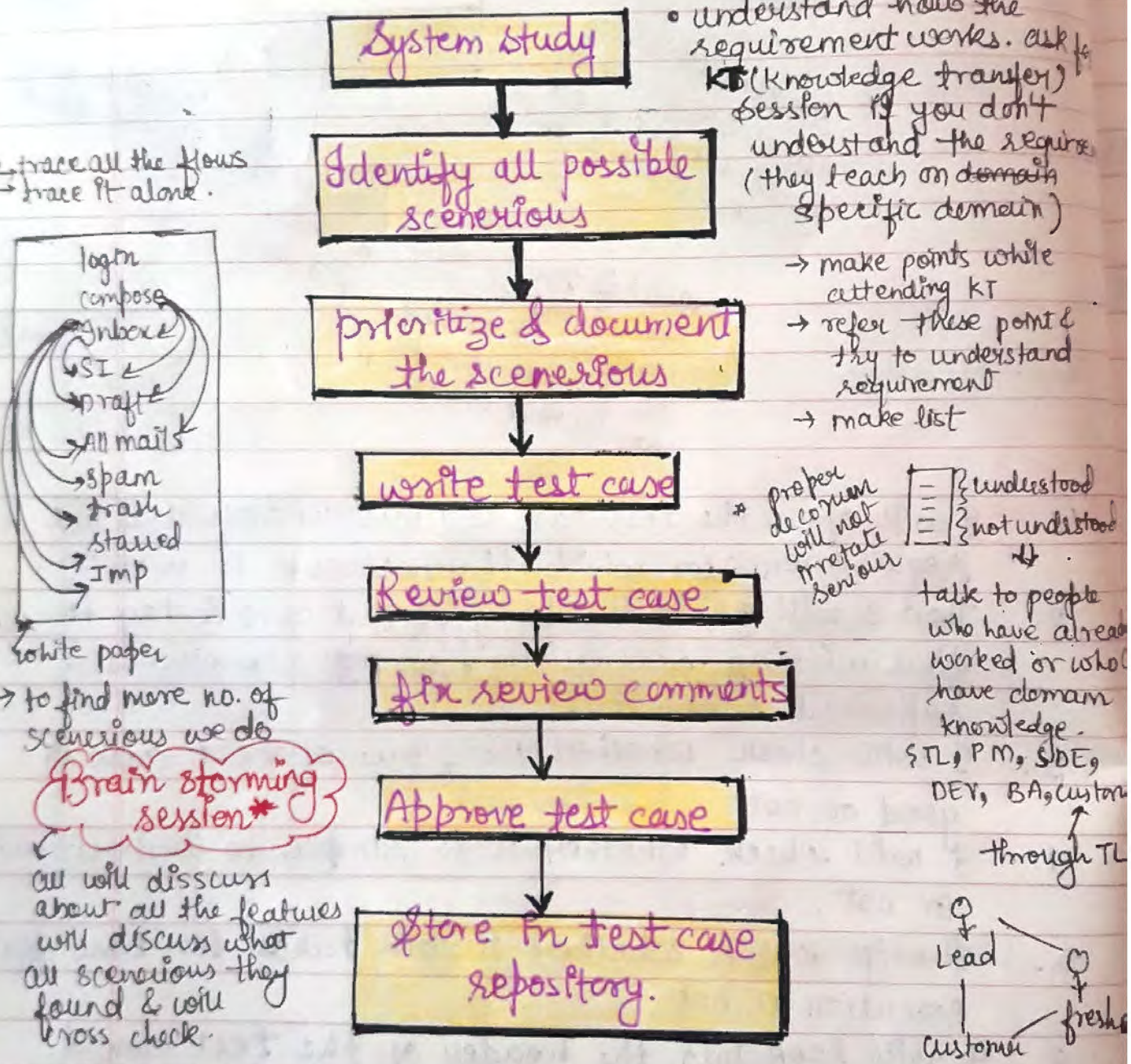
Q. How do you review Testcase?



1. I will open the test case & I will understand the requirement for which the test case is written.
2. Then I will go to the body of Test case & try to find missing scenerious, wrong scenerious & repeated scenerious.
3. I will check whether the flow of test case is good or not.
4. I will check whether it is simple to understand or not.
5. I will check whether it will takes less time for execution or not.
6. I will look into the header of the test case & check whether all the attributes are covered or not.
7. I will check whether all the attributes in the header is having relavent data or not.

8. I will check whether template of the test case used is according to standard defined in the project or not.

Procedure to write test case :



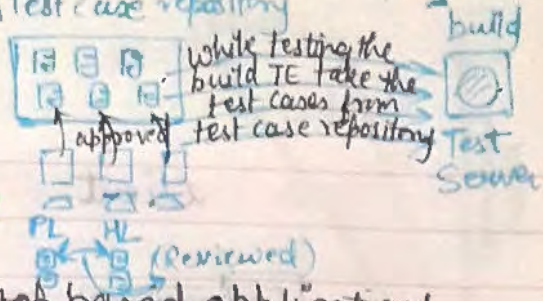
lead will say - to ↑ the % of scenerios : so every body will be in pressure & work

* while identifying the scenerios we do brain storming meeting. It is a meeting conducted by testing team where in they disscuss all the features & find all the missing, wrong, repeated scenerios & will try to ↑ the no. of scenerios & also prioritise them

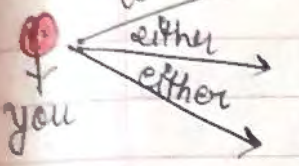
Repository → storage place / Database

here we store approved test cases

D:\Test case repository



Types of Application:

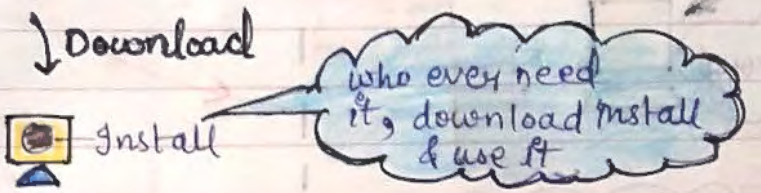


1). Stand alone / Desktop based application / windows application

2). client / server application.

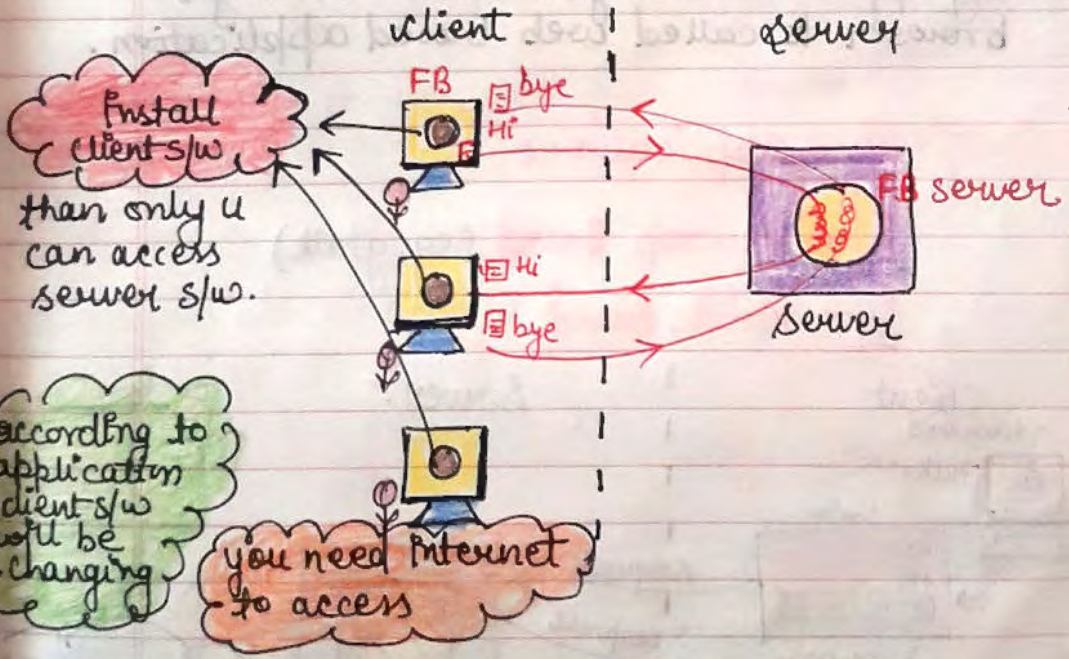
3). web based application.

1 eg. Notepad++, calc, autocad, ms-office, paint

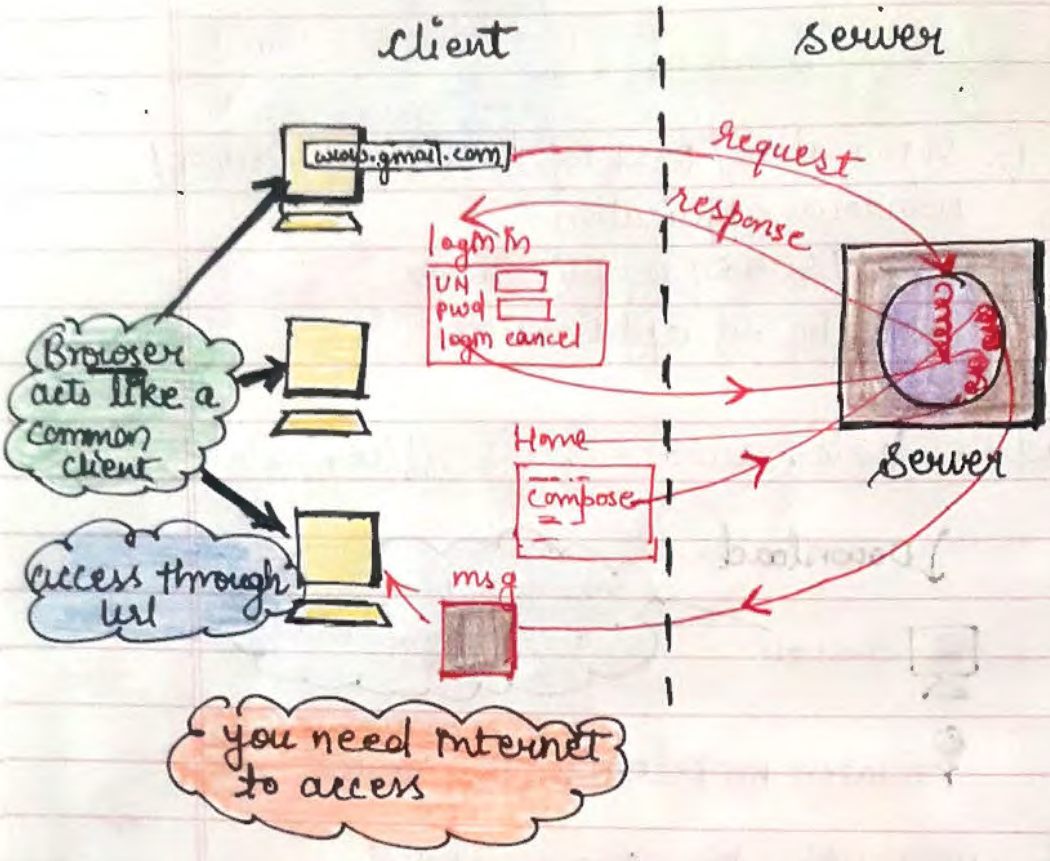


(later no Internet is needed

2 eg. Ola cab, whatsapp, Instagram, skype



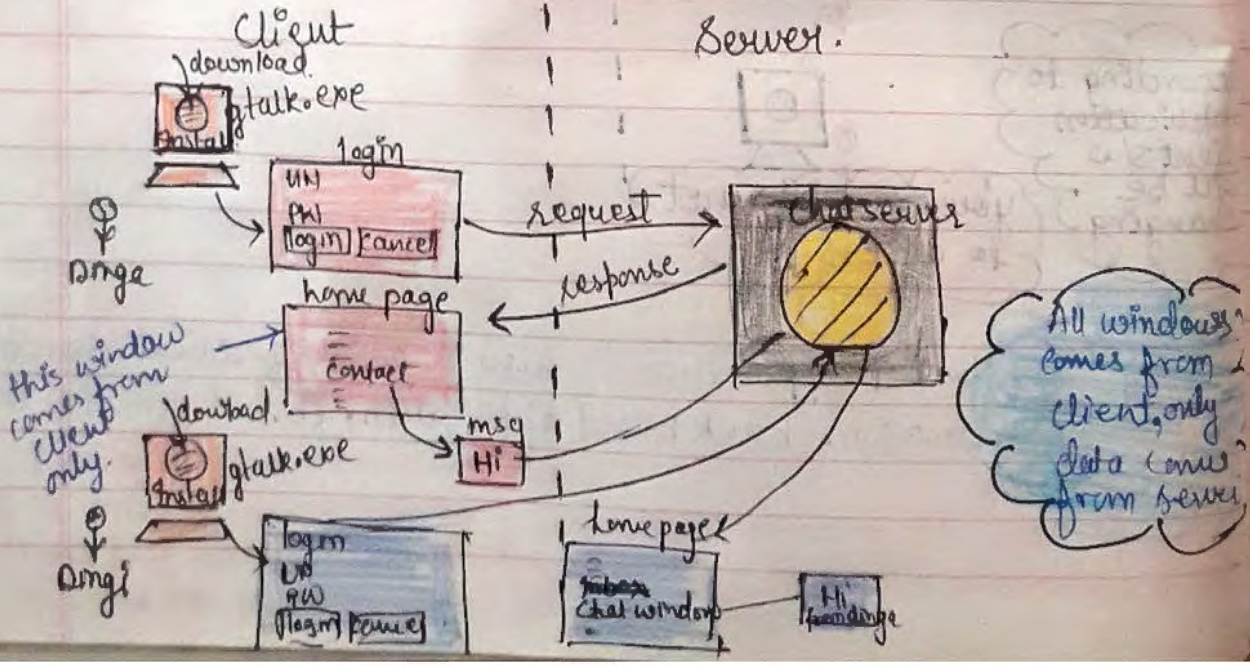
3 eg. web based application is also a part of client-server app.
- amazon, bank based application, fb.



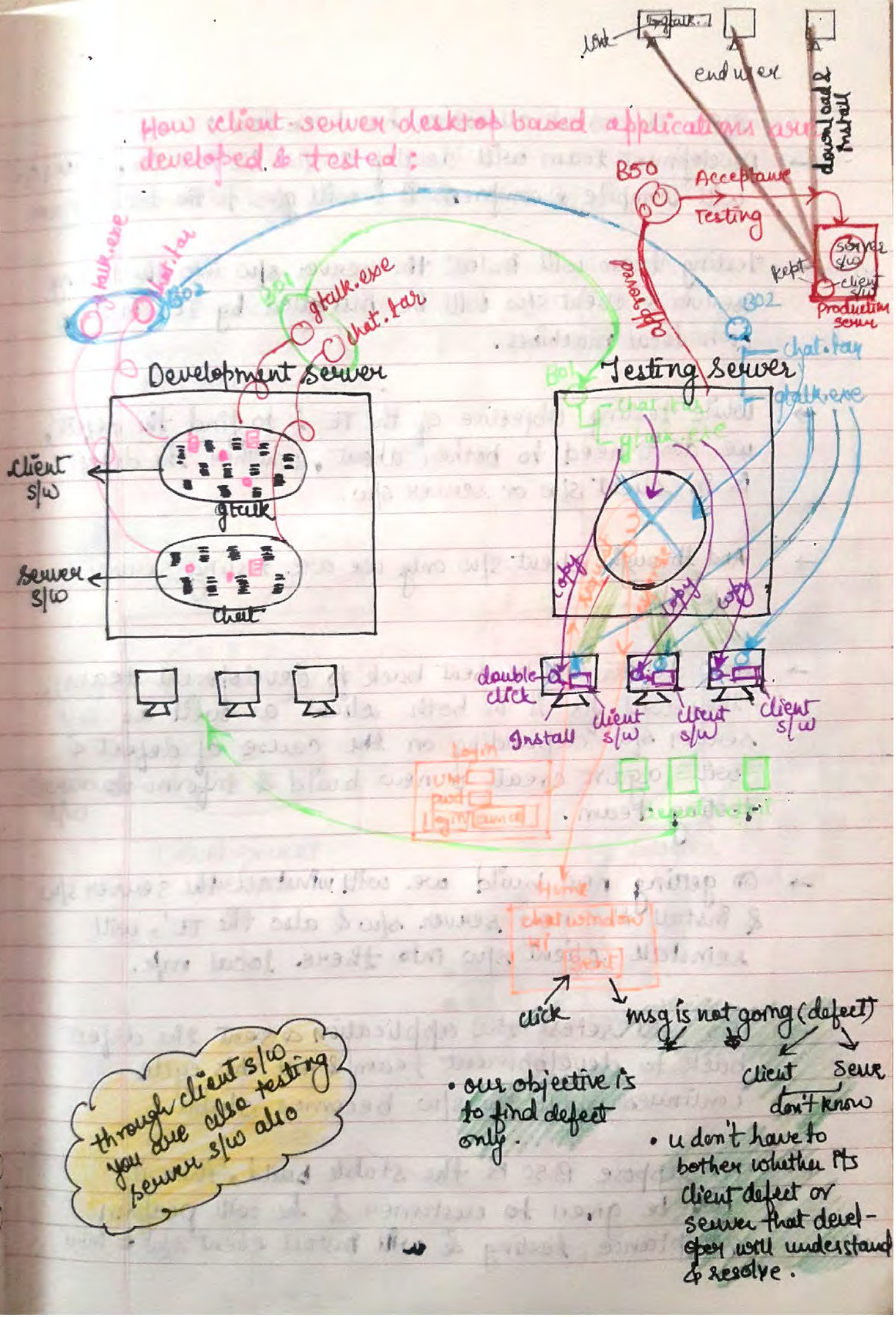
* Any application that you access through browser is called web based application.

Client/server application:

TYPE → Desktop Based (eg. gtalk)
TYPE → Mobile Based



How client-server desktop based applications are developed & tested:



through client s/w you are also testing server s/w also

- our objective is to find defect only.
- u don't have to bother whether its client defect or server that developer will understand & resolve.

After the work allocation has been done :

→ Development team will develop 2 s/w's (client s/w, server) will compile & compress it & will give to the testing team

→ Testing team will install the server s/w into the testing server & client s/w will be installed by TE's in there own local machines.

→ While testing objective of the TE is to find the defect, we don't need to bother about, whether the defect is in client s/w or server s/w.

→ And through client s/w only we are testing server s/w also.

→ Now defects will be sent back to development team they will fix it in both client as well as server s/w depanding on the cause of defect & will again create a new build & inform to testing team.

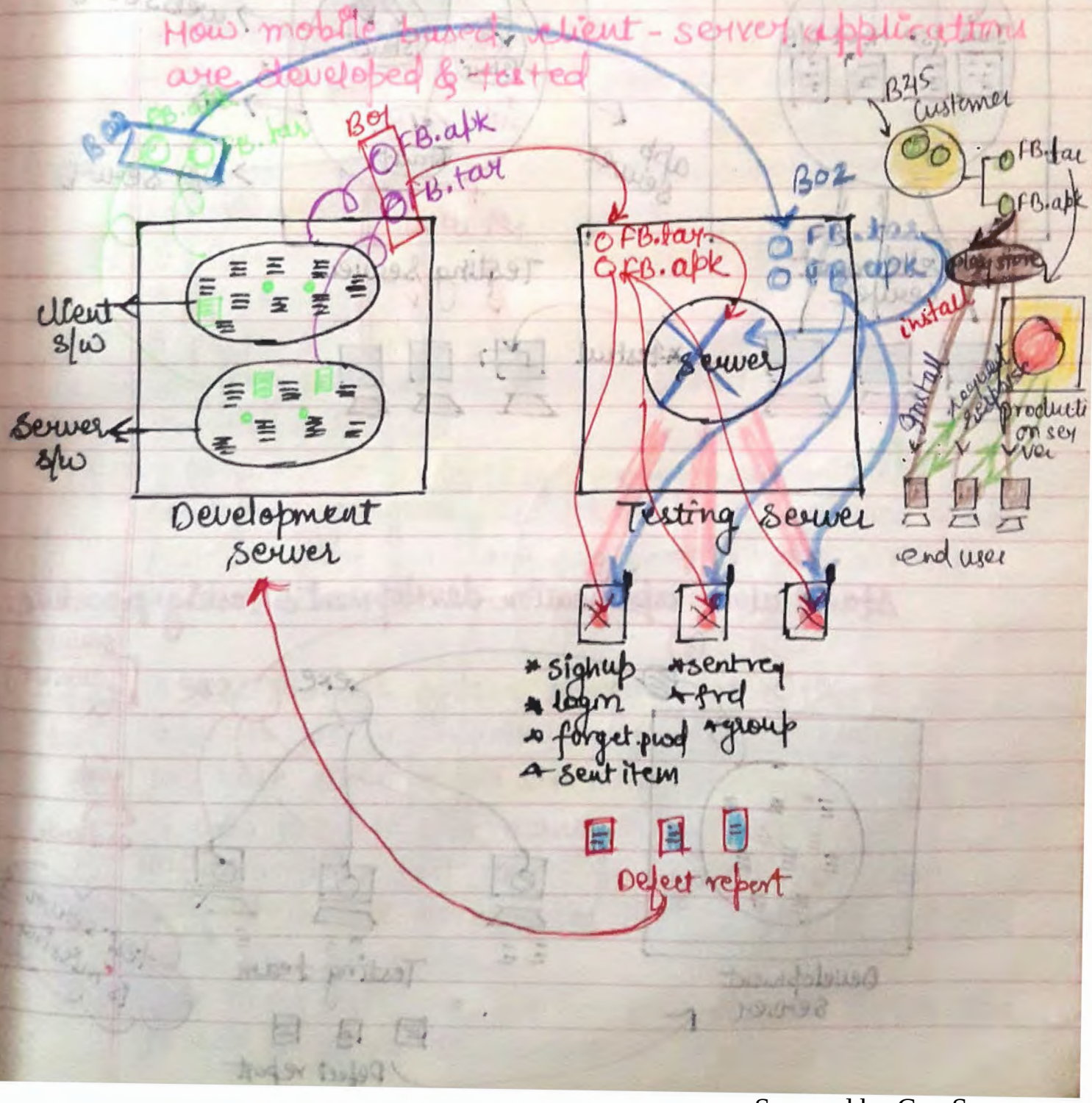
→ On getting new build we will uninstall the server s/w & install the new server s/w & also the TE's will reinstall client s/w into there local m/c.

→ TE's will retest the application & sent the defect back to development team & ~~till~~ this cycle continues uptill the s/w becomes stable.

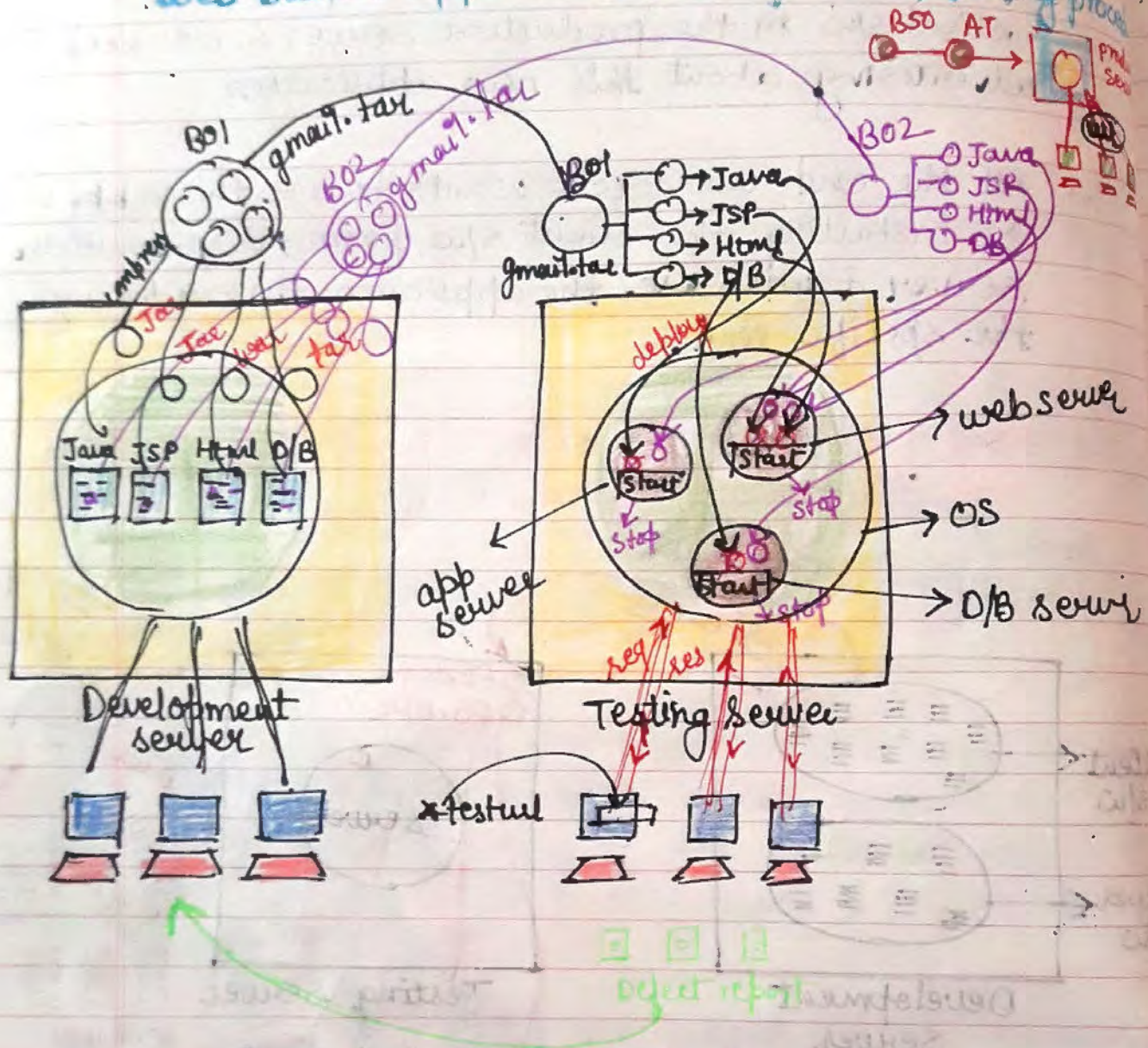
→ For suppose B.50 is the stable build, then it will be given to customer & he will perform acceptance testing & will install client s/w & into

production server & will also ~~ket~~ keep (store) client s/w in the production server & will keepon advertising about this new application.

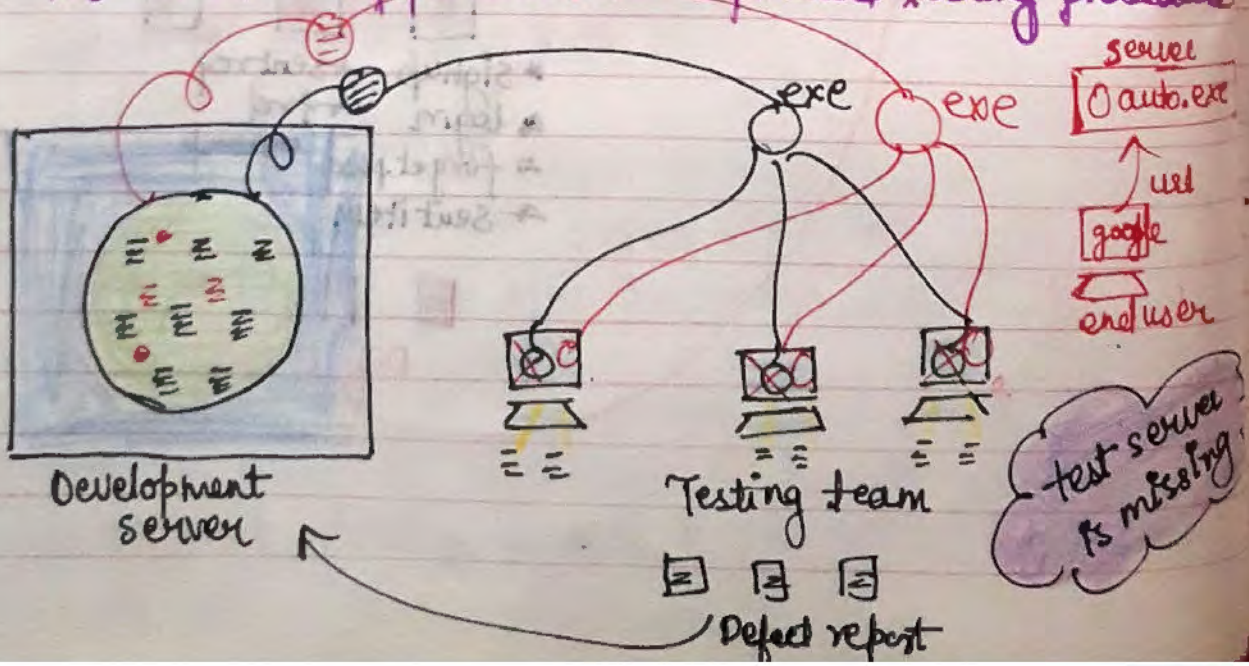
→ All the end users who want's to use this app, will be installing the client s/w from this production server & will use the application throug through the specific link.



Web Based application development & testing process



Stand alone application development & testing procedure



Q. How do you convence ur manager that you have tested every thing?

(OR)

How do you ensure that ur test coverage is good?

- followed strick procedure to write test case
- used test case design techniques.
- Traceability matrix
- CCA (code coverage analysis)
- adhoc testing (randomly testing)
- after getting build also I got some more scenerios.

Ans. My test coverage is good becoz I followed certain activities while writting test cases:

1. I followed strick procedure to write test cases so my test coverage is good.
2. I prepared traceability matrix to ensure that every requirement has got atleast 1 test case so. my coverage is good.
3. I followed code coverage analysis technique & identified programs which are not covered while test execution for those program I have written the test cases.

1a). I did through system study becoz of that I was able to derive maximum possible scenerios.

b). we have doen brain storming session becoz of that I was able to find many missing scenerios.

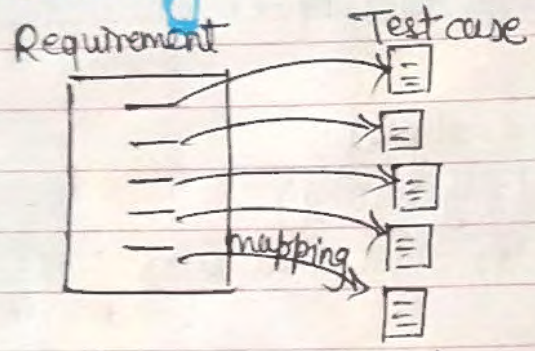
c) while writting test cases I applied test case design technique because of that I was able to find accurate scenerios.

d) while doing test case review process, we were able to add missing scenerios

e). While doing test execution & found new scenerios & added back into the test cases.

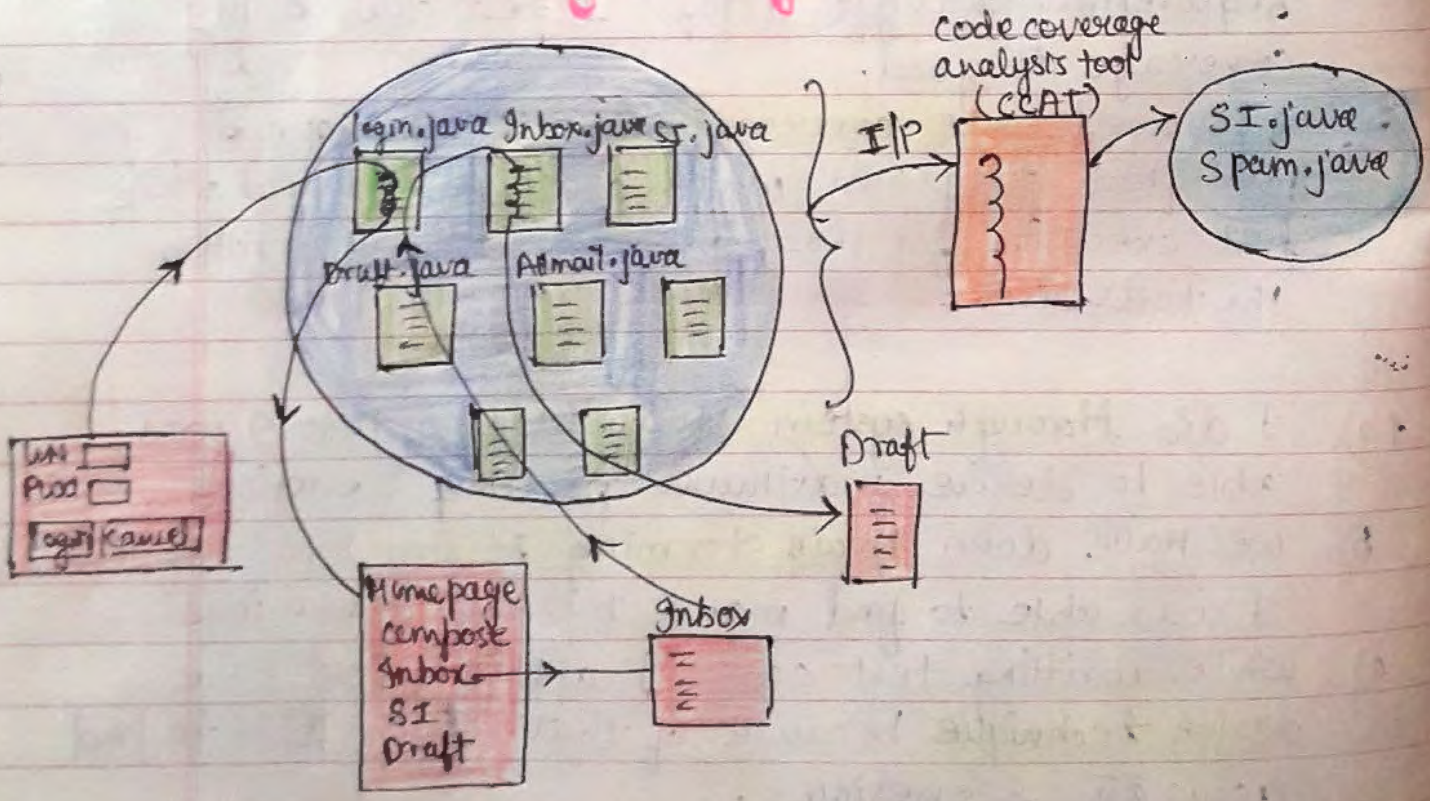
f). & also performed adhoc testing because of that our test coverage is improved.

Traceability matrix:



It is a document which ensure that each requirement has got atleast 1 test case & it is also k/s mapping the test case

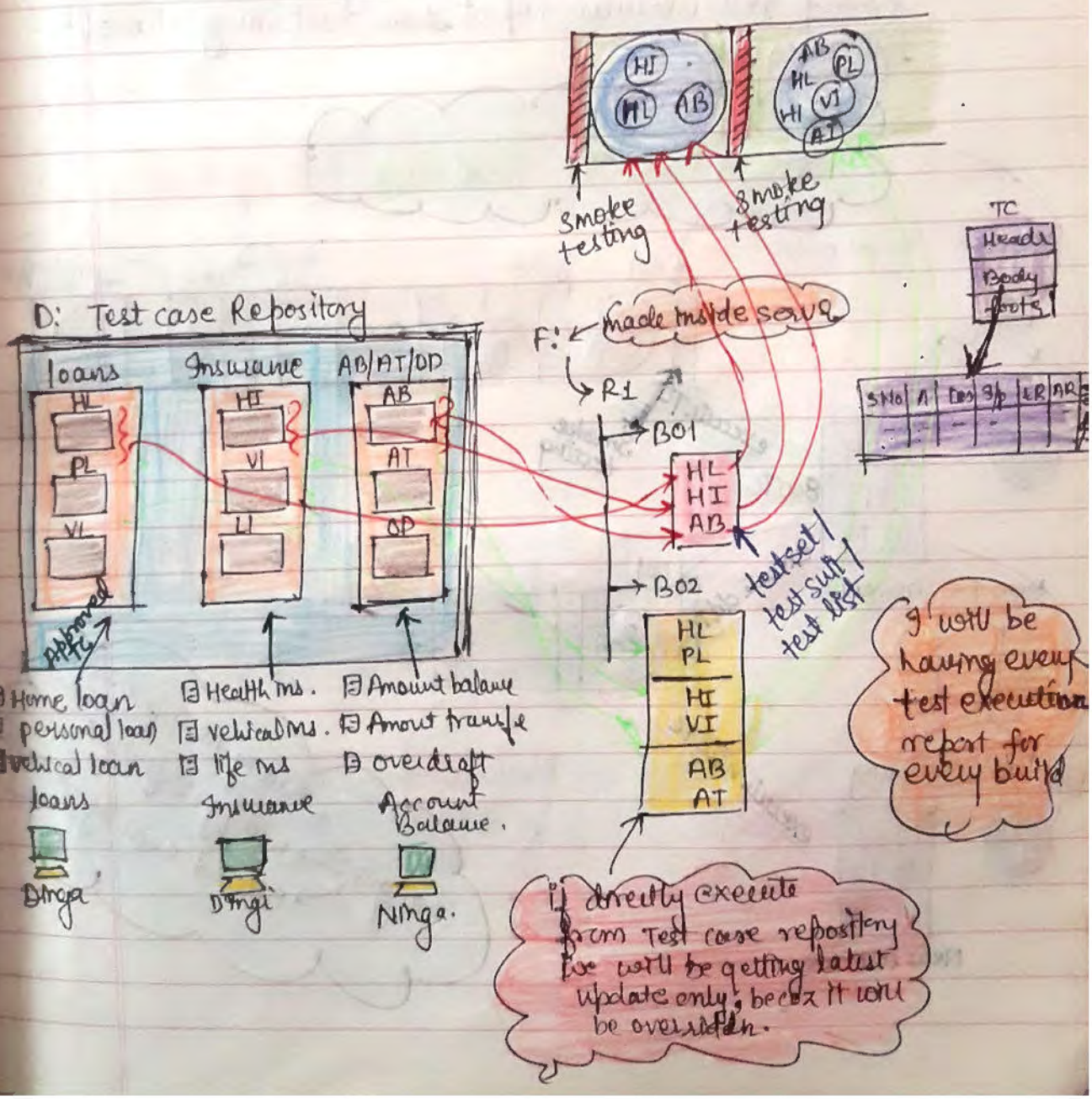
Code coverage analysis:



- testing old feature is called regression testing, by executing old test cases we do regression testing.

There might remain some program, which will not be executed & identifying it manually is difficult so we use coverage analysis tool, - this tool will list all the program while that which have not been executed. It may happen because we might have missed that test scenerious. So after doing code coverage analysis we may find some more missed scenerious.

Test Execution Process:-

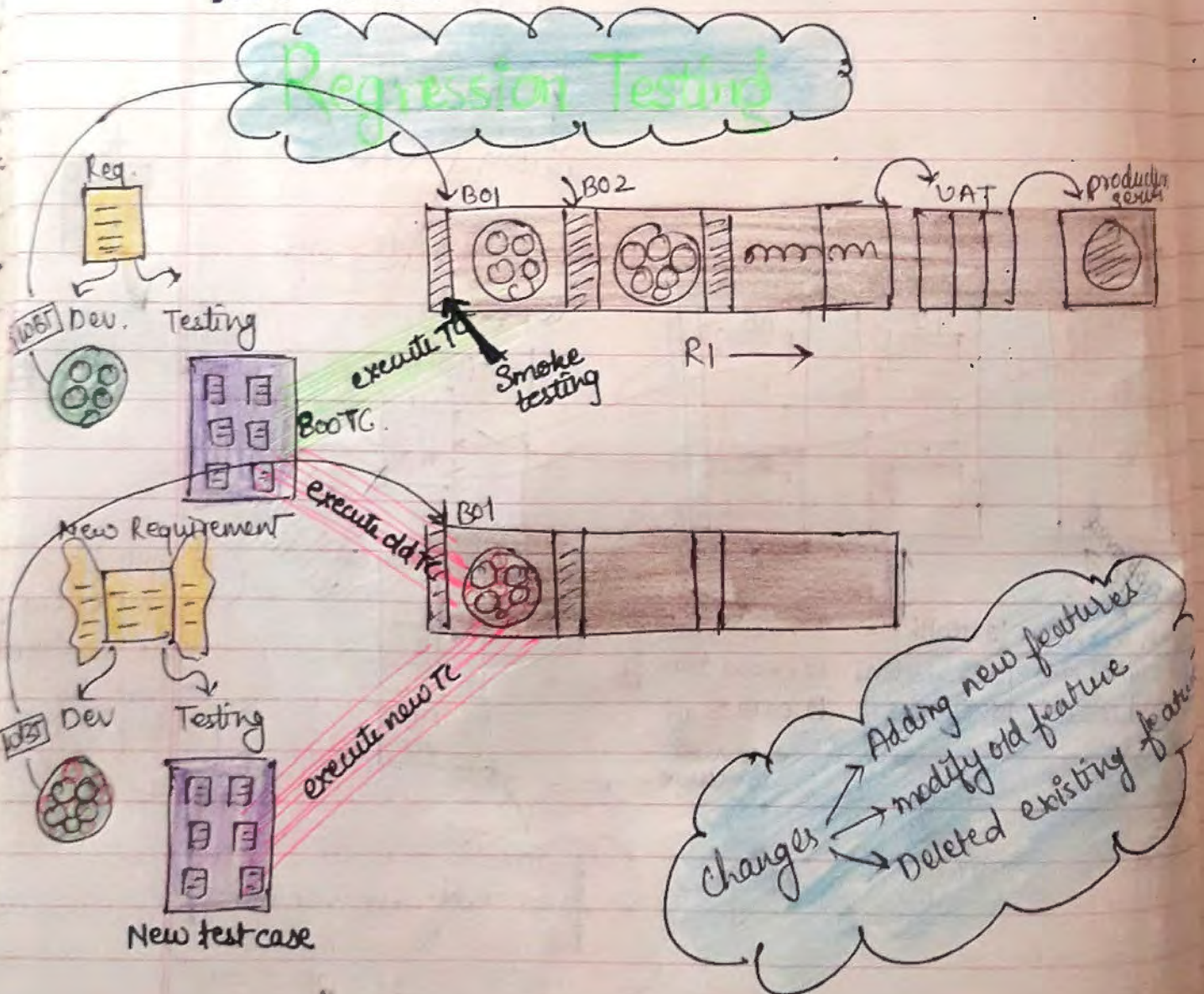


→ All the approved test cases will be stored in the test case repository.

→ Now when the developers give the build, tester will be copying the respected module test cases from the test repository and create test suit / test set / test list.

→ Now test er's will test the application by executing test cases. & will fill the actual result & status field in ~~that~~ it will sent the report to developers regarding defect.

→ This cycle will continue & for every build we will be having test execution report ~~conti~~ containing the old test cases also.



Defination:

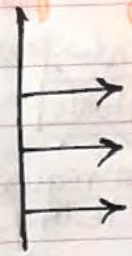
Testing the unchanged features to make sure that it is not broken because of changes is called regression testing.

The changes would be addition, modification, & removal of features or defect fixes.

(OR)

Re-execution of same test cases in different test cycles or releases to ensure that changes are not introducing the defects in unchanged features is called regression testing.

Types of regression testing:



- Unit regression testing (Testing only the changes)
- Regional regression testing
- Full regression testing

1. UNIT REGRESSION TESTING:

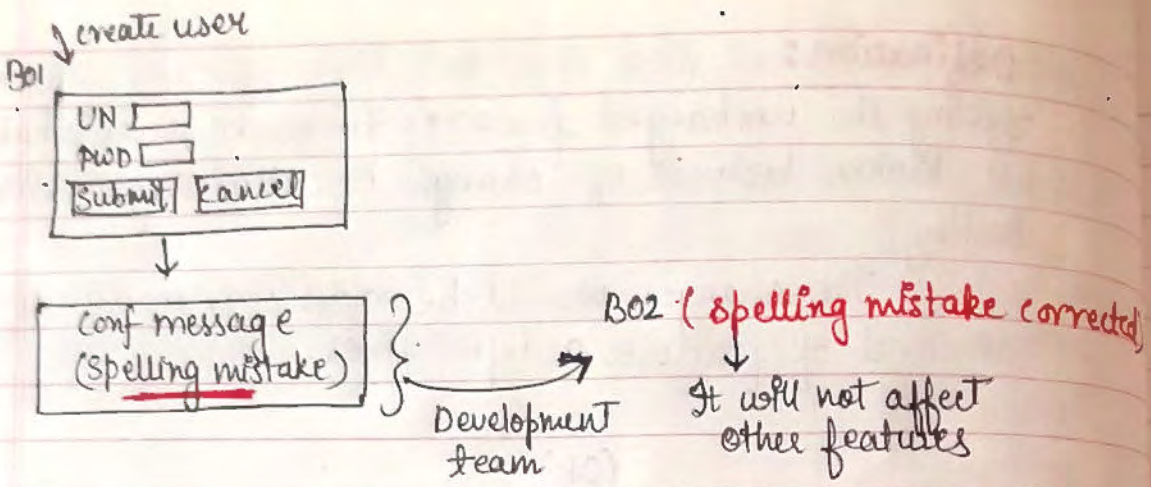
Defination:

Testing the changes or bug fixes is called unit regression testing.

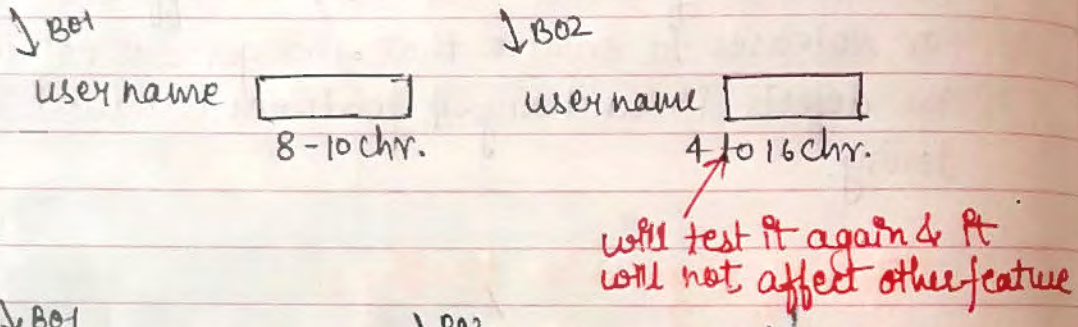
Note:

When ever we are assured that changes will not affect any other features we go for unit regression testing.

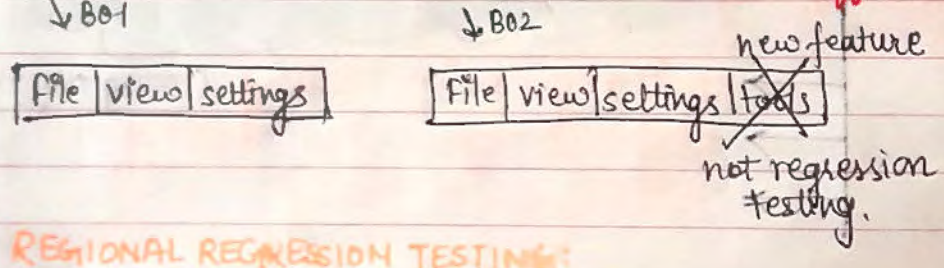
eg1.



eg2.



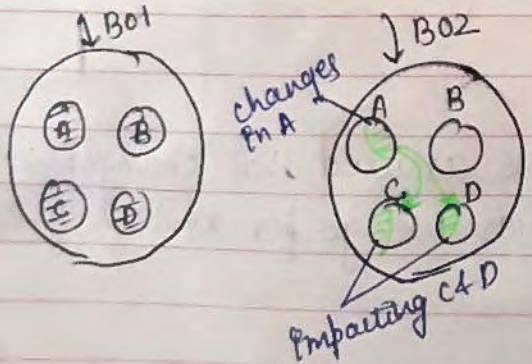
eg3.



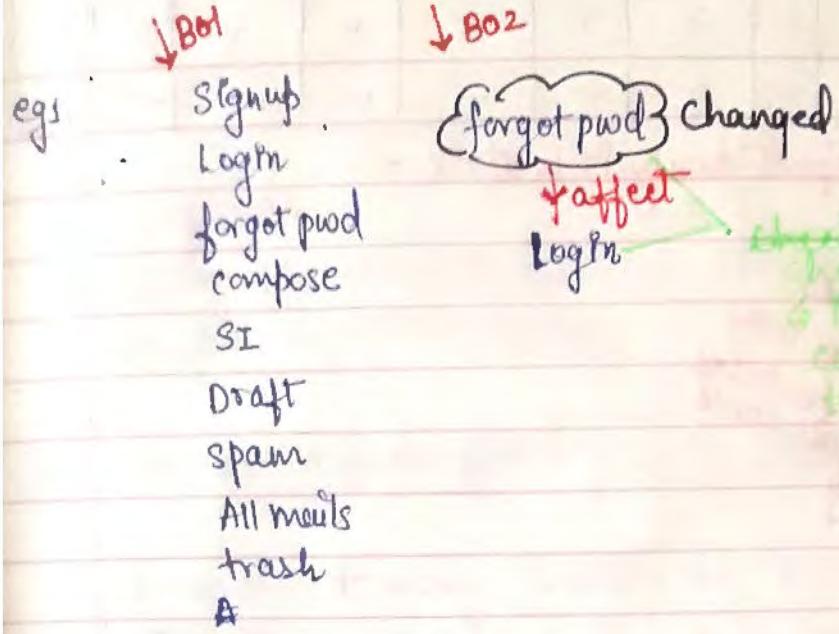
2. REGIONAL REGRESSION TESTING:

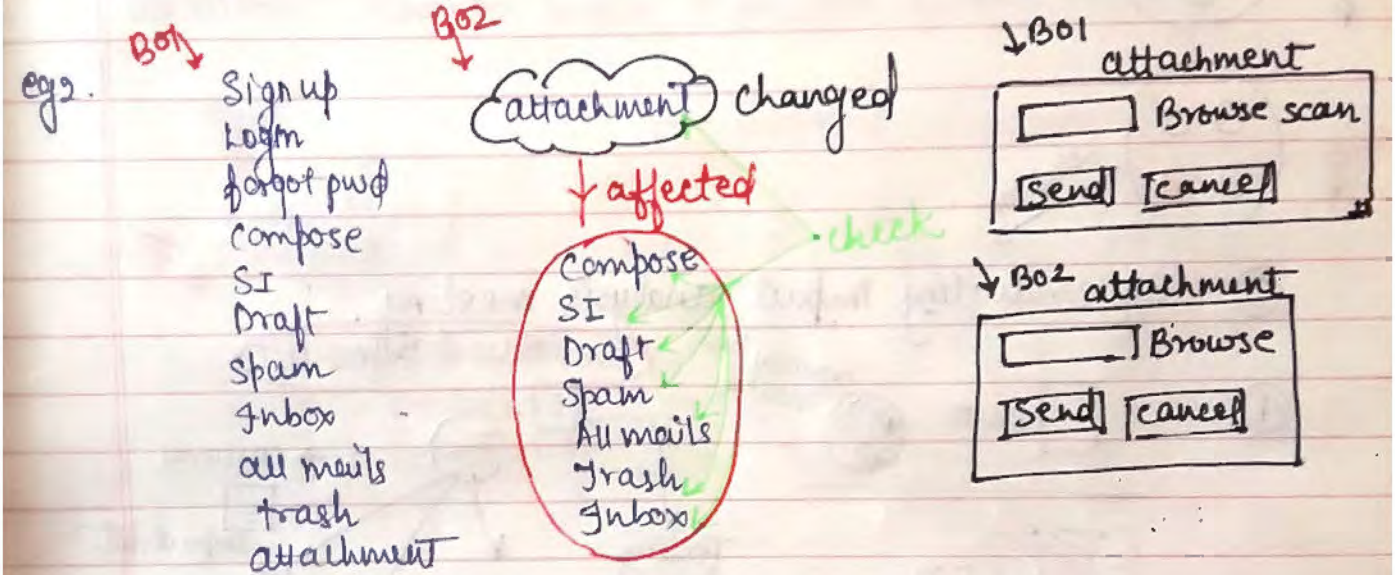
Defination:

Testing the changes ~~plus~~ and impacted areas is called as regional regression testing.



• based on domain knowledge we get to know impacted areas.





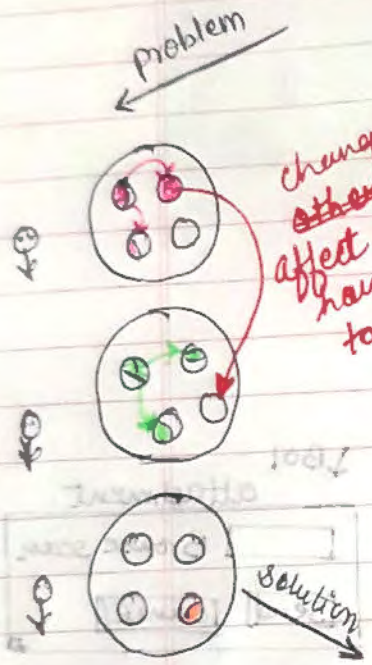
Q. How do you identify Impact area?
(OR)
How do you do Impact analysis?
(OR)
How do you pick regression test cases

① Based on domain knowledge: as a tester, I will be knowing how each & every module works & also how all the features are related, based on that knowledge I will be able to identify impacted areas

② By preparing impact analysis matrix.

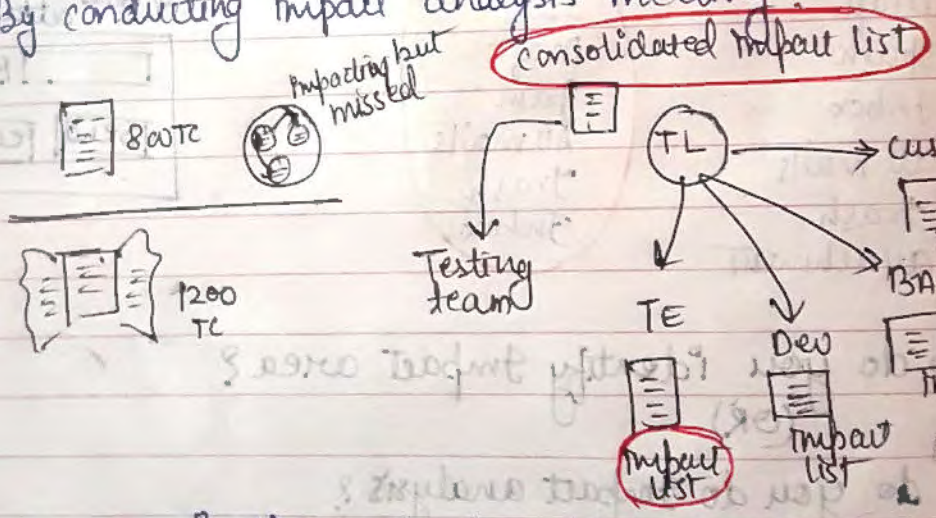
SNo.	Changes	Signup	login	compose	SI	Draft	spam	All mails	trash	starred	Smtp	Call
01	forgot p/w	X	✓	X	X	X	X	X	X	X	X	X
02	attachment	X	X	✓	✓	✓	✓	✓	✓	✓	✓	X

- Impact analysis matrix -



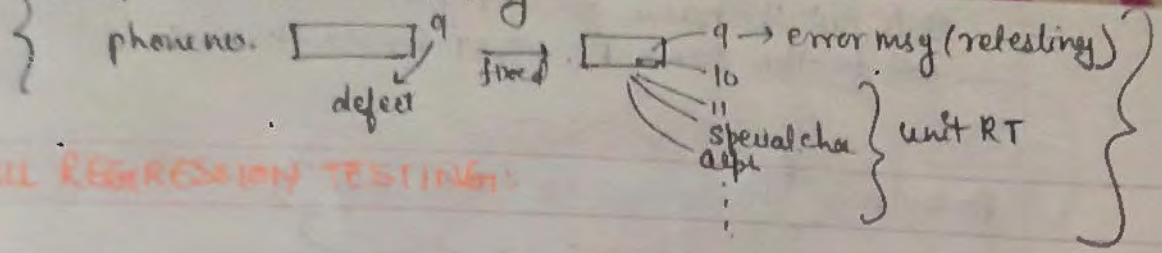
③ By conducting impact analysis meeting.

④



Communicating with test ~~lead~~ er, developer, BA & customer test lead will get the respective impact list ~~with~~ then he will make a consolidated impact list out of it & will give to testing team.

Diff b/w unit RT & retesting



④ FULL REGRESSION TESTING:

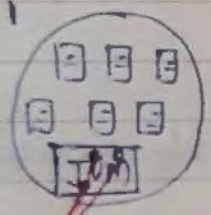
Defination:

Testing the changes and all the remaining areas is called as full regression testing.

Q. when do we go for full regression testing?

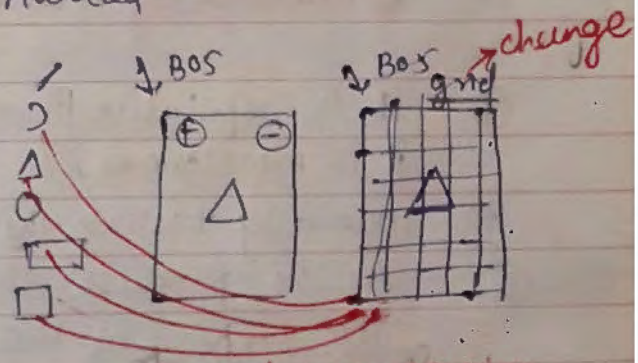
① when ever too many changes are made in the project.
② when ever changes happen in the root of the project.

eg:1



Changes in JVM, thus we need to do full regression testing.

eg2. Autocad



③ Before giving the s/w to the customer, for last few cycles we do full regression testing.
④ After every few cycles we do regression testing in order to ↑ the quality of the product.

Q. What is the difference between regression testing & retesting?

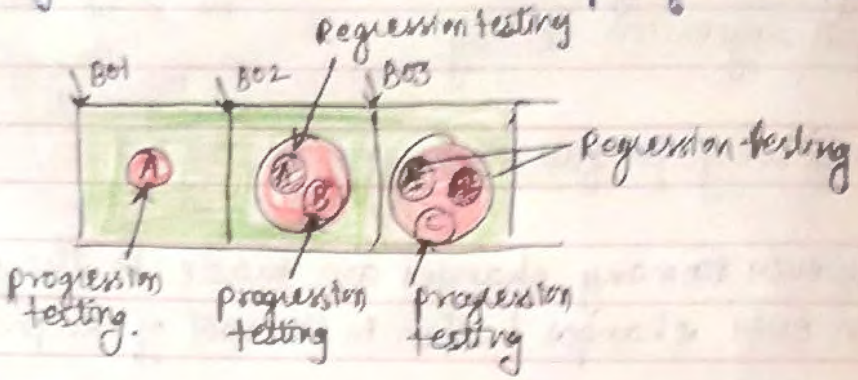
Retesting: check whether the bug is really fixed or not.
Regression testing: checking whether the unchanged feature to make sure that it is not broken because of changes.

It is just the name given regression testing, In regression testing we do the same UT, IT, ST for old features.

Drawback → **Progression testing**

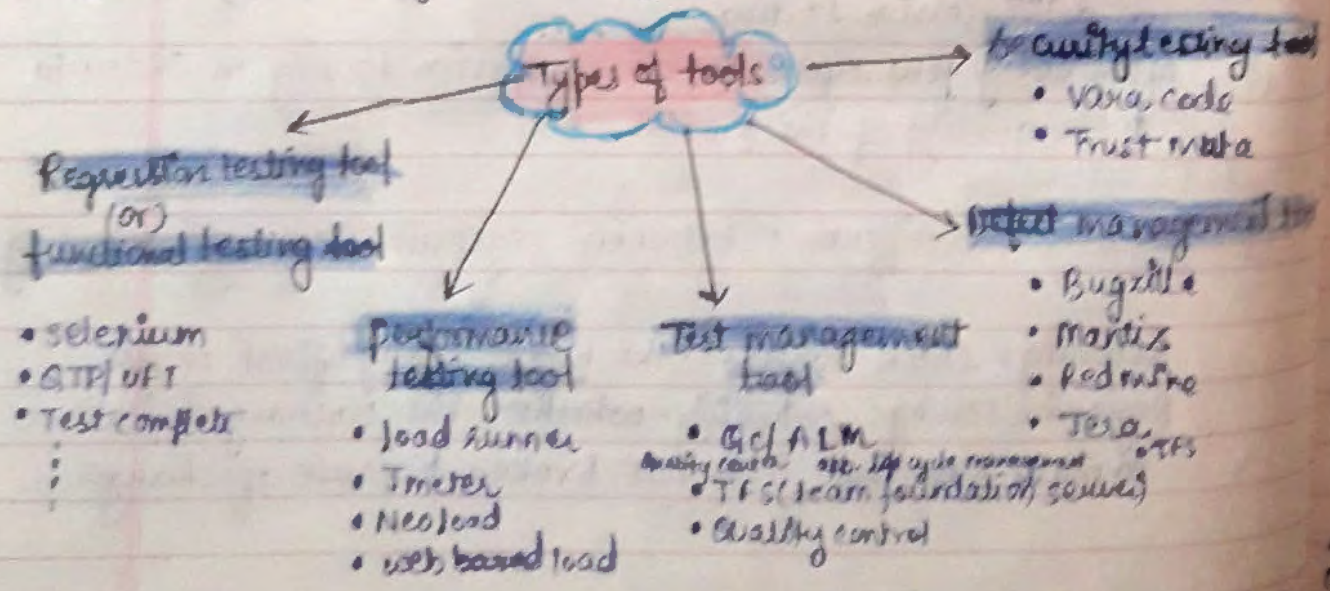
Defination:

Testing the new module is called progression testing.

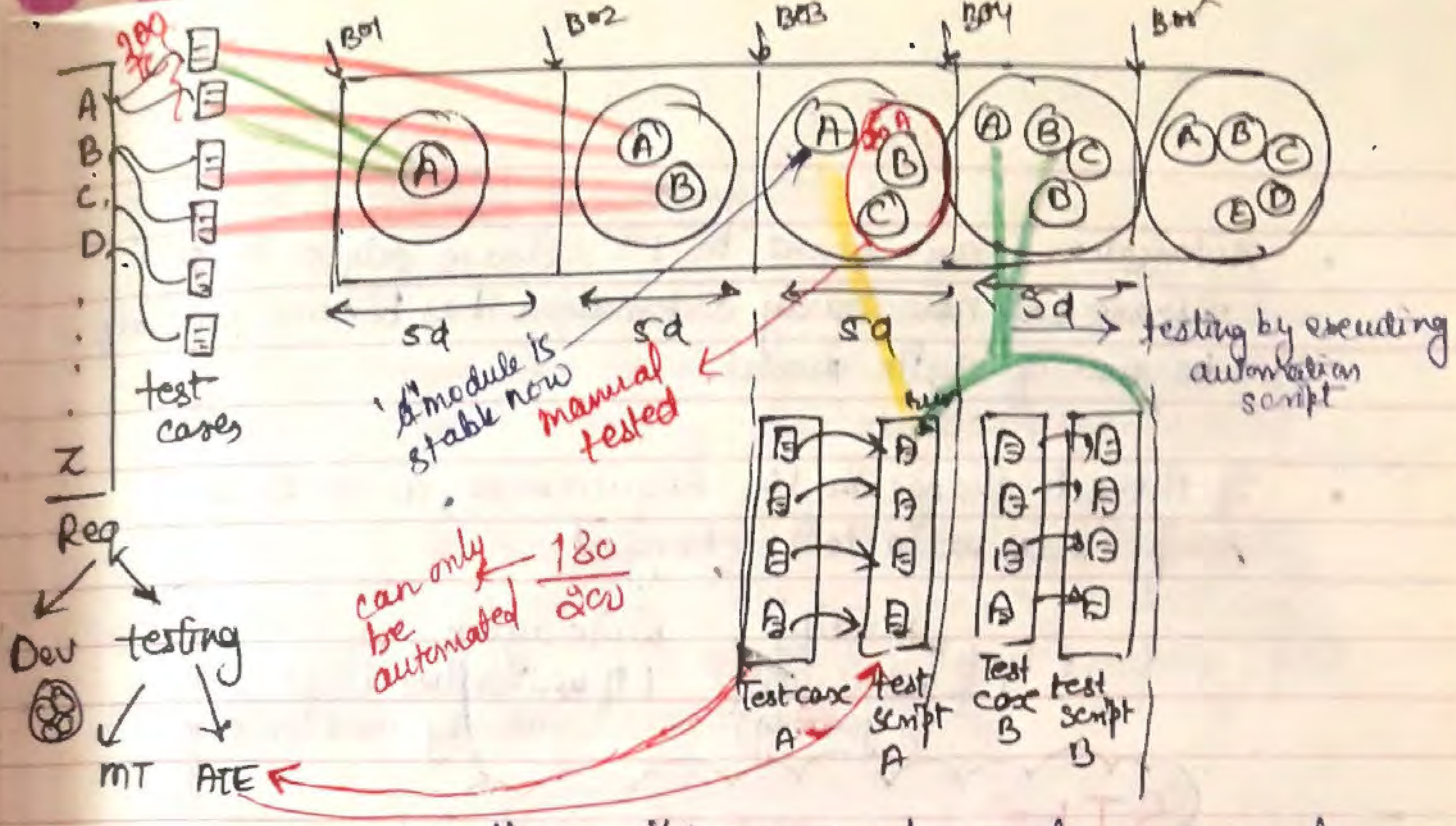


Drawback of manual testing:

1. It is repetative in nature over the period of time, It is monotonous.
2. manually testing the slw always consumes more time.
3. As the product size ies the test cycle duration also ies becoz of that turnaround time will be more
4. It require more resources
5. More human error.
6. No accuracy in testing



• time taken to deliver the prod to customer is called turn around time



- even automation script are ready we do 2-3 rounds of regression testing manually before delivering the s/w to customer

- Baseline: comparing the result of automation testing with result of manual testing in order to test - test script

- Defect will be raised by manual team even when it is found by automation team otherwise these automation Er. will get bussy in retesting (manually) on getting the fixed build rather than doing there work (ie. writing test script)

- Defects found by executing the automation test scripts will always be less as it is ~~made~~ written after the s/w got stable.

- Some set of test cases can't be automated always like captaza, fringerprint, barcode reader (where manual intervention is required)

use of captza.

gmail ← competetor → yahoo

will create fake

akansha.abc@gmail.com
akansha.123@yahoo.com
akansha.@gmail.yahoo.com

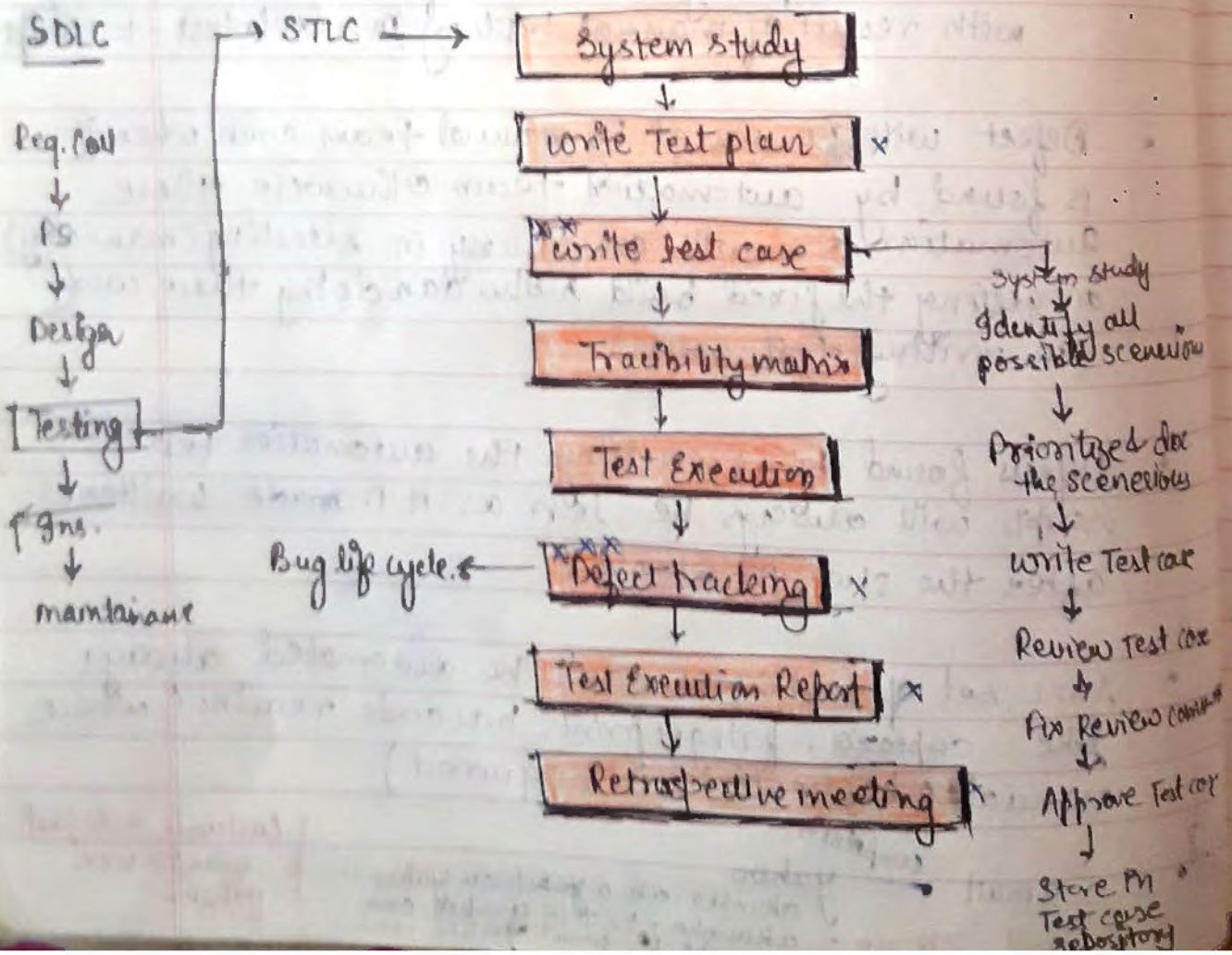
Customer will feel bore to use yahoo -!

- Automation can start in 1st release only or in 2nd release but now early automation has become possible because of agile model.

- If there is change in the requirement, automation script have also to be changed.
  - modify (if ~~not~~ written by you only)
  - write again (if earlier script is written by some one else)

STLC (s/w test life cycle

- It is the procedure to test the s/w. It has got different stages.



Q. what is the diff b/w SDLC & STLC?

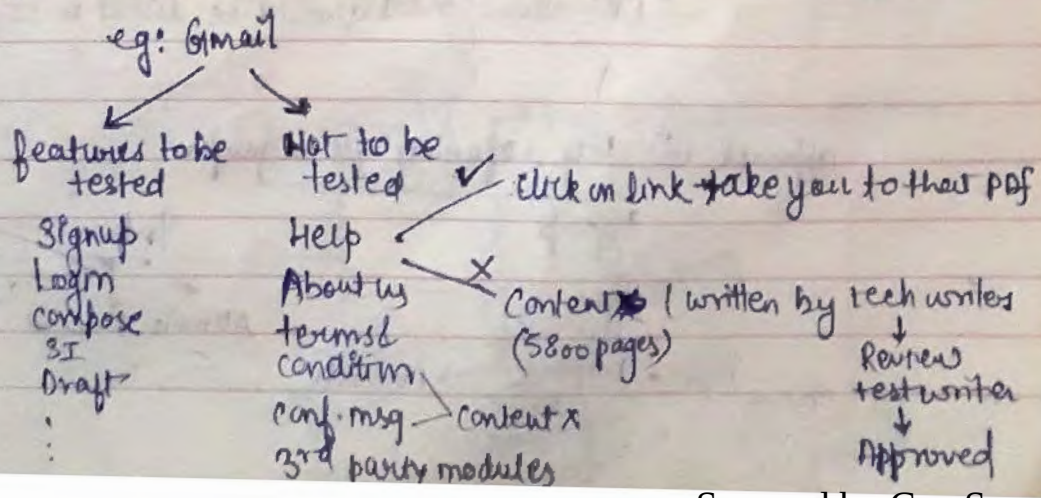
SDLC - It is the procedure to develop the s/w & it has diff stages like req. collection, feasibility study, Design, coding, testing, Installation & management.

STLC - it the procedure to test the s/w & it has diff stages like sys study, write testplan, write test case, tracibility matrix, test execution, defect tracking, test execution report & Retrospective meeting.

System study:
Here we go through the req. & understand the req. while understanding if we get any queries communicate with development team, bussiness analyst & customer and get it ~~clarify~~ clearify.

Test Plan:
It is a document which derives all the future testing activities. It has got diff sections like

1. **objective**: This section covers aim of preparing test plan.
2. **scope**: This section covers list of features to be tested (In scope). This section covers list of features not to be tested (out of scope)



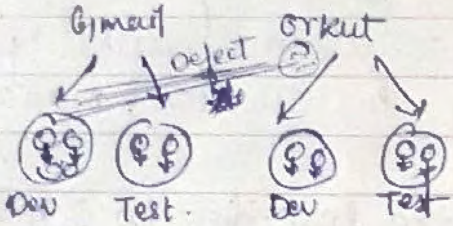
3rd party modules

eg① ABC → Appointment integrated → Op consultation / Ip cons. / pharmacy ← XYZ

TE
- will not check appointment module.
- check only whether it is integrated properly or not

eg② online shopping → payment

eg③



eg④ Application.
↳ facebook → opens → Blank page

you will raise this defect (because link is not given properly)

3. Approach: This section coveres, how we are going to test the product in future.

eg.1 to get job

ⓐ spider	ⓑ spider	ⓒ spider
manual ✗	manual ✗	manual ✓
↓	↓	↓
job	Java	Java
	↓	↓
	SQL	sql
	↓	↓
	python	selenium
		↓
		proj.

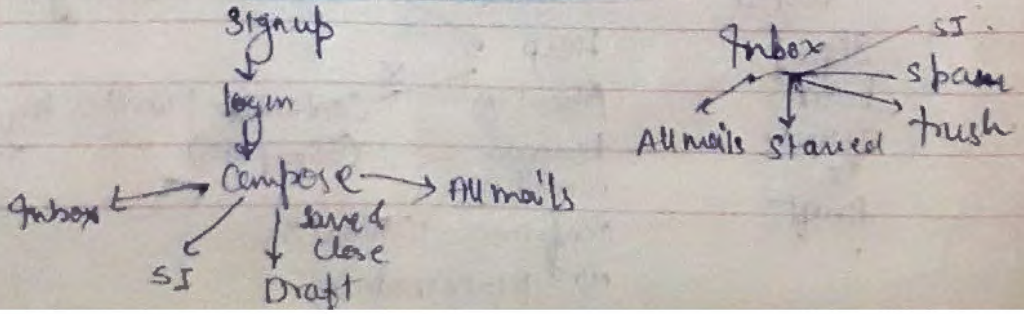
eg2

Approach ① By refering high level scenerios.

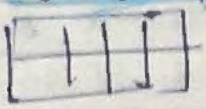
① login to gmail
② Click on Compose link
③ Enter all the data & Click on Sent button
④ Click on Inbox link.

ER: Sent mail should be listed in SI page.

approach ② By refering flow graph



approach ③. By refering detailed test case.



approach 3 is good, if long term project
approach 2 is good, if short terms project

~~at~~ it will be sent to customer & he will approve which approach to follow.

4. Test methodology:

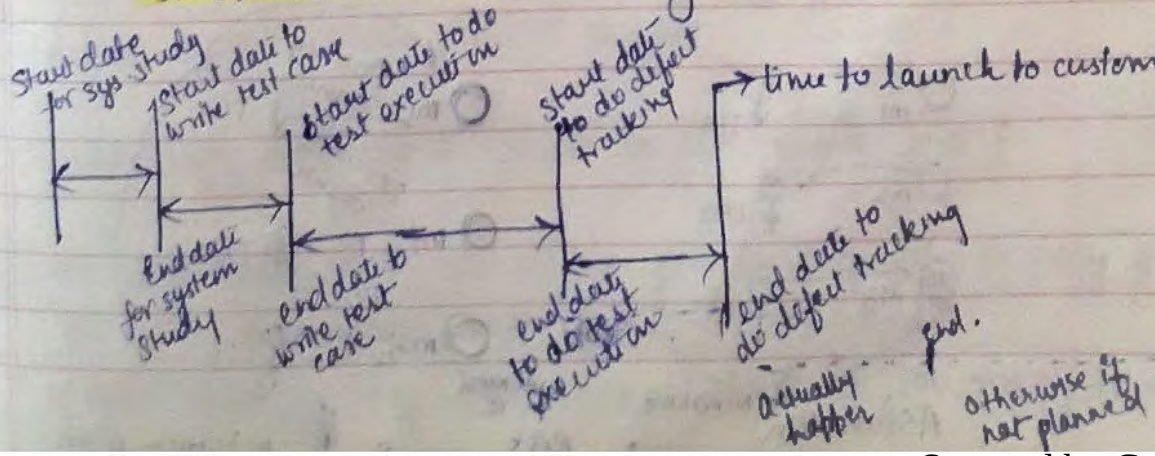
This section covers what are the types of testing we should conduct in future.

Web based application:
⇒ smoke testing, FT, IT, ST, Regression testing, UAT, performance, usability, compatibility, acceptance, Adhoc, exploratory, globalization, security, recovery.

Stand alone application:
smoke testing, FT, IT, ST, Regression testing, UAT, usability, compatibility, acceptance, Adhoc, exploratory, globalization, recovery, Installation & uninstallation.

5 Schedule:

This section coveres when exactly which activity should be started & which activity should be completed.

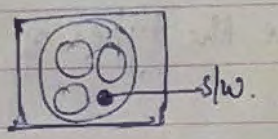


⑥ Test Environment: This section covers what are the H/w & S/w to be used in order to ~~st~~ setup test environment. This cover procedure to install the S/w.

① procedure to Install the S/w

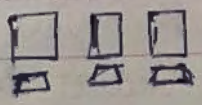
≡} yet not build.

②ⓐ Server side.



Software:
OS?
webserver?
app server?
D/B server?
H/w:
≡

ⓑ Client side

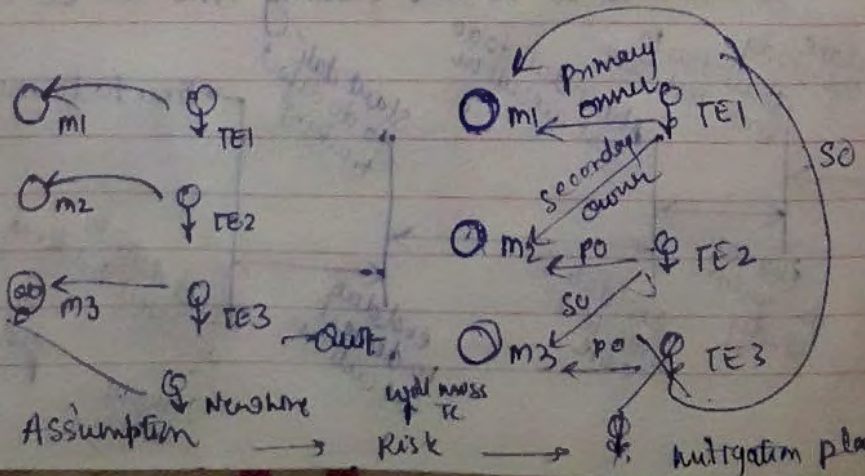


Software:
OS - W10
Browser - Mozilla, chrome, opera
MS-office
≡
Hardware:
≡ } 2GB Speed
4GB RAM.
10TB.

③. ~~Assumption~~

④ Risk

⑤ Mitigation plans / Backup plan



assumption : all the TE's will be there till the end of the proj.

Risk : If some one quit chances are there new Er. might miss lot of defects

Mitigation plan : New Er. should take help from secondary owner & test the application.

Deliverables :

This section covers the o/p of testing team to the customer by the end of the project.

(test case document, test plan, traceability matrix, test execution report, graphs & matries & release note).

Release note - new features add, modified features, defect fixes of previous release, pending defects, ~~zero~~ of the current release, removed features.

Q Why we get critical defects in the ~~end~~ nth moment of the s/w?

① Because of adhoc testing, test Er. would have got creative scenerious at the end of the project.

② Test Er. would have neglected in the begning & finding at the end of the project.

③ Because of nth moment changes done by the developer that might introduce few critical defects.

④ Test Er. There are the chances fixing the major or minor defect might create critical defect.

Q. Test stop criteria or when do we ~~stock~~ stop test?

①. We stop testing when ever the product quality is very good or when ever the product quality is very bad.

product quality very bad means there are too many blocker

& critical bugs, when end-to-end critical busstness scenario are not working

Product quality is very good means all the end to end scenerious are working fine, where there are no blocker & critical defects, there might be few bugs left but they are major or minor & it is less than the acceptable limit set by the customer.

(2) When it process the budget & if it is crossing the deadline of the schedule.

(11) Defect tracking :

This section coveres how we should track the Defects in future. It involves procedure to track the defect.

1) Procedure to track the defect
   - * severity
   - * priority

2) Severity levels.
   - * Blocker (or show stopper)
   - * critical
   - * major
   - * minor

3) Priority level
   - * urgent
   - * high
   - * medium
   - * low

4) Defect tracking tool to be used
   Bugzilla / mantis / redmin / zeal / TFS

5. Effort estimation:

This section covers total estimation of the project.

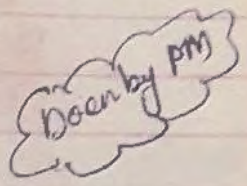
Total man hours.

→ 5000 hr

5000 × x $

= z $?

1 hr = x $



$\frac{5000}{8 hr.} = 625$ days

Customer counts in 3 months i.e 22 days × 3 = 66 days.

$= \frac{625 \text{ (total days)}}{66 \text{ (estimated days)}}$

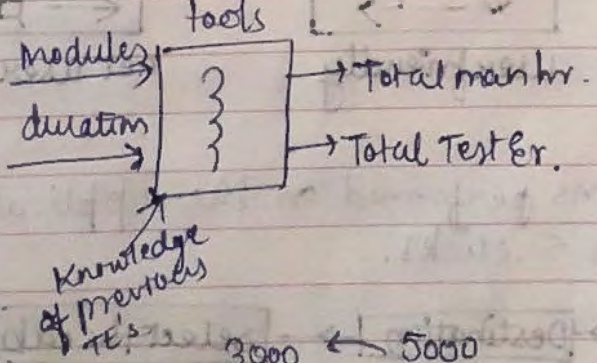
= 10 TE

How to find total man hrs?

Sl No.	module Name	Activity: System Study	Identify scenarios	Prioritised document Scenarios	writing Test case	Review test case	fix review comm	Traceability matrix	test exe	Defect tracking	Total hr
01	compose	2hrs	-	-	-	-	-	-	-	-	-
02	Inbox	-	-	-	-	-	-	-	-	-	-

Total 5000

estimated.



3000 ← 5000

+ 30%

6500

9000

Usability testing

- non-functional testing.

Defination:

Testing the user friendliness of an application is called as usability testing

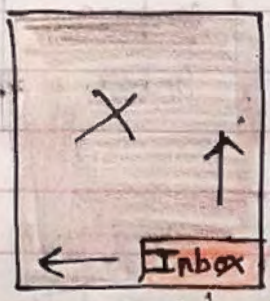
(OR)

The effort spend less on an application to use it is called as user friendly application. Testing this is called as usability testing.

On what bases usability testing is doen?

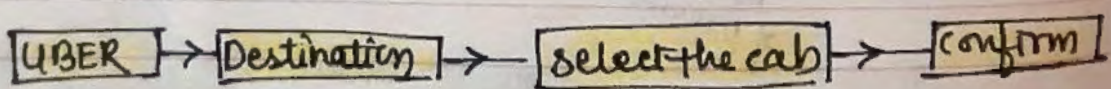
① All the frequently used features has to be from top to bottom & from left to right. being a test Er. we list out all the frequently used features of an application & check if it is from Top to bottom & left to right.

user friendly

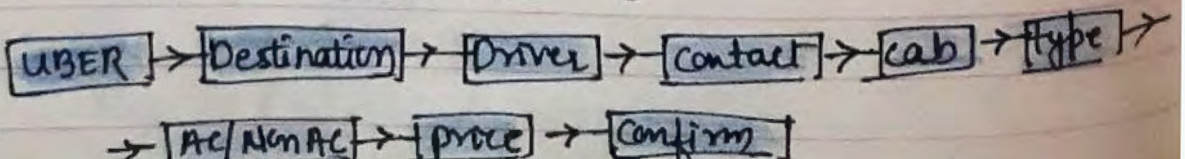


not user friendly

② No. of actions performed on the application has to lesser than 5 clicks.



user friendly



not user friendly

③ Easy to understand:
The app has to be simple & it can be used by any end users (frequent users & new users)

④ Look & feel of the application:
The application used by the end user is always judged based on the look & feel of the application (pleasent to your eye)

For what applications we do usability testing?

① If the application is expected to be very simple & can be used by any end users, then on such applications we do usability testing.

② If the application doesnot require any training than we do usability testing on such application.

③ If the application generates a lot of revenue than we make sure we do usability testing (e.g. e-commerce application, banking app.)

④ If the customer is keen on userfriendly of the app. than we do usability testing of the application

Who does usability testing?

It is a good approach if the end-users does usability testing on the app rather than TE's doing it, because by the time TE's performs various testings on the application the app. becomes userfriendly to them.

How to do usability testing ?

① Identify end users who frequently use the application & who is the new user.

② Provide the check list to both the end users

eg. Flight booking application

Activities	simple to understand	look & feel	No. of actions
Search			
Book			
Status			
Cancel			

5 - Very go
4 - good
3 - Averag
2 - Bad
1 - poor

Checklist

③ Ask the end users to rate the application / modules based on simple to understands, look & fell & no. of actions

④ collect the checklist from the end users & finally consolidate the features which has got the rating 3 & lesser than 3.

⑤ Sent it to the development team raising it as a bug & asking them to make it more userfriendly.

When to do usability testing:

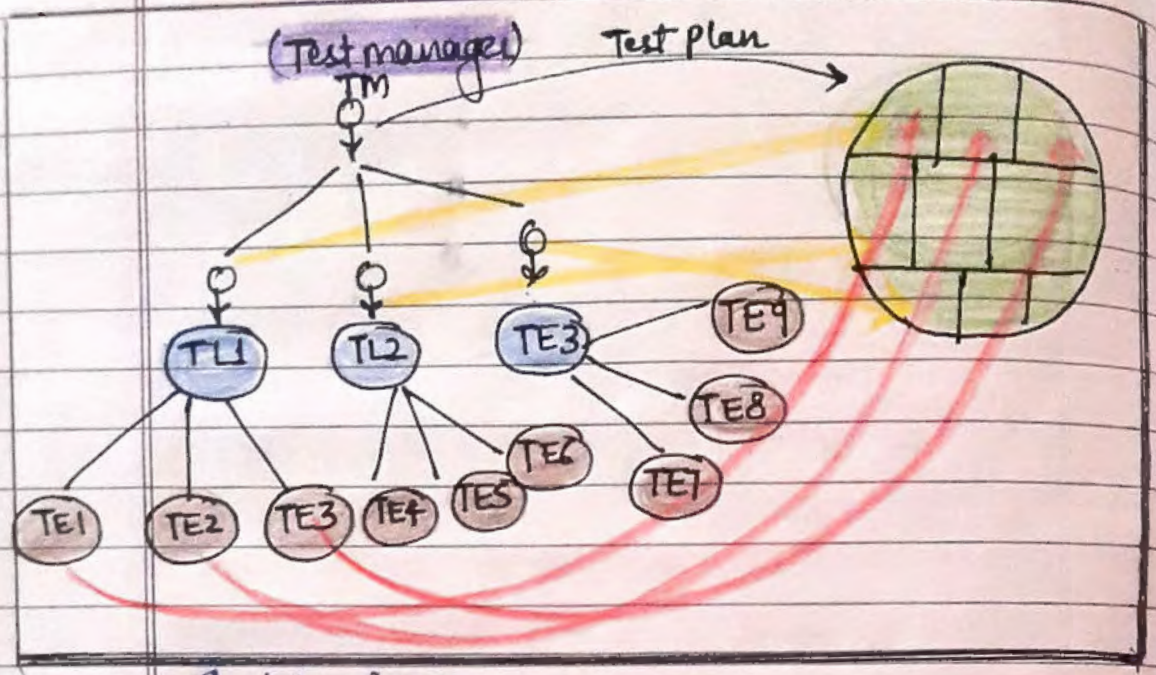
① When the customer is very keen on the userfriendliness of the app. we will do it from the initial build onwards.

(2) If the customer is not very keen on the userfriendliness of the app we will do at the end when all the possible testing has been done.

Assignment

Prepare a checklist for makemytrip.com, Yatra.com & cleartrip.com on activities search, book, status & cancel.

13. Roles & Responsibility



In interview :-
Test plane ~~was~~ is written by
60% → TM
40% → TL

But actually :-
60% → TL
20% → Senior TE
20% → TM

Test manager ① define test life cycle & ensure that people are following it

② He prepare test plan.

③ Interact with developer, customer & management

④ Handle issues & ast asclation.

⑤ Effort estimation.

Test lead:

1. Assign the work to test Er & ensure that they complete the task within the schedule.
2. Consolidate the report sent by test Er & sent it to customer, developers & management.
3. Write / review test plan.

Test Er:

1. Review test plan.
2. Write test case
3. Review test case
4. Execute test case
5. Perform diff types of testing on the product
6. Involved in the test execution, defect tracking
7. Involved in preparing test execution report.

Automation test Er:

1. Setup the test Env. & Install the product
2. Identify the test cases to be automated.
3. Automate all the regression test cases

④ Execute & maintain automation test script.

14. **Templates:**

This section covers templates for all the documents which we are planning to prepare in future

① Test case format/~~template~~ template :
② Traceibility Matrix
③ Test execution report
④ Defect report
⑤ Graphs & matrixs.

Q Who write test plan?

Test lead or test manager

Q. who review test plan?

Test lead or test manager, development team even customer can also review test plans

Q. who approve the test plan?

Test manager or customer approve the test plan.

15. **Test automation:**

① This section covers which are all the test cases can be automated.

② which are all the test cases can't be automated

③ which automation tool to be used.

Traceability Matrix:

It is a document which ensure that every requirement has got atleast 1 test case.

Sl. No.	Module Name	High level req.	detailed req.	Test case Name	Automation Script Name
01	Loans	HL	R1.1	TC1	TS1
			R1.2	TC2	TS2
		PL	R2.1	TC3	TS3
			R2.2	TC4	TS4
		ML			

Traceibility Matrix

Advantages of traceibility matrix:

① It gives traceibility from high level requirement till automation test script

② If suddenly requirement change, we will be knowing which exact test cases & automation test script to be modified

③ we can ensure that every req. has got atleast 1 test case.

Drawbacks of traceability:

(1). It will not ensure we get 100% coverage.

eg. 1 Req → should have 15 scenerios → but u have written 10

Q What is the diff. b/w traceability matrix & test case review?

In traceability matrix we check whether every req. has got atleast 1 test case

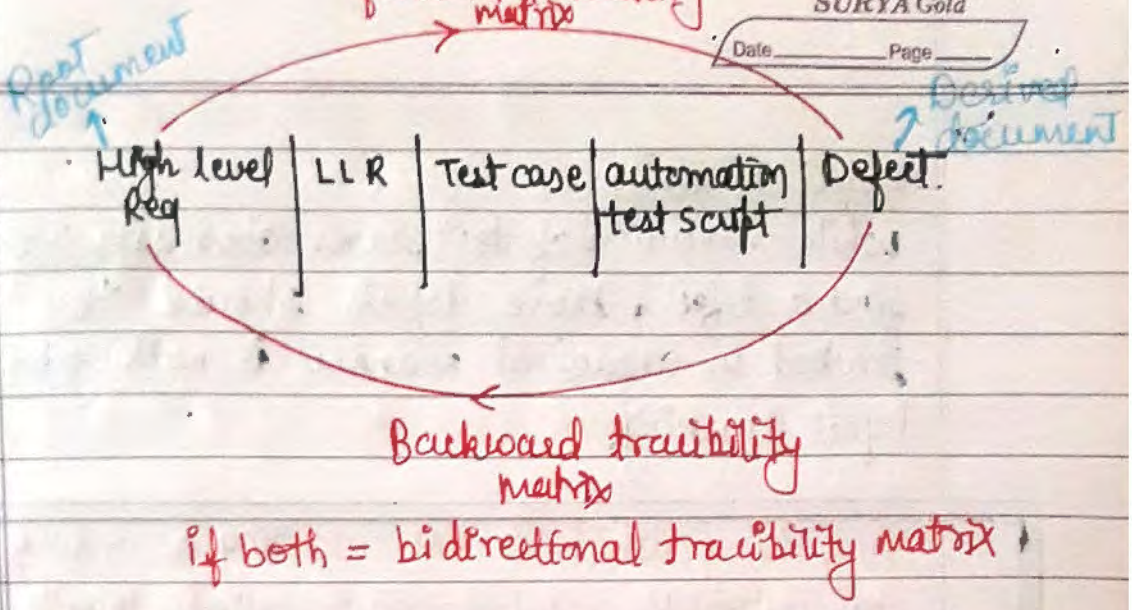
In test case review we check whether all possible scenerios are covered for the specific requirement

In traceability matrix we don't check test case is having all possible scenerios

In test case review we don't check whether all the scenerios ~~are~~ for a testcase is covered or not.

Q. How do you do traceability matrix without having requirement document?

I will go through application & I will list all the feature & I will check all the feature has got atleast 1 testcase or not.



if both = bi directional traceability matrix.

mapping the requirement from root document to derived document it is called forward traceability matrix.

mapping the requirement from Derived document to root document is called backward / reverse traceability matrix.

And if its functioning in both the direction then we call it as bi-directional traceability matrix.

Test Execution:

This is the phase where in we test the s/w more than 45-50 test cycle.

This is a phase where the tester are productive to the organisation & this is where we find the defect.

And this is the stage where we ensure that quality of product is good or not.

Defect tracking:
while execution of test cases obviously we find a defect, those defects should be tracked in organised manner is nothing but defect tracking.

Anything which is not working according to customers requirement is called defect

we encounter defects because of the following reasons:

1. Wrong implementation
2. Missing implementation
3. Extra implementation.

Wrong implementation:

eg1. Sent mail → Inbox
→ Trash

eg2. FB — friend req sent → accepting automatically

eg3. A 10,000 — 500 → B (Amount transfer from A to B but happing from B to A)

Missing requirement:

eg1. Gmail Inbox — select all — not there

Extra requirement:
If developer develop anything which is not ask by customer.

Q. What is the diff b/w defect, bug, error, failure?

Deviation from the req specification is called defect

Bug, It is the informal name given to defect

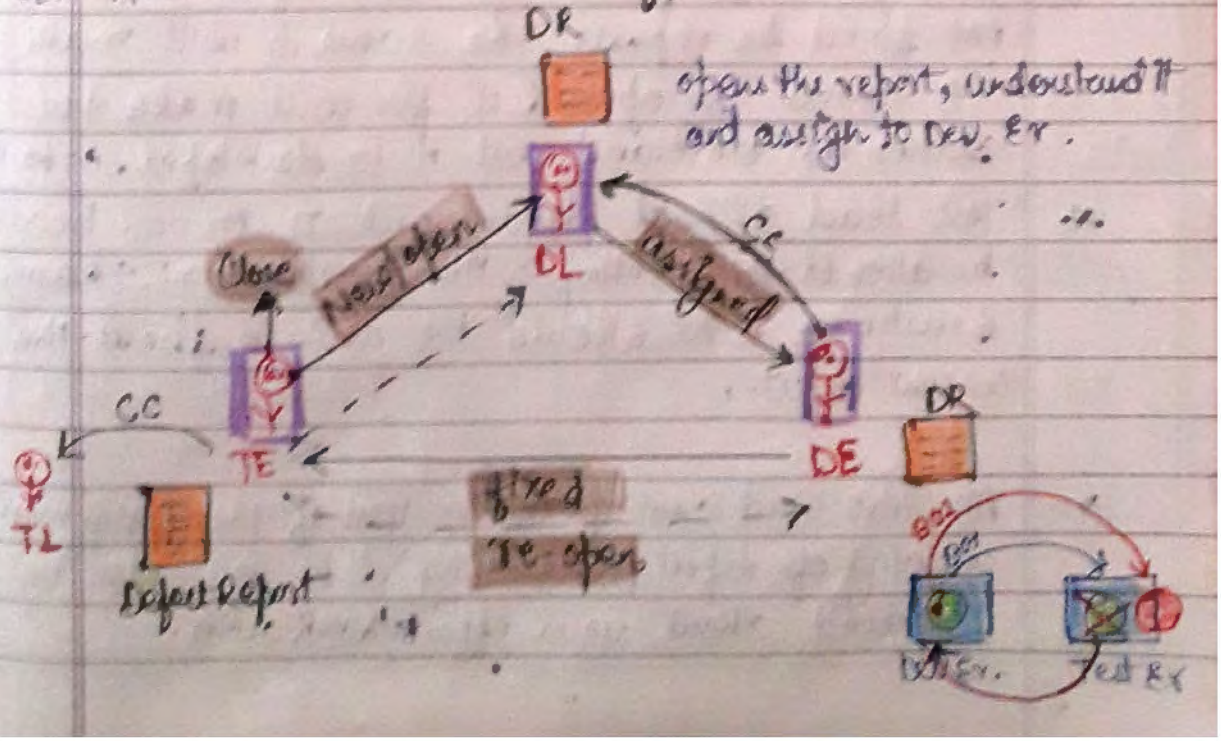
Error, It is the mistake done in the program because of that we are not able to compile or run the program.

There are 2 types of errors, one is compile type error and run type error. Compile type error is because of syntactical mistake, run type error is because of logical error.

Defect causes/leads to failure.

Error → Defect / Bug → failure
(prog) (given by TE) (given by customer).

Q. How do we track the defect?



* If the TE finds the defect, he will raise it by making a defect report & will sent it to Devels Development lead (if don't know the respective developer) or to Developer (directly if he knows who have developed that feature) keeping in Sending cc to developer.

* While sending the defect report he will make the status as New/open.

* Now developer will open the mail & understand the difect, if its a really a defect he will sent it to respected developer to fix it & make the status assigned

* Now developer will open the mail & will cross check it, if he find it as defect he will fix it & will create a new build & inform & sent to respective TE as well as put DL in cc & will make the status fix.

* Now if tester finds that the defect is still not fixed he repeats the same & will make the status re-open. & if fix will make the status as close and sent it to developer. cc to DL

* Test lead should always put TL in cc. because he attends the meeting with development team, PM & customer so he should be aware about the current defects.

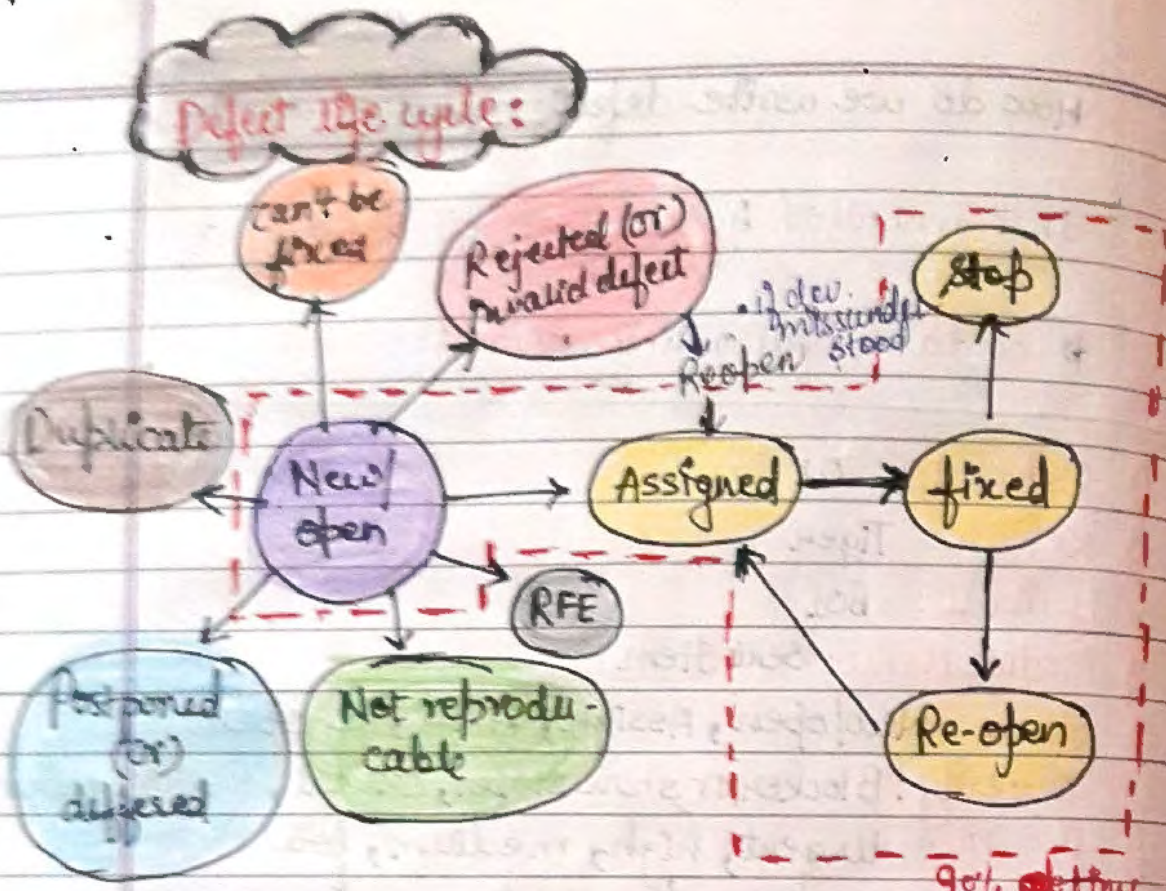
* Also test lead can speak for you if the assignment or fixing of defect takes long. & he will also be aware about your efforts/work done.

How do we write defect report?

It is generated by using defect tracking tool.

It contains following attribute:

1) Defect ID: 01
2) Release: Tiger
3) Build ID: B01
4) Module Name: Sent Item
5) Status: New/open, Assign, fixed, Re-open
6) Severity: Blocker or show stoper, critical, major, minor
7) Priority: Urgent, high, medium, low
   - Urgent → P1
   - high → P2
   - medium → P3
   - low → P4
8) Test Environment: OS → w10
   Browser → Mozilla
   Chrome
   IE
   Browser version →
9) Test data: Username: Ding, P/w: Ding1
10) Test case Name: Gmail-compose. All possible scenerios
11) Brief Description: Sent mail is not displayed in sent item page.
12) Detailed Description:
    1) login to gmail with valid login credential
    2) click on compose link
    3) Enter all the field & click on sent button
    4) click on SI link
13) Expected Result: Sent mail should be listed in SI page.
14) Actual result: Sent mail is not listed in SI page
15) Attachment: [ ] Browse.



Q. What is Reject status?

Test Er. calls it as a bug & sent it to development team. development lead reject the defect & say it is a feature.

Q. Why we get reject status?

① Becoz of tester misunderstanding the requirement.

eg. amazon

admin → login → added 5 product

customer → login → showing 50 products

searching for college bag it will show all sort of bags.

Developer write such a logic to make impression to customer but it might understand it as defect.

eg. In gmail click on save & candle, mail will be saved to draft. It might understand it as defect.

② Test Er. might refer to old requirement.
eg whatsapp $\frac{old}{req}$ group - only 1 admin
$\frac{new}{req}$ group - can have multiple admin

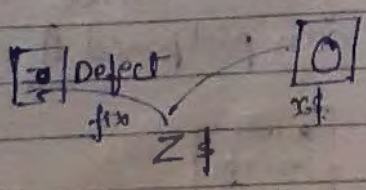
③ Because of improper installation of the product

④ Because of extra feature.
(anything which is not asked by customer but added to the s/w it will be considered as defect only);-if

⑤ **Can't be fixed:**
① Developer, they are accepting it as a defect but they are not in the position to fix it the defect because of some reasons.

① When ever technology is not supporting Dev team will say can't be fixed.

If the defect is critical they will have to change the technology & fix it. for minor & major they r not going to change the technology.



② when ever the cost of fixing the defect is more than the cost of defect developer might not fix it

① new it has come } Reason for duplicate status
② common features }

③ If there is minor defect in the root of the product developer says can't be fixed.

Duplicate:

you sent a bug to development head but the same bug is already sent by another TL so the DL change the status as duplicate.

Search the defect in defect repository before sending it to the development team.

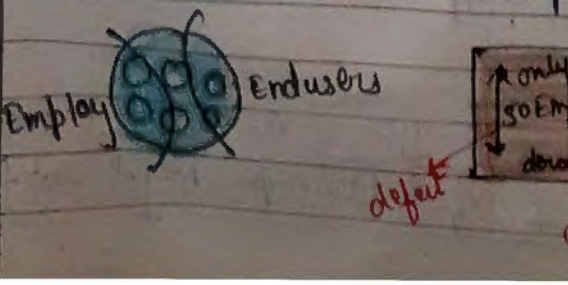
Postponed

Development team, they are accepting it as defect but they are not in the position to fix it so they change the status to postponed.

① If you find a minor defect & sent it to development team & Dev team is busy in fixing the critical & blocker defect, since they have insufficient time so they postpond the minor defect.
eg. spelling mistake

② you found a bug in a feature when development team expecting a lot of changes from customers side, they make status to postponed

③ If there is a bug in a feature which is exposed to internal user, we can postponed it.



Not Reproducable:

If you find a defect & send to developer he is not getting that defect.

(1) Because of Improper defect report -
Developer is not able to reproduce the defect you have found, by reading your defect report.

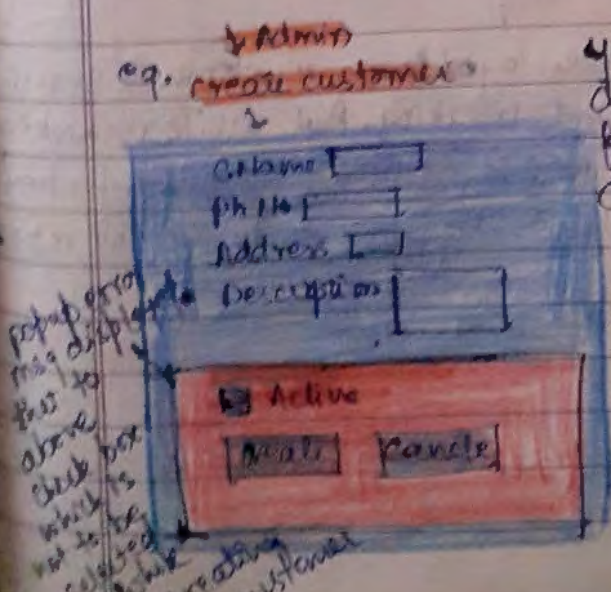
eg. 

You sent a defect to developer that clicking on to login than click on compose & sent a mail, blank page is displayed

↓

Developer opens up the report & try to do the same step but he coundn't find the defect you have found. Because you have not mentioned that on sending multiple mails we get the blank page.

eg. Same above eg. for multiple attachments not getting send.

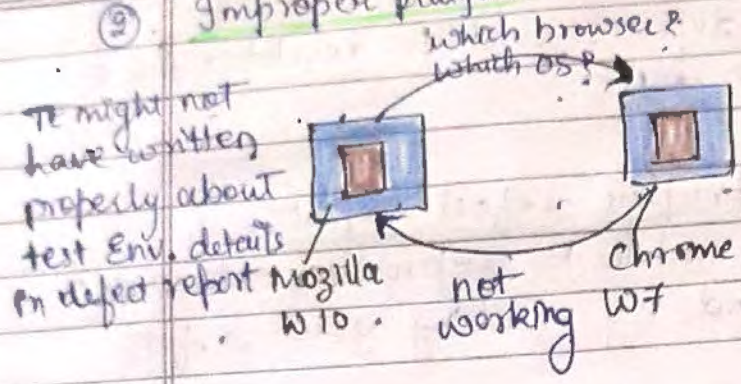
eg. 

You have sent the defect to developer that login as Admin fill all the entry & click on create an error msg display

↓

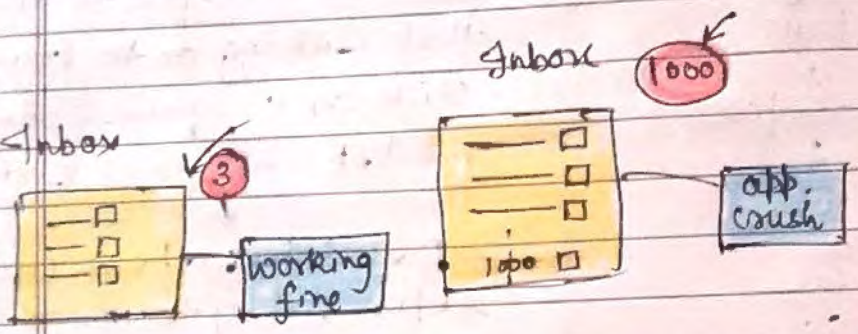
Dev. opens the report & tried it but he is not getting because defect was due to you have select the check box which is not to be selected & error msg was again pop up msg which was displayed above check box & screen shot also didn't helped him.

② Improper platform:



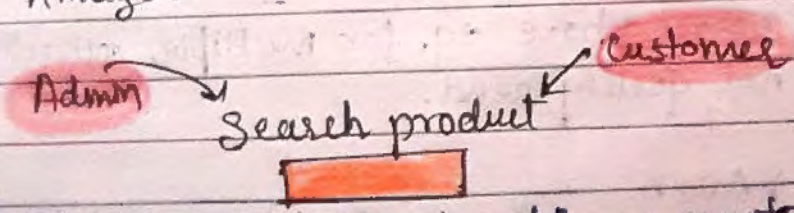
③ Because of Incorrect data.

eg1.



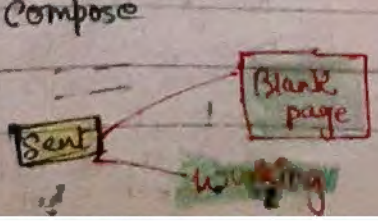
TE ~~was~~ will sent the defect to dev. that Inbox is crushing. But ~~for~~ dev it's working fine. Here TE might ~~no~~ have ~~trea~~ clarified that for 1000 mails its not ~~co~~ working

eg2. Amazon.



Here TE might have loged in as customer & searc option for him is not working. But while rechecki developer ~~might~~ has loged in as admin & for h ~~is~~ search option is working fine so he couldn't rep the defect.

④ Because of Inconsistent defect.



SURYA Gold
Date Page

Request for enhancement (RFE)

eg. 

Req: must select & delete 1 at a time.
TE: select all option is not not there

eg. whatsapp
Adm can only add & delete member

facebook group
↓
moderate feature (Who can post, ~~add &~~ accept req & send req). (will create 2 admin)
TE: ~~Add~~ Moderate feature is not there

Defn: Test Er asking for any changes in the features it is called as Request for enhancement. Now PM will ask to customer, if he says yes than it will be added to that module by the developer.

Defect seeding:
- Intentionally adding the defect in the S/w to check the efficiency of test Er. Is called Defect seeding.

Defect masking:
- If 1 defect is masking / hiding the other defect it is called defect masking.

eg. → apply OD (defect) → Approve OD (defect)

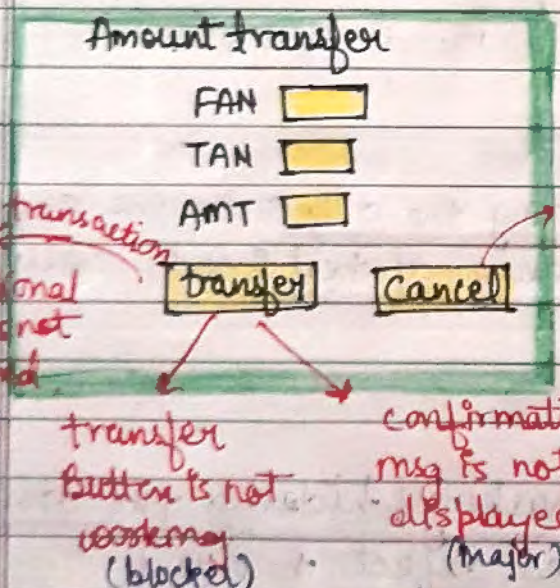
there is defect in ~~apply~~ approve ~~approve~~ OD thus thus which can't be checked becoz there is defect in apply OD only.

eg. Sent mail button is not ~~sending~~ sending the mail then how can I check whether ~~sent item~~ that mail displayed in sent item list or not. If there is defect in sent item list.

Severity & priority :

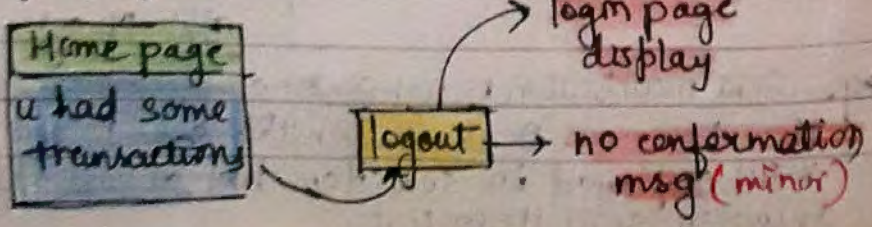
- Severity is decided based on the impact of defect on ~~the~~ customer bussiness.
- The severity levels are blocker or showstopper, critical, major & minor.
- Which defect has to be fixed first is decided by priority. or the order in which ~~developer~~ defect is fixed is decided by priority.
- Priority levels are urgent, high, medium, low.

eg.



If you find a defect, try to analyse the bussiness prospective.

eg. login gmail.



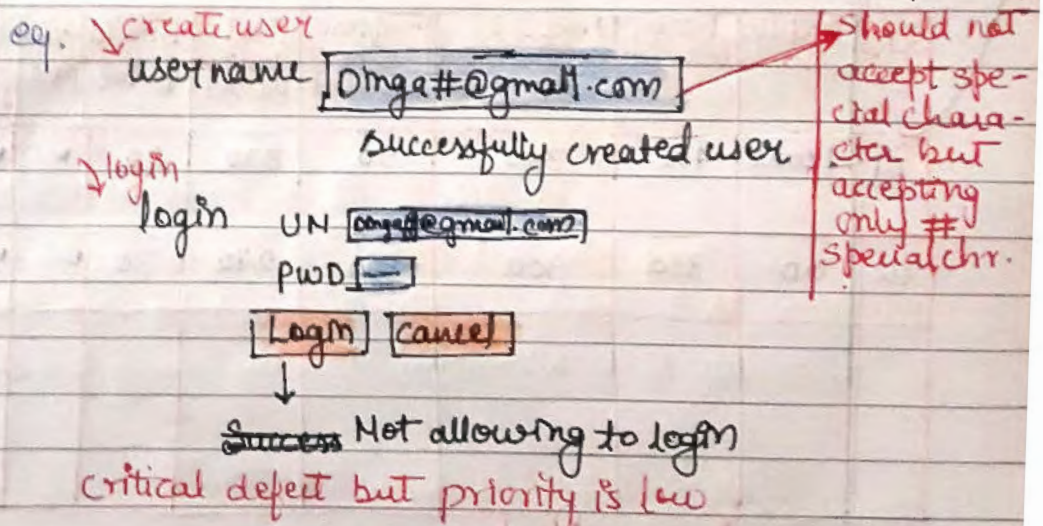
SURYA Gold
Date _____ Page _____

95% of times
Blocker defects will have urgent priority
critical will have High
Major will have medium
minor will have low

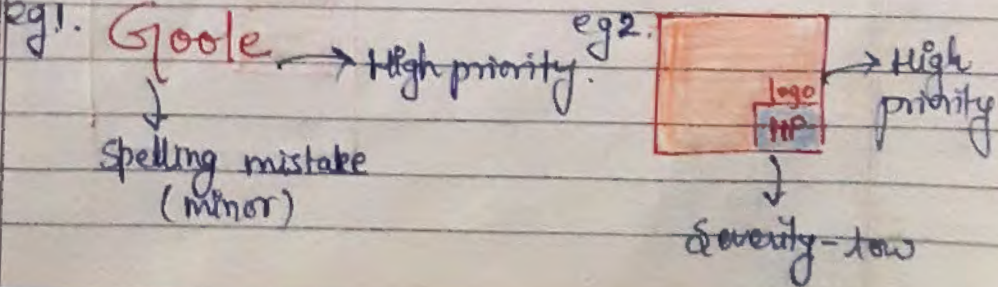
Q. Give an example for high severiousity & low priority

eg. click on help link & if it take us to blank page, it is a critical defect but priority is low.

eg. clicking on to calander link & it is taking to bank page, it is a critical defect but priority is low.



Q Give an example low severity (minor/major) & high priority?



company logo & company name even when the severity is low but should be fixed first mean high priority.

Test Execution Report:

loans

SLNo	Test case Name	Status
01	Test case-Hi	pass
02	Test case-approve loan	fail
:		
350		

loans_approved

StepNo	Action	I/P	ER	AR	status

which step is failed you will kno

SLNo	Module Name	Total test case	No of test case executed	No of test case not executed	Total Test case passed	Total TC failed	% pass	% fail
01	Loans	400	350	50	330	20		
02	OD	350	300	50	270	30		
Summary	loans	OD						

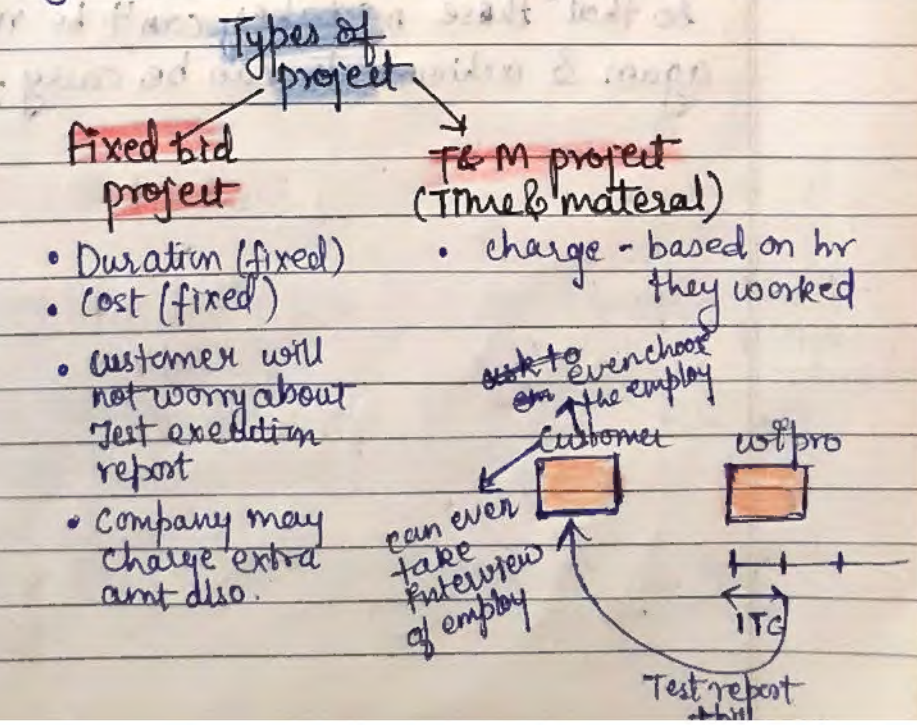
BO1
BO2
BO3

At the end of every test cycle we prepare test execution report. It is a document which has got information about:

1. Total Test case
2. Total no. of Test case executed
3. Total no. of Test Case pass
4. Total no. of Test cases failed
5. % of pass
6. % of fail

we send this doc. to dev team, bussiness analyst & management. at the end of last test cycle we send the repot to customer that is the end of test life cycle from the customer point of view.

From the testing point of view we have 1 more activity ie. retrospective meeting or also called as proj. closer meeting or postmartam meeting.



In case of Time & material proj, customer can ask for a test execution report at the end of each test cycle.

Retrospective meeting:

Conducted by PM

Discuss → achievment ?
→ Mistake ?

Mistake
- we ~~mig~~ have not reviewed the test case.
- we have done lot of full regression testing ~~not~~ instead of it. we would have done regional regression testing

Achievments
- It is good that we have ~~followed~~ done traceibility matrix.
- At the end of every Test cycle we ~~should~~ have done Adhoc testing

We document this & it is beneficial while writting the test case for next release/next so that these mistakes can't be repeated again & achievments can be carry forward

SURYA Gold
Date Page

Adhoc Testing

also called as monkey/ gorilla testing

Defination:

Testing the application randomly without following ~~any~~ refering any formal documents like requirement, test cases & test scenerios is called adhoc testing.

Why we do adhoc testing?

1. When the product is launched to the production, chances are there end user might use the app. randomly & they might find some defect to avoid that test er. only should test the app. randomly & find the defect.

2. If we follow the req & test the application the no. of defect we gona catch is very less so we should come up with creative scenerios which is out of requirement & test the app. so that we can find more defects.

3. To some how ~~brea~~ break the product we do adhoc testing.

4. To some how ↑ the bug count we do adhoc testing

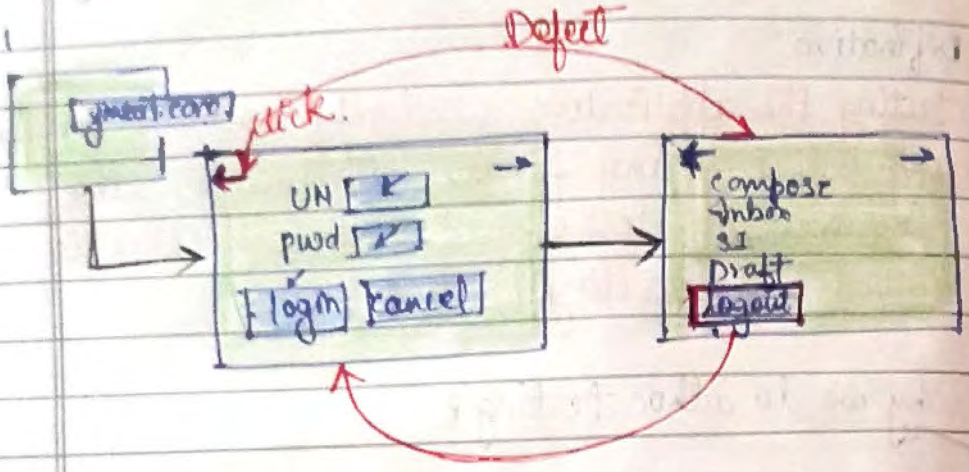
5. To check whether the s/w is working according to the implicit req. we do adhoc testing.

6. To some how ↑ the test coverage we do adhoc testing.

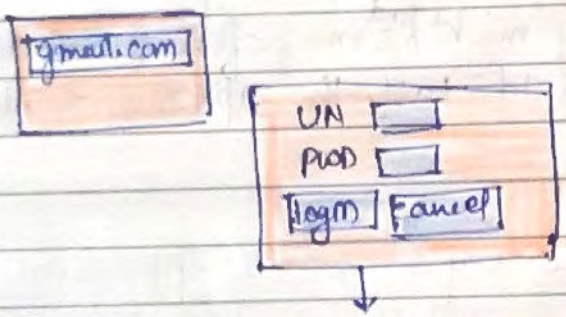
Explicit req : given by customer
Implicit req : by common sense (eg. customer give req for logm: it's should have logout this is common sense)

How to do adhoc testing?

eg1



eg2. Mozilla



Home page
↓
copy url of home page
↓
logout
↓
paste home page url.
↓
Should not take you to home page, logm page will be displayed

eg3. Mozilla
↓
logIn
↓
Home page
↓
copy url

Chrome
paste
↓
logm page will be displayed, not

eg4. at same time

Mozilla	Chrome
↓	↓
login	login
↓	↓
Home page	Home page
↓	↓
compose	changed pwd
↓	
Error msg: Session Expire	

eg5. Mozilla + New tab

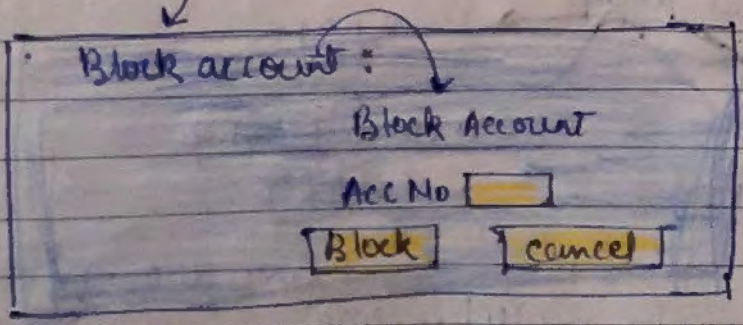
Mozilla	New tab
login	gmail.com
↓	↓
Home page	Home page.

eg6. FMS (froad management sys)

↓ manager

ICICI

↙ End user ↘ manager



- manger have blocked, & some body is transfering money to that blocked account it should not get transfered.
- loans feature might be processing for this blocked user.
- If this user is already login & Manager blocked it from that point of tym only it should not work rather than - blocked user login

eg. appointment → Consultent → pharmacy → Data to

Is in stock

forgot to click on submit

again clicked on submit later on 1st sys. Stock 5-10 = -5

Defect

patient → pharmacy → Data to → Submit

clicked on submit selecting 10 ⇒ Stock 15 / 10 / -5

eg. Us, Pw, click Login → Home page; multiple click → Not displaying Home page

• now they are not even allowing for multiple click.

eg. create user

User, Pwd

Submit, Cancel

Right click → copy, cut, undo → Blank

It's a defect but user able or not it depend

eg. password

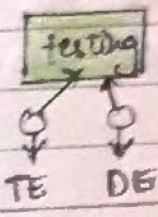
Smoke testing: +ive testing
adhoc:- -ive testing

Q when we do adhoc testing?

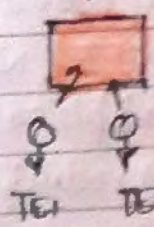
A when the application is relatively stable than we think of doing adhoc testing.

Note:
while doing smoke testing we should not do adhoc testing becoz smoke testing is +ive testing & adhoc testing is -ve testing

pair testing



Buddy testing

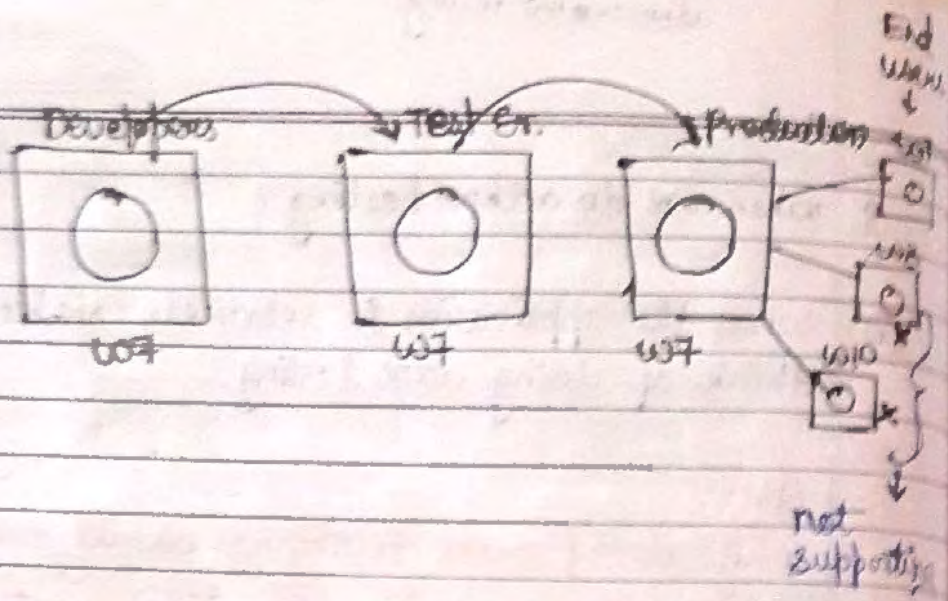


while doing pair testing & Buddy testing we can do adhoc testing.

Compatability Testing:

- non functional testing

Defination:
Testing the application in diff s/w & h/w platform is called as compatibility testing.

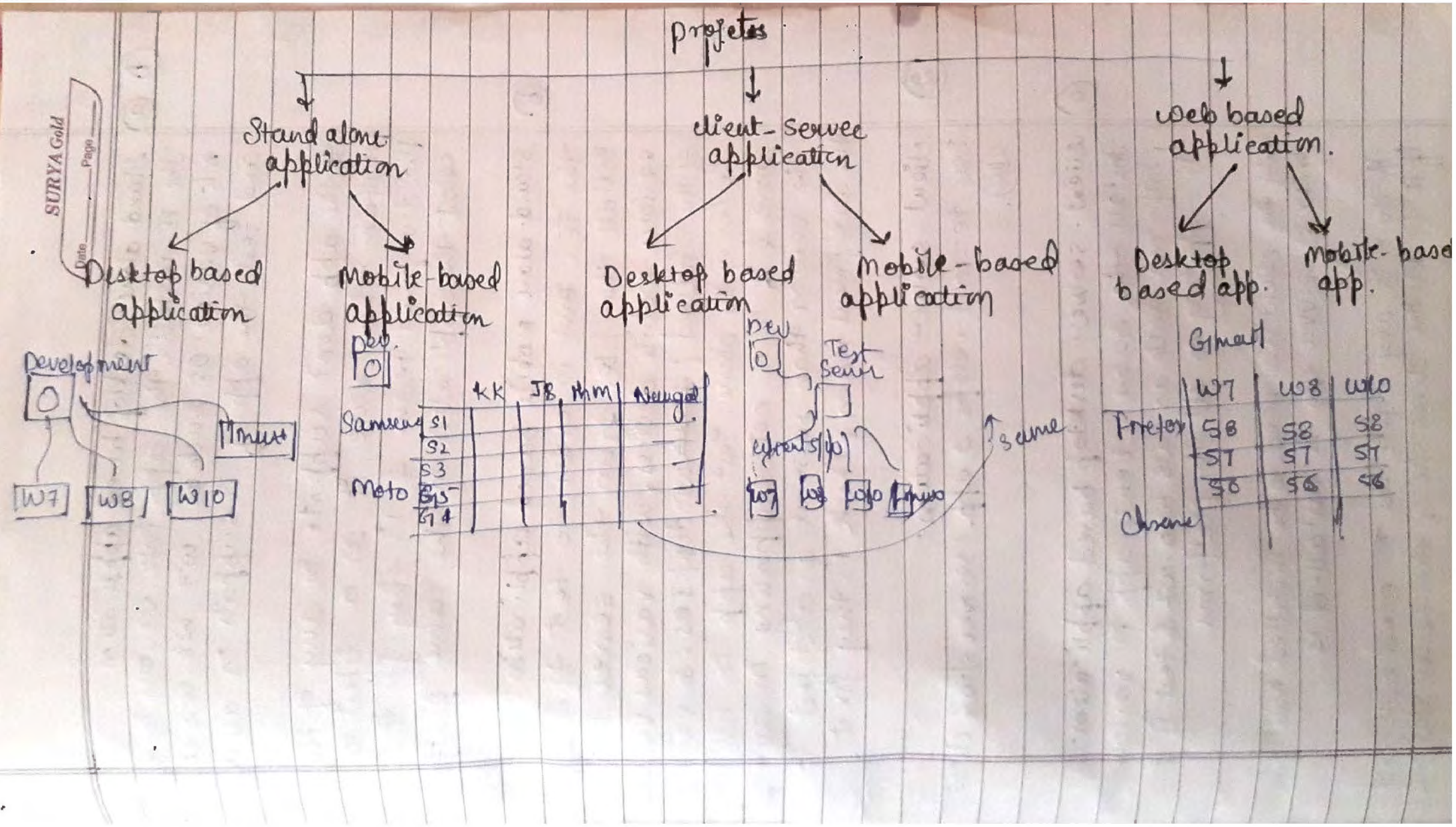


If the app is not tested in diff platforms it will not work for all the OS used by the end users. Hence TE make sure they test the application in various platforms & if it is working fine then only they will launch it to end users.

Why to do compatibility testing?

① To make sure developers develop the application in diff platforms for eg windows 7, v38, v10 etc & if the TE is testing the application only in one platform & if the product is launched to end users then some of the features may not support in all the OS. So in order to avoid this the application is tested in all the possible platforms or multiple platforms if not the customers of the app will go down.

② Compatibility testing is done in order to provide consistency to the app in all the platforms.



(1) (a) Stand-alone desktop based application:
The TE install's the app into various local m/c of various OS such as (W7, W8, W10 etc) and test the app if it supports in all the OS.

If the app. don't supports in any of the platform then raise it as a defect to the development team & it's they fix it retest the application in the same platform

(b) Stand-alone mobile based application:
The TE's ~~first~~ in order to test the app. Install the file from the server into various mobile devices, with various versions of the android / ~~iphone~~ windows, IOs, & various series of the phone. Test the app. in all the above & if it is not supporting in any of the version than raise it as a defect to the development team & check it's they fix it.

(2) client-server application:
Dev. team develops 2 app. (server s/w & client s/w).

(a) client-server desktop based application:
Install the same client app. in various platforms such as W8, W10, W7 & test if it is supporting in all the platforms.

only the client s/w need to be installed multiple times & server s/w is installed is.

If there is any defect the TE need not find if it is in the client s/w or the server s/w.

Mobile-based client-server application:

The build given by the developer will be containing s/w's of which server s/w will be installed in testing server & client s/w will be installed in diff mobile devices by the TE's & then they perform testing of that application. If they find any defect they will raise it to developer & this cycle will continue uptill s/w become stable ones s/w will be stable. It will be given to customer, & installed in production server & client s/w will be installed in some portal like appstore, playstore goods play from where the customer users can download & install. & tester in order to test the app. Install the client s/w file into various mobile devices with various versions of android / windows / IOS & various series of the phone. Test the app in all the above & if it is not supporting in any of the version then raise it as a defect to development team.

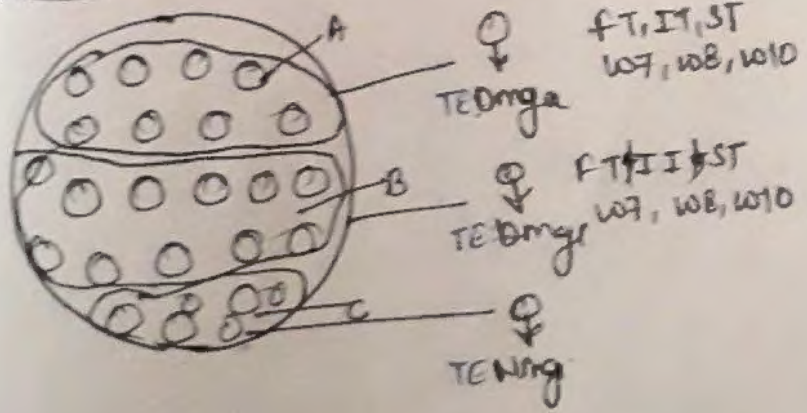
Web based application:

- Desktop based — Install the same s/w on various platforms such as WS, W10, & test if it is supporting in all the versions.
- (b) Mobile Based. — Install the s/w into various mob devices, with various versions of android / IOS & various series of the phone & test the app

Note:- Compatibility testing is not a seperate testing It is performing FT / IT / ST in diff platforms.

How to allocate work in compatibility testing?

1st Approach

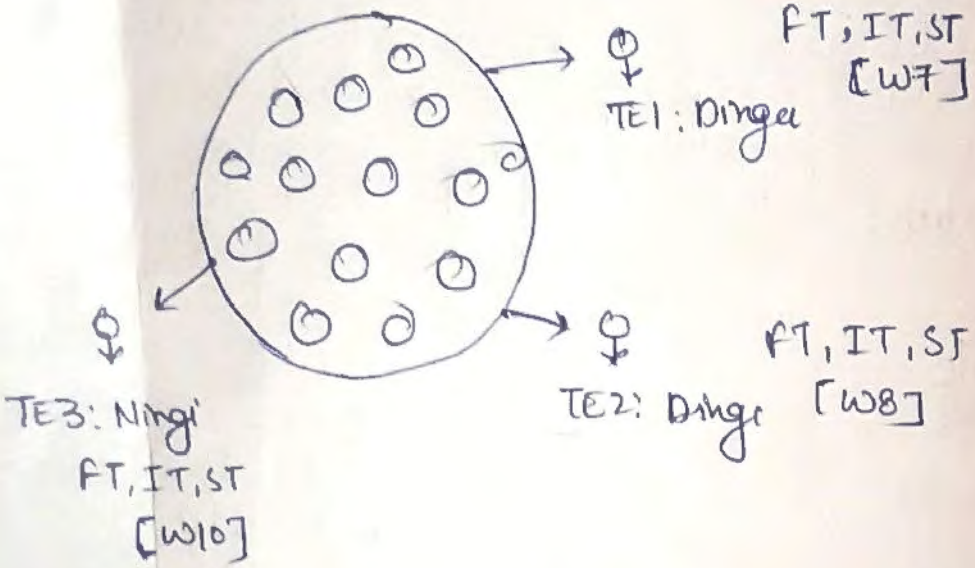


Compatibility testing → performing FT, IT, ST in diff platforms.

group & segregate all the related modules & features & assign it to TE1, TE2 & TE3 & ask each TE to perform FT, ST, IT on different platforms.

eg. TE1 can di assign all the modules of A & he performs FT/IT/ST on windows 7, 8 & 10.

2nd Approach

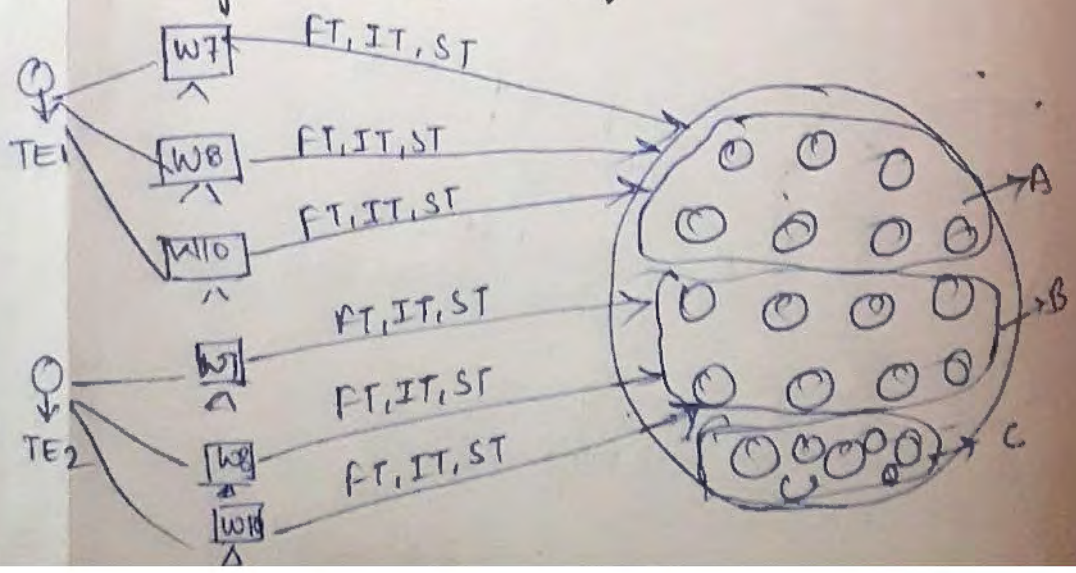


All the modules of the application is assigned to 1 test where he performs FT, IT, ST in only 1 platform. Similar another TE performs FT, IT, ST on all the modules in one platform.

This is not a good approach & is normally not follo organisations.

How to setup an Env. for CT?

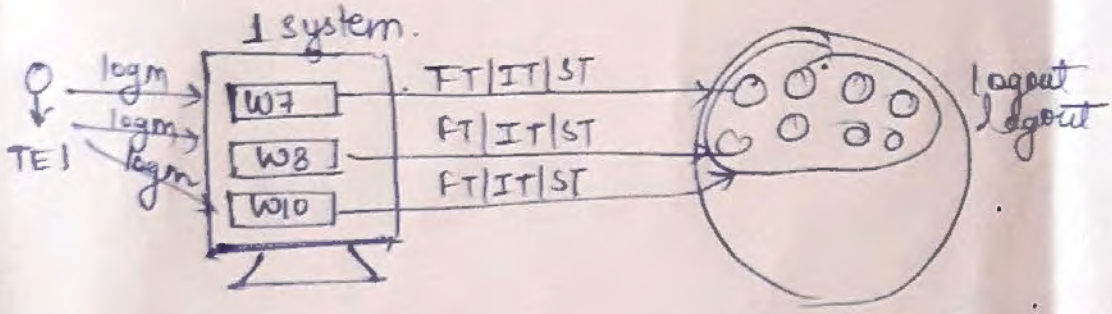
(1)



Here each test er will be alloted three diff m/c ~~with~~ installed with 3 diff os & then they will test the module alloted to him.

If the app is not running well any of the s/w he will raise it as a defect that this app. is not supporting that particular os. But ~~for~~ testing way we need more no. of systems

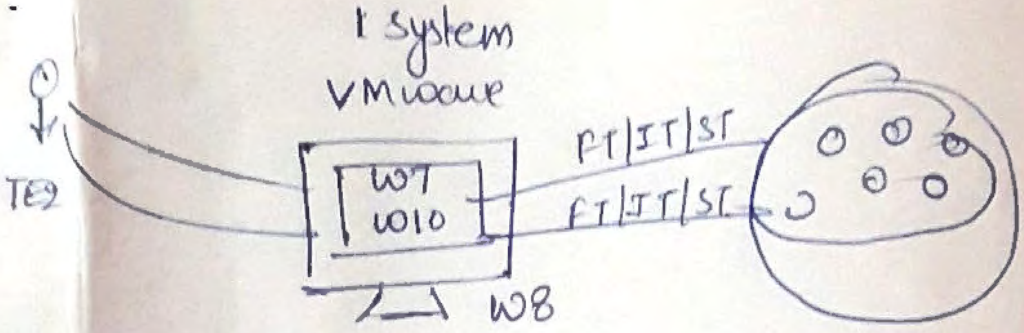
2).



Here TE1 will be given only a single sys installed with all the 3 diff os in it & he will have to logm ~~with~~ for eg. W7 & check the app. & logout & similarly check for another os also.

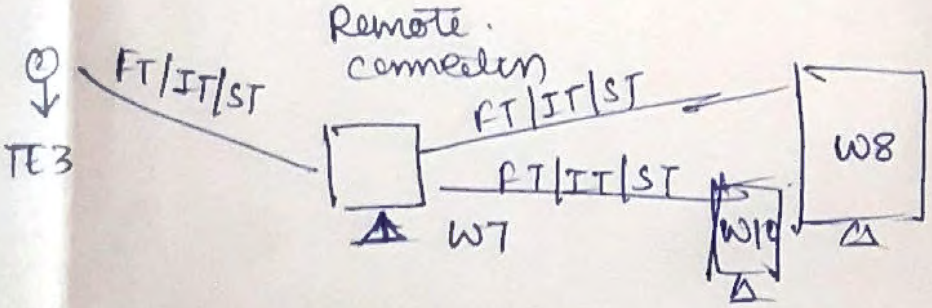
~~If he finds the defect &~~ If the app doesn't run on any of the os he will raise it as a defect that, that particular os is not supporting. But in this approach performance issue will be there so its not better approach.

3).



Here ~~same~~ ~~diff~~ ~~OS will~~ be Here we will be having only 1 sys m which a virtual m/c is installed which will allow to access that sys with W7 & W10 also & then the TE will be testing m all the OS.

4).



Here we can make remote connection with sys ~~diff~~ containg diff OS & check ~~IT on~~ the apps compatibility m all the OS.

Date: / /

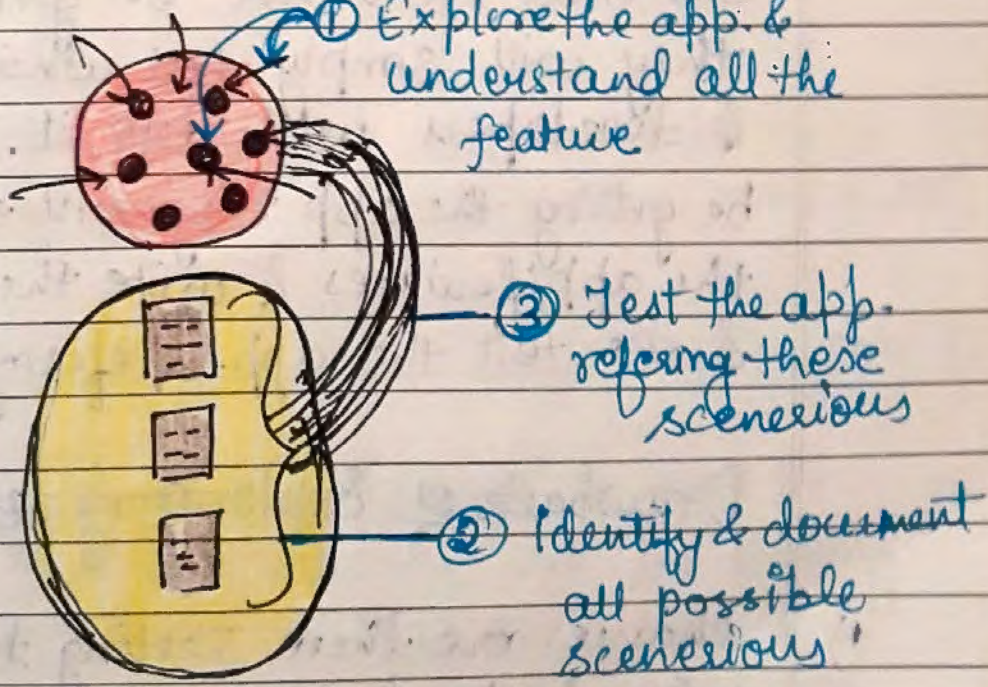
Exploratory Testing:

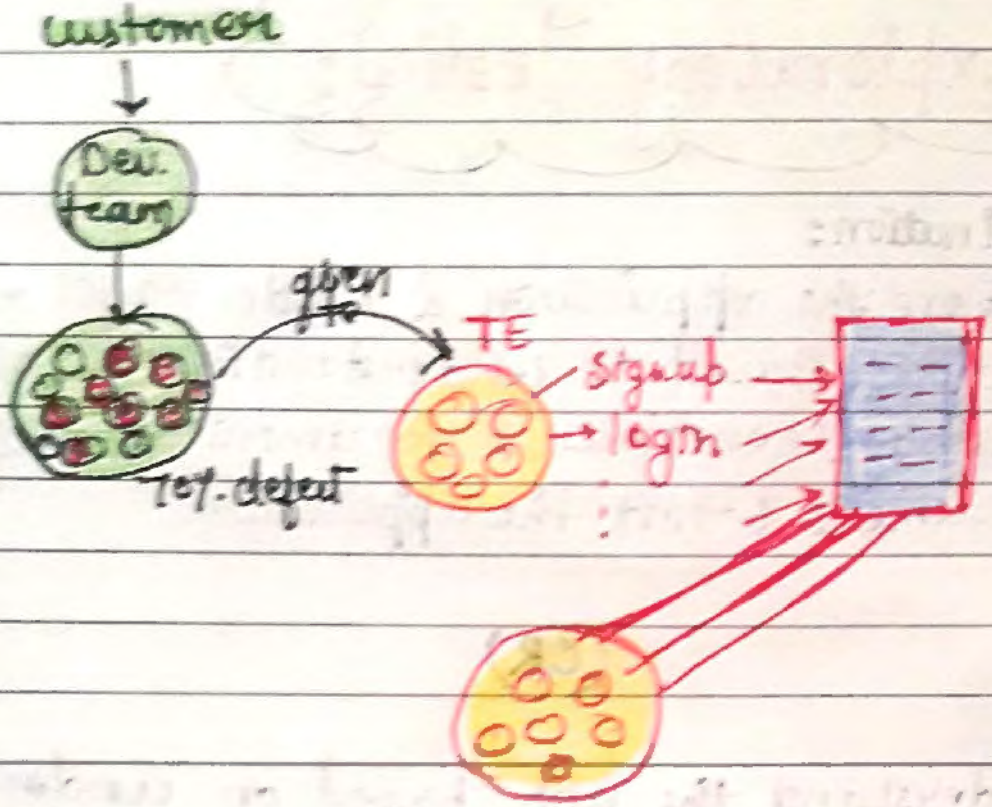
Defination:

Explore the application & understand all the features. Based on understanding Identify all possible scenerious & document it. Refer the document & test the application.

(OR)

understand the app. based on understanding identify scenerious & test the application.





customer will not be giving any CRS they will simply tell what they want & developers will plan it. Now TE's will be getting the app they will explore & understand the app features & write the possible scenerios & ~~the~~ test the app. Refering these scenerios.

Drawback of Exploratory testing:

1). Chances are these Testing team might misunderstand feature as a bug & defect as a feature.

If req. change or new req. comes & TE is not aware he will be checking with old understanding only.

Date: / /

2) If the featuring is missing we will never come to know that it is missed.

How do you work on the above problem?

1) By interacting with developer, BA & customer very closely.

2) Based on the domain knowledge try to test the application.

3) Based on the knowledge of the proj, which is already released to the production.

4) When ever there is no req. we can take error msg as a req. & test the application. (IT not tested)

How do they manage exploratory testing proj?

The TE what ever he understand he will be documenting it & here also we will be having test plans & test cases so that in future if he quits the job, other TE can perform the testing refering those TP & TC.

When or why we do exploratory testing?

1). When ever there is no requirement.

2). When ever there is a requirement but not understandable.

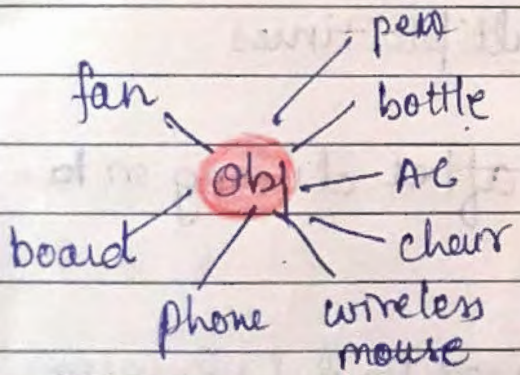
3) Req is there & its understandable but the time is not there to go through it.

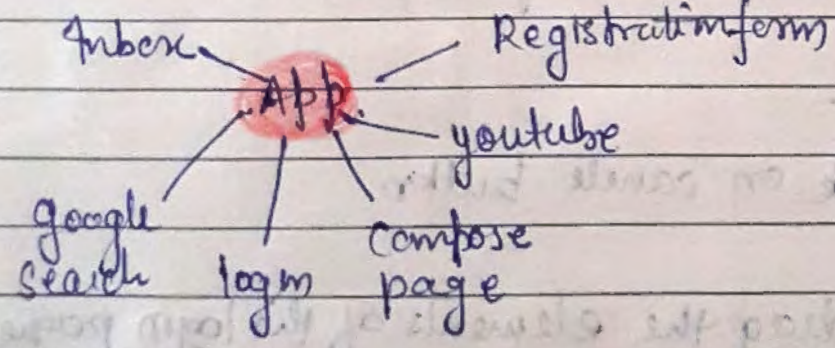
~~what?~~	Smoke	Adhoc	Exploratory
What?	Testing the basic or critical features of an app ~~without doing before doing through testing.~~	Testing the app randomly without any formal doc. like Req, TC & EP Test scenerious.	understanding the app. based on understanding write the possible scenerious & test ~~EP~~, the app going ~~through~~ these scenerious.
When?	When ever we get the new build	when the app is relatively stable	when we don't have req.
Why?	To check whether the build is testable	To ↑ the bug count	Req is not there

Date: / /

+ve or -ve	positive	negative	both
Req.	we ~~have~~ refer	we ~~don't have~~ refer.	we don't have.

How do you test ____ ? → obj | application.





ST, FT, IT, ST, Adhoc, compatibility, usability, acceptability, recovery, globalization, performance.

Date: / /

Globalisation Testing :

Defination:

Developing the s/w for multiple languages is called globalisation.

Testing the s/w which is developed for multiple language is called globalisation testing.

There are 2 types of globalisation testing:

1). Internationalization [I18N] testing.

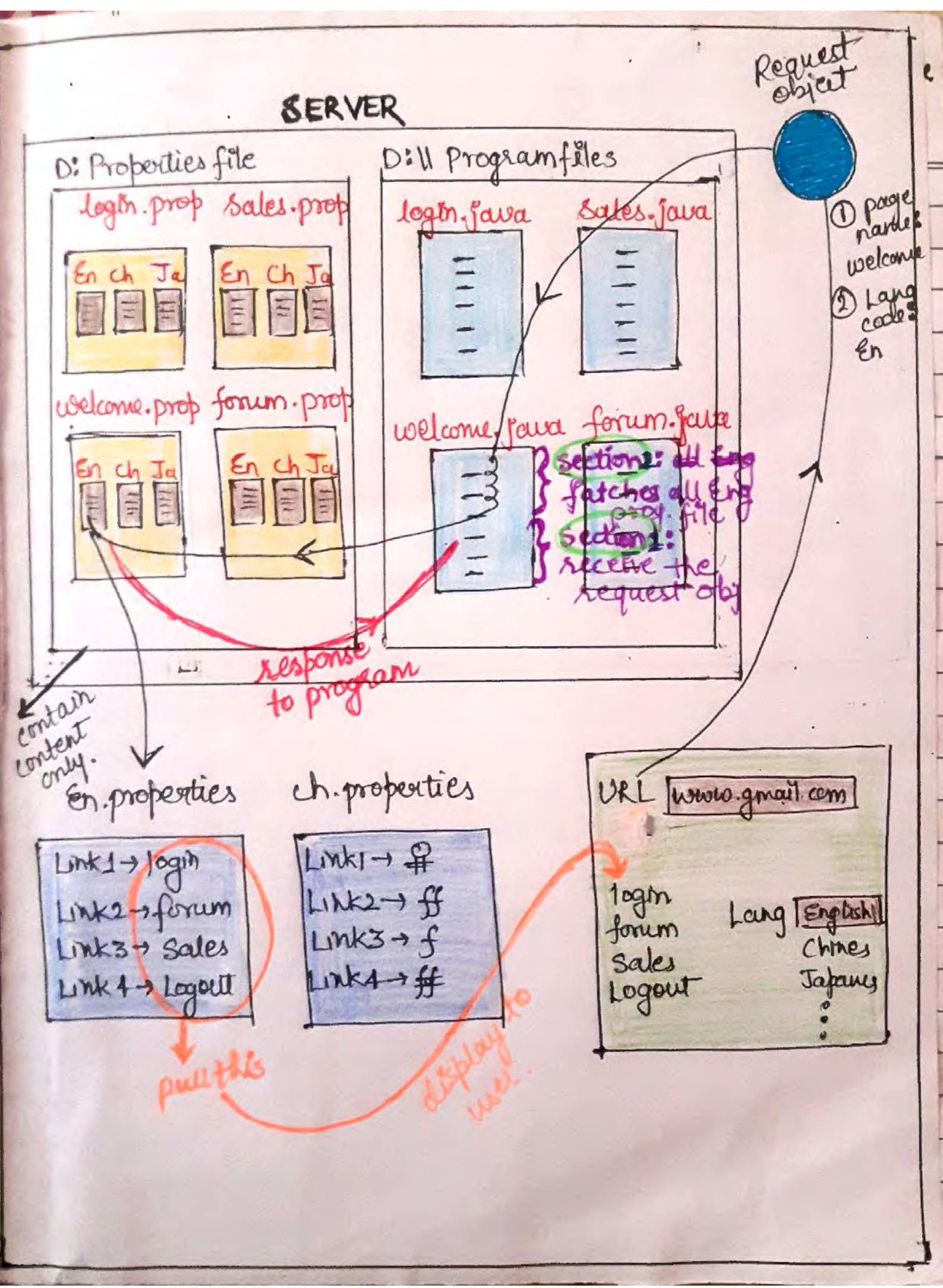
2) Localization [L10N] testing.

Any s/w developed for multiple lang will have 2 sort of files one is properties file & other is program file.

Previously translater tool were used but it don't work fine eg: welcome → Hindi

↓ ↓

कुशल आओ



Internationalisation (I18N) testing:

Here we check whether the right content is displayed in right language or not & right content is displayed in right place or not.

we do this by adding suffix & prefix to properties file & it is also called pseudo translation.

Localisation testing:

Check whether the features are changing locally according to country standard or culture is called localisation testing.

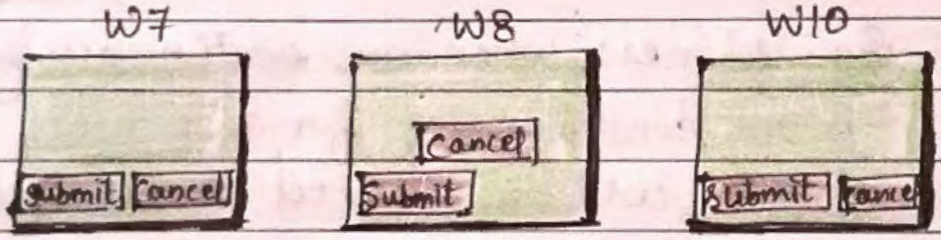
eg: time, currency,

Date: / /

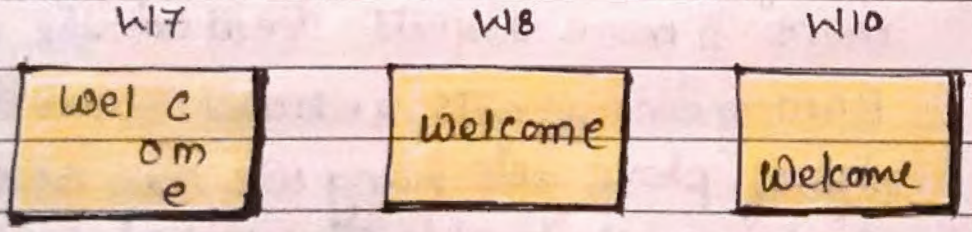
Compatability testing Defects

1. Alignment issue
2. scattered objects eg: Blurred mages
3. Object overlapping
4. change in look & feel (colors, font, size)
5. scroll bar display issue.

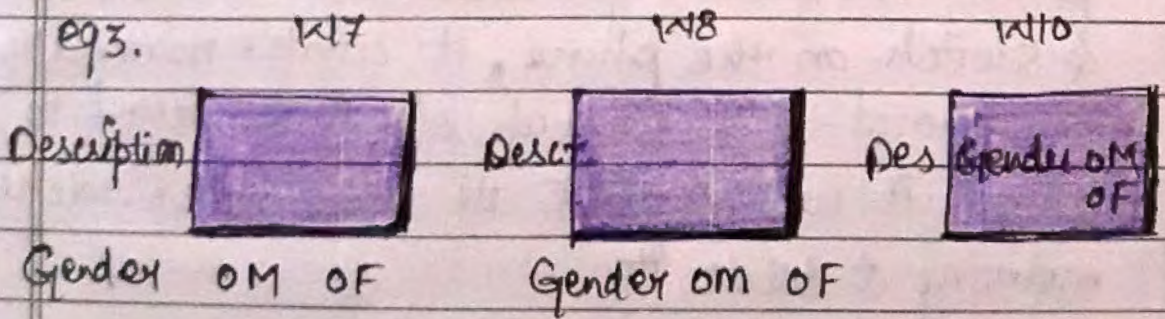
eg1.



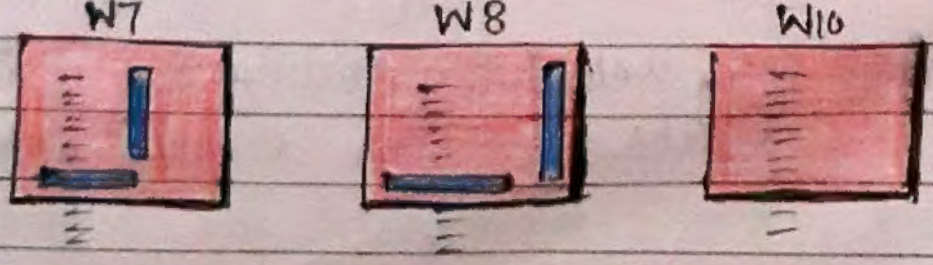
eg2.



eg3.



eg5.



Date: / /

Reliability Testing

→ Non-functional testing.

Defination:
Testing the applications functionality continuously for a particular period of time.

Eg: phones; when we ~~continuously~~ use the phone continuously for a period of time, the objects will be created in the RAM memory every time when we use the phone. When more & more objects created the size of the RAM memory gets reduced. Hence the functionality of phone gets hang we can switch off the phone so that objects created will be cleared & switch on the phone, it works normally. To avoid this cleanup s/w is installed in phone it will collect all junk files in cache memory & delete it.

For doing reliability testing we write an automated program or script & run it.

In order to do ~~reliblity~~ reliability testing we use automation tools like WEIBULL ++, RGA tool (Reliability growth analysis)

Recovery Testing

→ Non functional testing.

Defination:

Testing the application how fast it recovers from a crash or disaster is called as recovery testing.

only senior Test Engineers or trained test engineer perform recovery testing.

steps to do recovery testing:

1. Introduce a defect & crash the application
2. whenever the s/w crashes, it should display a error log/ crash log msg mentioning the reason for the crash.
3. It should kill its all process before it disappears. eg. Task manger which shows all the process running in the background wherein end user

Date: / /

can choose the process to end it

4. Re-open the application wherein all the previous settings should be restored.

eg1. Closing the laptop abruptly wherein chrome browser open with 5 sessions & later once we restart the laptop again go to chrome browser it should display a crash log, start a new session.

eg2. Network ~~Issues~~ Issues wherein from High ~~are~~ n/w area when a download process is happening & immediately travelling to low ~~area~~ n/w ~~to~~ area which will pause the downloading & once again we return to high n/w area it should start from the place where it was paused & continue the ~~down~~ downloading the rest of the file. If it starts from initial place again proper recovery management mechanism is not updated. Hence loss of data, loss of money, loss of time.

Date: / /

Accessibility Testing / ADA testing / 508 act testing

→ Non-functional testing.

Defination:

" Testing the application from a physically challenged person's point of view is called as accessibility testing".

ADA - American Disability Act.

It is a Non-mandatory testing. If the application has to be released to multiple disabled users than we do accessibility testing.

eg. Google maps.

we can't do accessibility testing manually by a normal tester. hence we use automation tools like.

1). Wave
2). Infocus.

Agile: Notes

papergrid

Date: / /

Agile:

It is a Iterative & Incremental approach
The main goal of agile is customer satisfaction by quick delivery of working software.

Principle of Agile methodology:

→ Customer can change the requirement at any point of development stage.

→ Release should be short

→ There will be good communication b/w customer, BA, developer & Test Engineer

→ It is simple model to adopt

→ Developers & test engineers will be doing lot of meetings at regular intervals in order to improve the quality of proces.

What is agile testing?

Testing the software by following the principle of agile methodology is called agile testing.

Types of agile methodologies?

1. Scrum methodologies
2. XP (Extreme Programming)
3. FDD (feature driven development)
4. Crystal clear.
5. Lean & kanban
6. ASDM (Adoptive software development method).
7. DSDM (Dynamic software development method).

Scrum methodologies / Scrum Procers:

1. **Release:** Combination of sprints is called release.
2. **EPIC:** Complete set of requirement is called EPIC. One EPIC consist of multiple features or modules.

user stories / stories / story cards:

Features or modules of functionalities is called as stories one EPIC consist of multiple stories.

Date: / /

Story Point:

It is rough estimation given by developer & test engineer to develop & test individual stories.

Ex: 1Story point = 9 hours

1 SP = 6

1 SP = 8

Story point should be in Fibonacci series.

QA story points estimated based on the following aspects:

Time spent on understanding story
write test scenarios
write test cases
Review test cases
Test case execution
Defect tracking.

Developer story points estimated based on the following aspects:

Understanding requirements
Design
Coding
Reviewing code

Date: / /

WBT

Time spent on fixing the defects.

Swag:

The rough estimation given by developer & test engineer for every individual stories in the form of hours.

Sprint:

Sprint is the actual time spent by developers & test engineer to develop & test one or more stories.

Sprint planning:

Sprint planning is a meeting conducted by scrum master on the first day every sprint explain the requirement Identify list of task to be done Assign task to engineer.

Scrum Master:

Scrum master is a person who is responsible for delivery of the software to the customer within a planned period of time. Scrum master will track all the activities done by

developer, test engineer & BA.
BA/ senior test engineer / senior developer/ project manager & even customer can become Scrum Master.

Scrum meeting:

It is a meeting conducted by a scrum master on the daily basis.
It is also called as **daily stand up meeting**
This meeting is strictly bounded for 15 to 20 minutes.

In this meeting we discuss below mentioned points

1. what you did yesterday.
2. what have you planned today
3. Are there any obstacles or impediments

Sprint retrospective meeting:

It is a meeting conducted by scrum master on the last day of every sprint In this meeting we discuss the following points.

1. what went well
2. What didn't go well
3. Are there any actions plan

Release retrospective meeting:
It is the meeting conducted by scrum master on the last day of every sprint.

Bug trace meeting:
This is the meeting conducted by the test engineer or Scrum master a week or 2 week before the release BA, TE, Developer & SM will be a part of meeting

In this meeting test engineer list all the open & pending bugs which are not fixed by the developers in the current & previous release as a team we will re-prioritize the defect from the customer's bussiness point of view & deide how many bugs should be fixed as part of a current release & how many bugs can be move to upcoming release:

Product Backlog meeting:

This is a meeting conducted by a BA or SM a week or 2 week before the release.
In this meeting test engineer & developer come up with list of all the pending stories which are not implemented as part of the current & previous release.
As a team we will re-prioritize all the stories from the bussiness point of view & move the stories to the next upcoming sprint or release.

Resume points:

Experience:

Worked on Agile Scrum Development process.
Involved in Sprint planning meeting, Daily Standup meeting & Sprint Retrospective meeting.

Fresher:

Excellent knowledge on Agile Scrum Development process.

Date: / /

Very good knowledge on sprint planning meeting.

Very good knowledge on Scrum meeting.

Very good knowledge on sprint Retrospective meeting.

ALM / QC

Date: / /

ALM: application life cycle management tool
Qc: Quality Center.

* It is a test management tool, used to manage, testing activities of the projects.
  ↓
  entire STLC

* Testing activities of the project involves
  1. Requirement management
  2. Test Design
  3. Test coverage
  4. Test execution
  5. Defect management
  6. Report management.

* History:

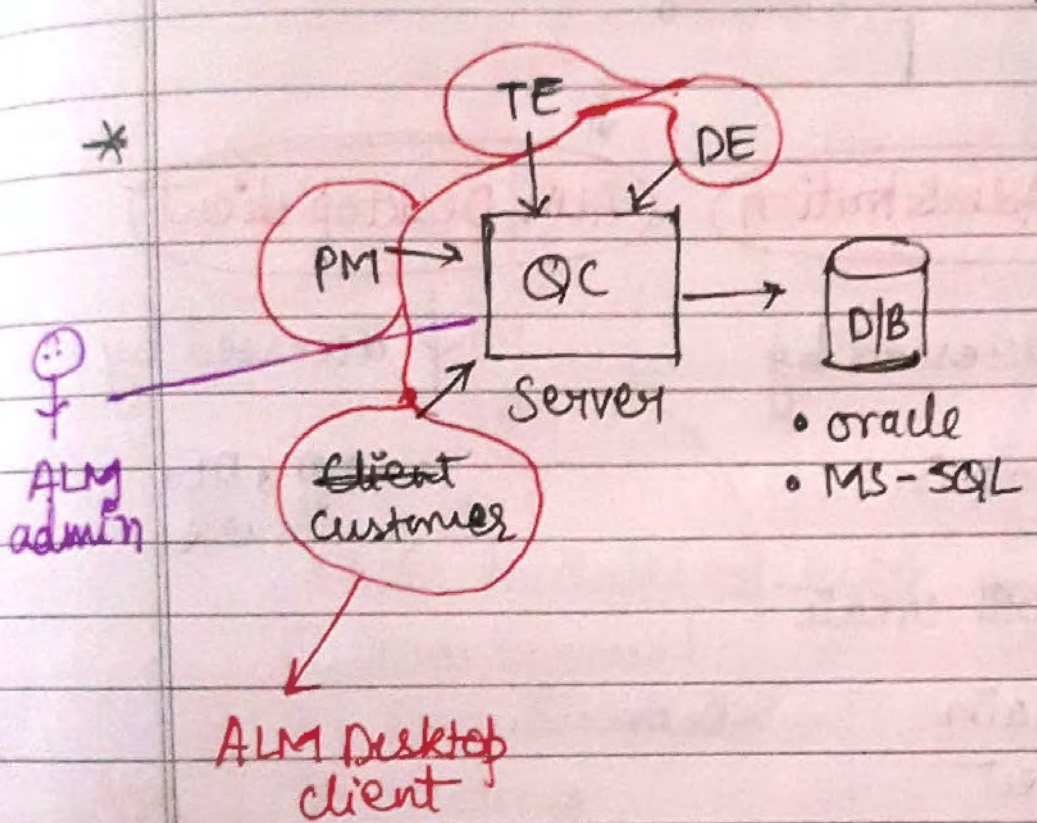
Mercury Interactive Cooperation —— TestDirector — 1985 — 1.0

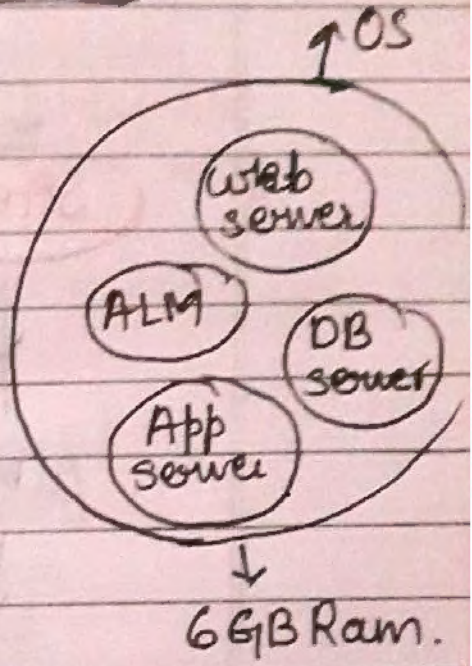
-"- —— QC — 2004 — 8.5

HP —— HP purchased & renamed as HPQC — 2007

-"- —— ALM —— 2011 — 11.5

current latest version — 12.53





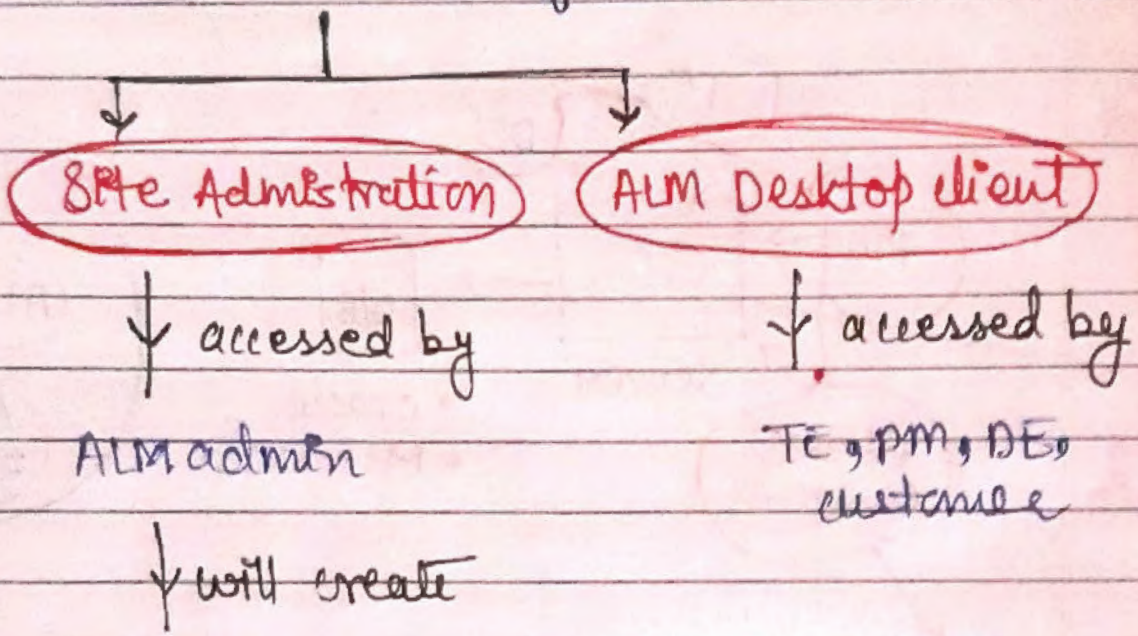
System admin	ACM admin
System Internet	Inside ALM tool

* QC will access DB, It can access only oracle or mysql database

* QC tool will be accessed by TE, PM, DE, customer.

Date: / /

* QC tool has 2 Interface



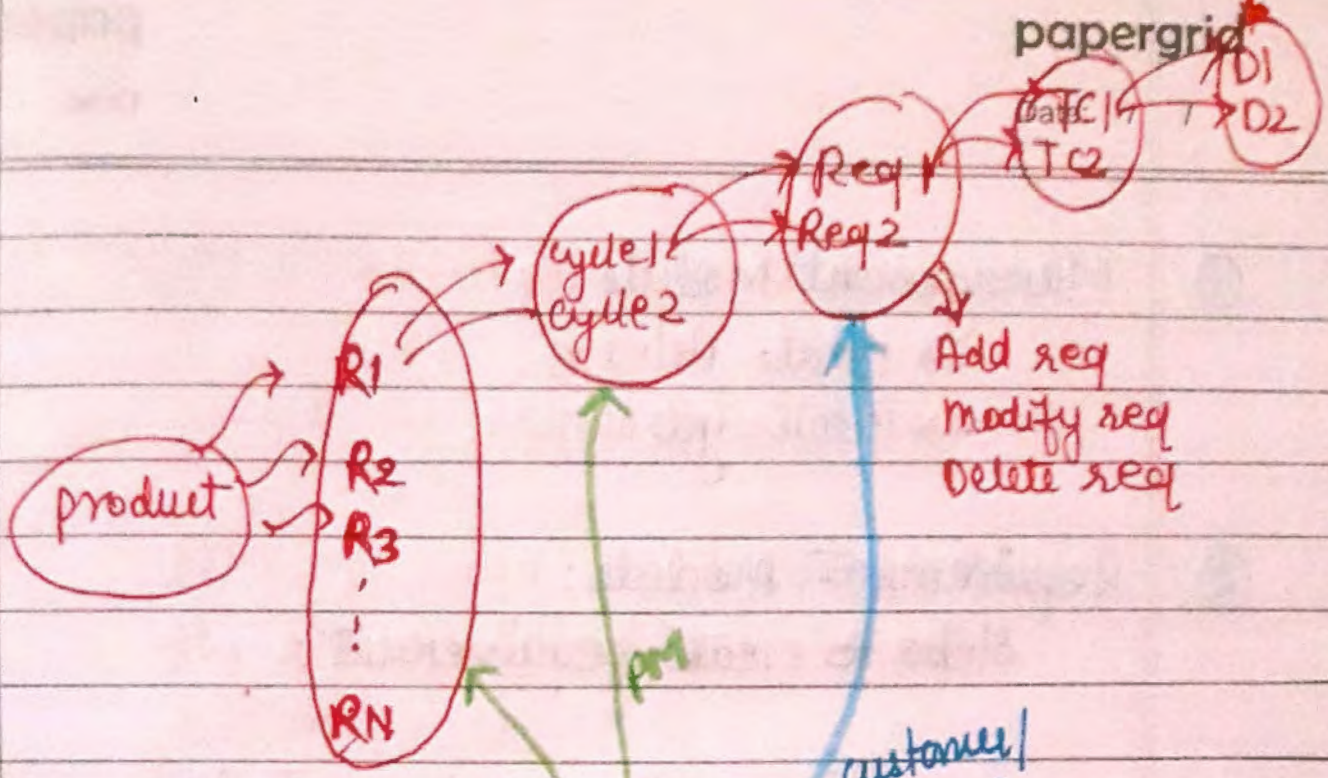
* To access QTP tool you need.

→ URL : https://Server name:8080/QC bin
(Server name → IP Address; 8080 → port No.)

→ UN
→ Pwd
→ Domain name
→ Project

While test execution

papergrid



customer/ BA/ PM/ you can read only

Main modules of ALM

1. Management
2. Requirement
3. Testing
   - → Test plan - write TC
   - → Test Lab - used for test execution, store copy of TC to be executed in each build.
4. Defect

TM

In XL

export but same template

If 50 build ⇒ 50 execution results.

BO1 → BO2

Defect (D1) (Raised Defect)

Check defect (D1) in next build. (Recheck defect)

① Management Module:
→ create Release
→ Create cycle.

② Requirement Module:
Steps to create requirement:

click on new requirement from
↓
enter the requirement name
& requirement details
↓
click on submit button.

- you can download the requirement in your local machine.

Que. How do you download requirement document from ALM?

Go to requirement module
↓
Open the requirement
↓

click on attachments option
↓
Select the requirement document
↓
click on Save as button, enter the file name & click on Save button.

(3) Testing module:

→ Test plan module:
It is used to write test case as well as automation test script.

Steps to write test case:

Go to test plan module
↓
Select the requirement test case folder (1st we should create)
↓
click on new test icon, & enter the test case name & objective of test case

Inside description
↓
click on ok button.
↓
click on new step icon,
enter the description
& expected result
↓
click on ok button
↓
Repeat the above
steps (last 2 steps) for
all possible scenerios.

Traceability matrix:

case 1: Backward traceability matrix (TC → Req)

Go to requirement module
↓
Open the requirement
↓
Click on test coverage
Icon.
↓

click on select icon

↓

select the required test case on the test plan tree

↓

click on left pointing arrow mark.

case 2: Forward traceability matrix (Req → TC)

Go to test plan module

↓

select the test case

↓

click on select requirement coverage

↓

click on add to coverage icon (left pointing arrow).

→ Test lab:

This module is used to execute manual test cases as well as automation test scripts.

Steps to execute test case:

① Steps to create test set folder:

Go to test lab module
↓
Click on new test set folder
↓
Enter the test set name & click on ok button.

② Steps to add test cases to execution grid

Click on select test case
↓
Select the test cases in the test plan tree
↓

↓

click on left pointing arrow mark

③ steps to execute test cases

Select the test case, in the execution grid

↓

click on run button

↓

click on begin run button

↓

see the description & perform action on the application.

↓

enter actual result in the expected actual result & the actual & expected result are same.

update the status as passed -

If the actual & expected result are different than update the result as failed.

↓

Repeat the above step for all possible scenerious.

↓

finally click on end run icon

Resume points:

→ Extensively work on test management tool ALM
→ Expertise in writing test cases & execution of test cases in ALM
→ Proficient in loggin & ~~or~~ tracking the defects using ALM
→ Prepared requirement traibility matrix using ALM.

White Box Testing

papergrid

Date: / /

Defination:

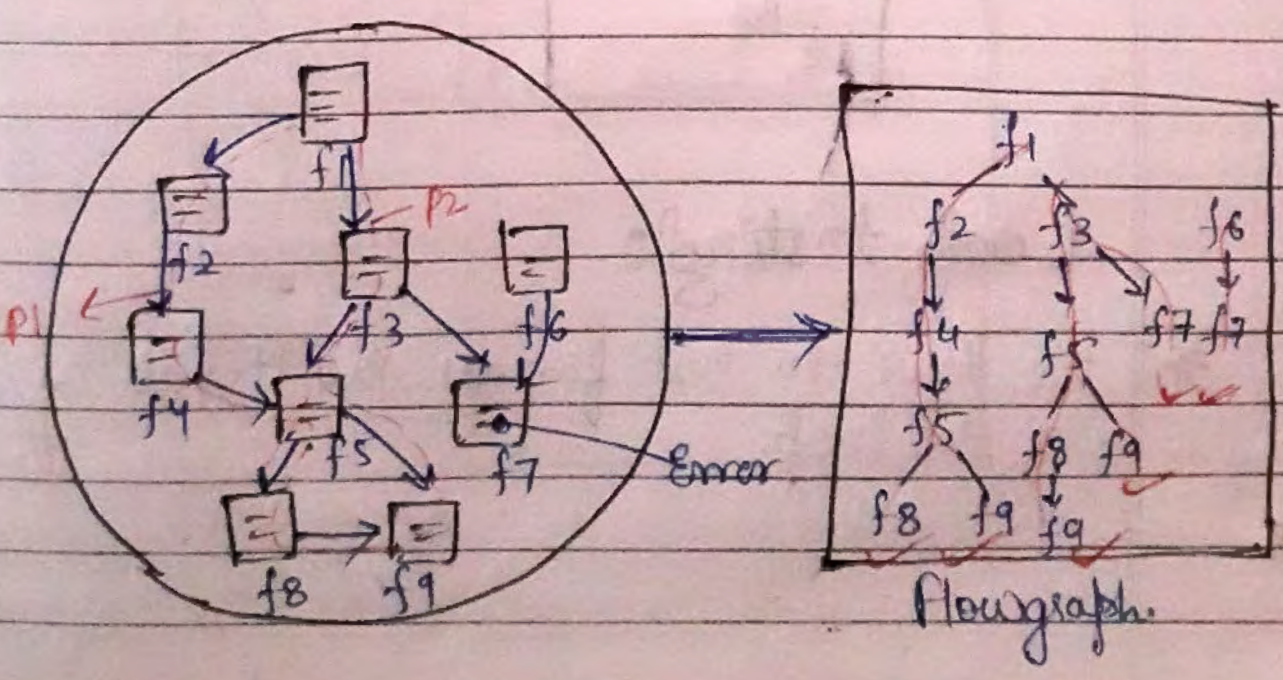
Testing each & every line of the program is called white box testing.

It is doen by developer before giving the build to the testing team in order to reduce the bug count. It can be doen manually or ~~can be~~ testing automated

Types of white box testing:

1) Path testing
2) condition testing
3) Loop testing.

Path testing:



* Write the flow graph & test all the independent path.

* When ever there is a defect fixes we should test all the dependent paths.

Condition testing

* where in we should test for both true & false values.

```
if (condition)
  {
  = B1
  }
else
  {
  = B2
  }
```

Loop testing:

* We have to test for all the Iterations of the loop.

1 2 3 4 5 6 7 8 9 10

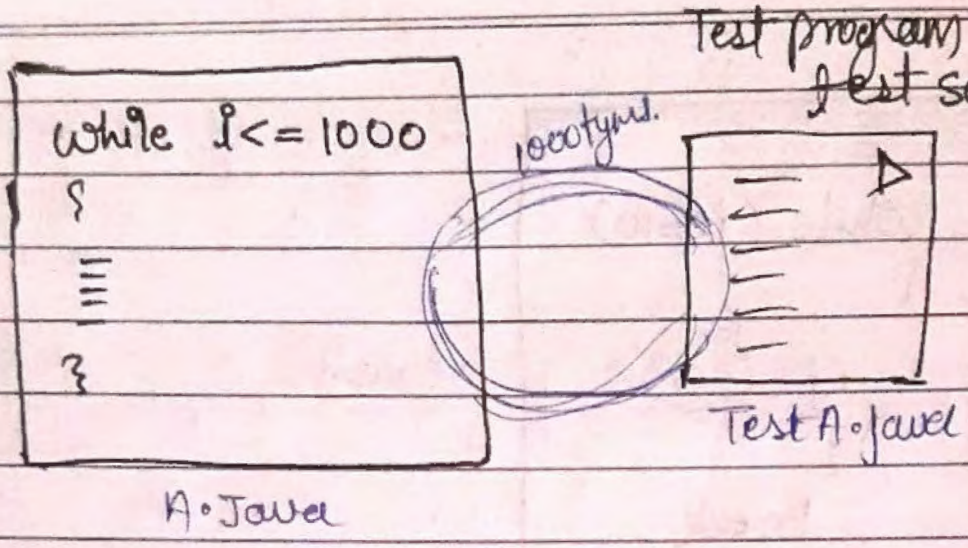
```
while (i<=10)
{
    if(i=5)
    {
      Break
    }
}
```

crash.

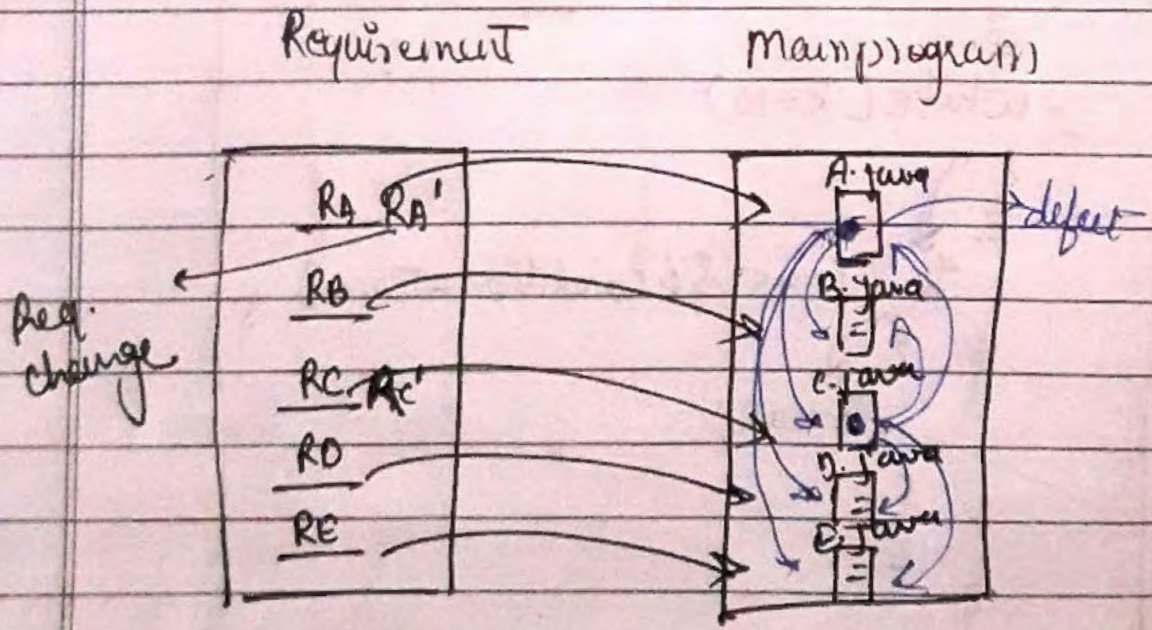
```
Condition = True        false        CheckBox ☑
while(i<=10)
{
    if (i==5 && Condition=True)
    {
      Break;
    }
}
```

i = 1, 2, 3, 4, 5, 6

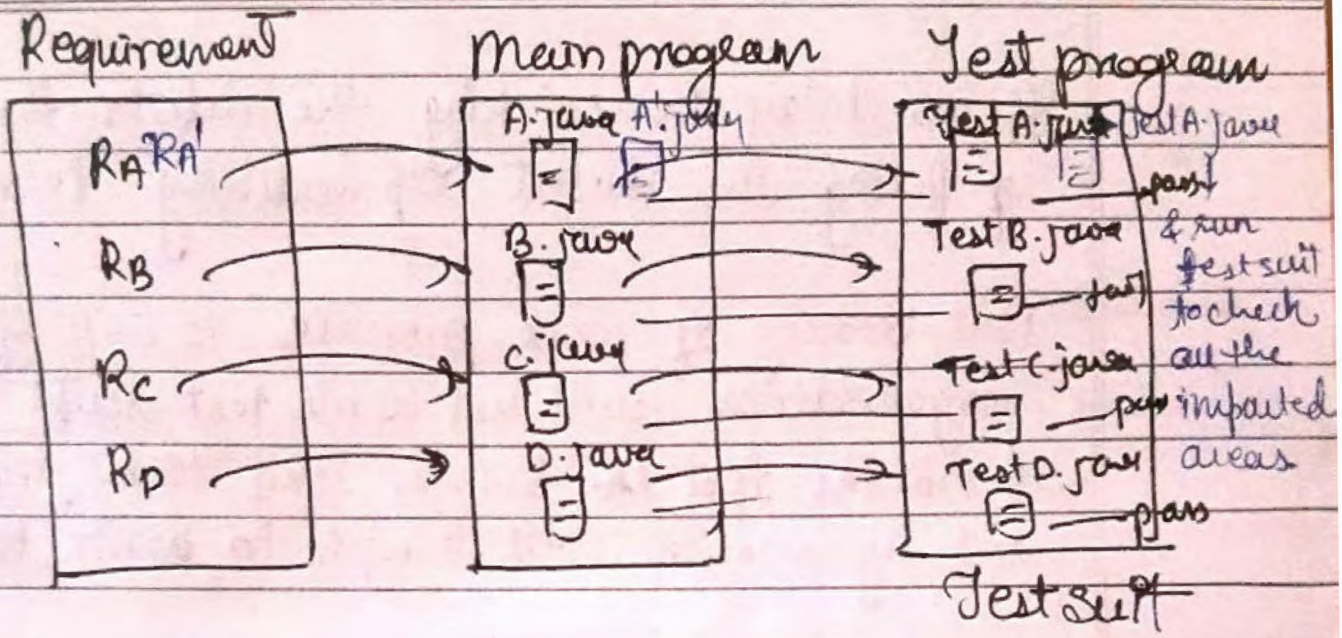
Date: / /



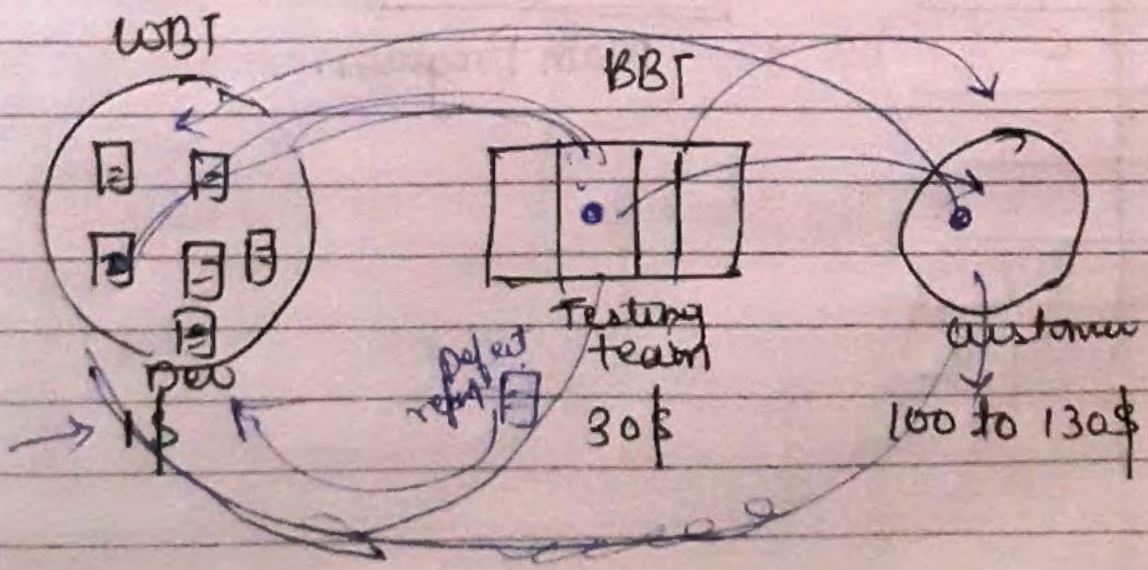
When no. of iterations will be more, developer will write another program (called test script) in order to check all the iterations of the program.



Problem: Developer will ~~spend~~ waste more time on checking the impacted areas



now time will be spend on doing changes.
~~not on~~
For long term project, doing manual WBT will be very tough. so we need to write test script.

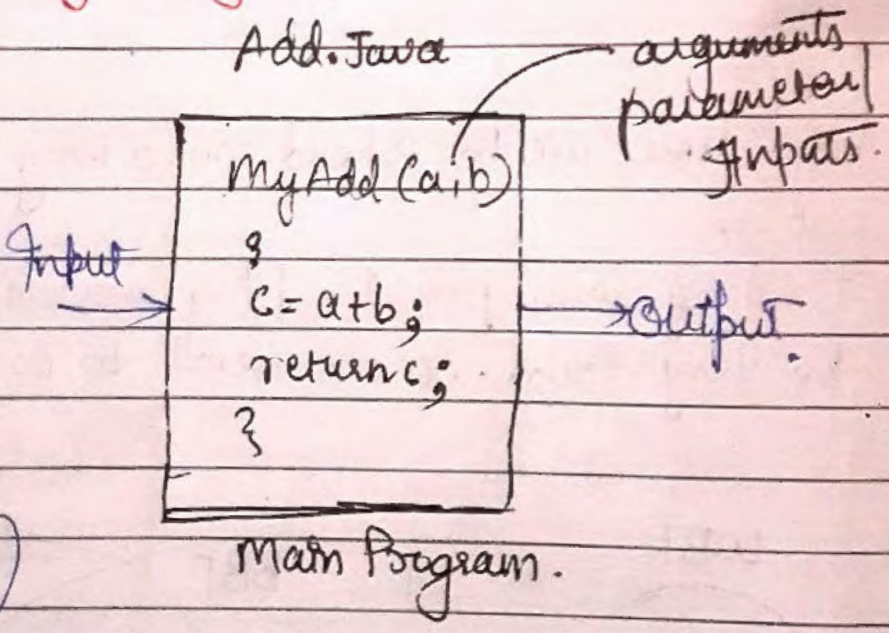


Date: / /

If we delay in catching the defects the cost of fixing the defect exponentially increases.

But because of work pressure 70-80% of organisation will not write test script & so the test ln. but for long term projects 20% organisation will, comple to wrote test scrip.

Manually doing UOBT :-



Input		o/p
a	b	c
10	20	30
10	-5	5
5.2	35	8.7
10	A	Error

Test Scenarios.

WBT using automation:

Input a	Input b	output c	
10	20	30	Test scenarios
10	-5	5	
52	35	87	
10	A	Error	

Add.Java

```
myAdd(a,b)
{
c=a+b;
return c;
}
```

i/p → o/p

Test Add.Java

```
x = call Add.myAdd(10,20)
If(x==30)
{ print 'pass'
}
else
{ print fail
}
```

(matching) pass ← compare (x, 30) → (not matching) fail

Test 1 Add.Java

```
x = call Add.myAdd(10,A)
If(x==Error)
  print pass
else
  print fail
```

Test 2 Add.Java

```
x = call Add.myAdd(10,-5)
If(x==5)
{ print 'pass'
}
else
'print fail'
```

Test 3 Add.Java

```
x = call Add.myAdd(52,35)
If(x==87)
'print pass'
else
{ print 'fail'
}
```

compare (y, 5)

so we observe

size of test prog is greater than main prog

Scanned by CamScanner

papergrid

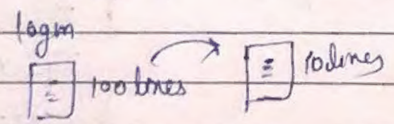
Date: / /

Reducing the No. of statements without changing the output is called ~~cead~~ code optimisation.

White box testing from memory point of view:

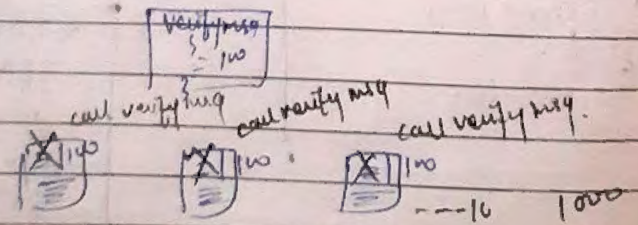
Q. What are the typical mistakes doen by the developers so that which increases the memory size?

1. Because of logical mistake.



2. ~~Because of not using in built functions.~~ → msg sent fun is available but we have written again

3. Because of not using funtions for repeated steps.

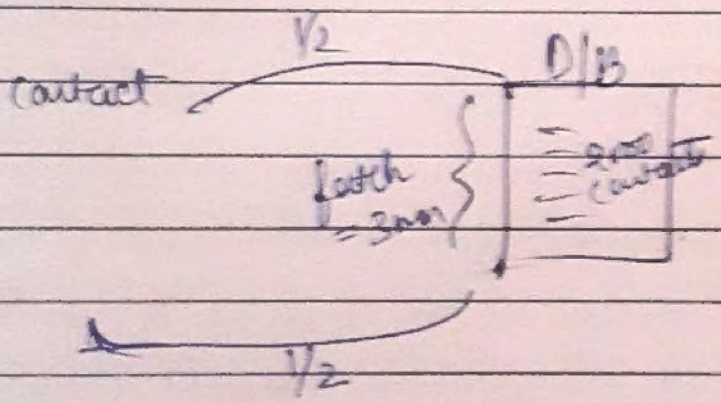


④ Because of unused variable, functions & objects.
~~⑤ Performan~~

White Box testing from performance point of view:

Q. What are mistakes done by the developer which is reducing the response time?

①. Logical mistake.



Total response = 4 min

↓ optimise

fetch only 20 contacts & display 20
next fetch 20 contacts & display next 20
~~(it will take~~ (It will take same time but customer will get feel that its ~~taking~~ fast)

eg. In amazon (keep adding products in cart)

Date: / /

(2). Because of using nested condition instead of switch statement.