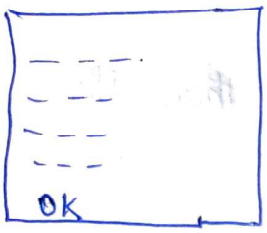
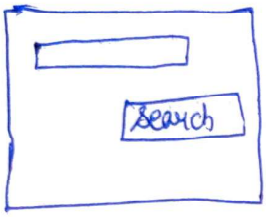
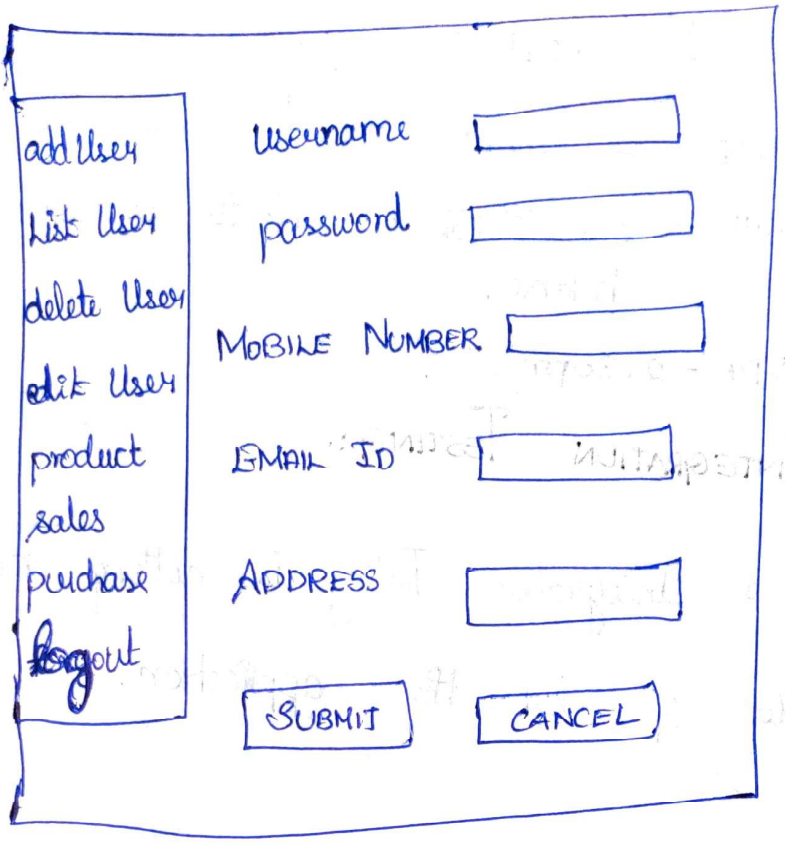
PRIORITY OF SCENARIOS :-

List User

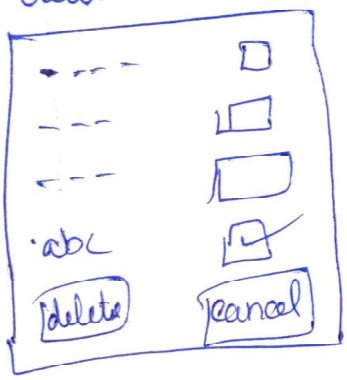


Search User

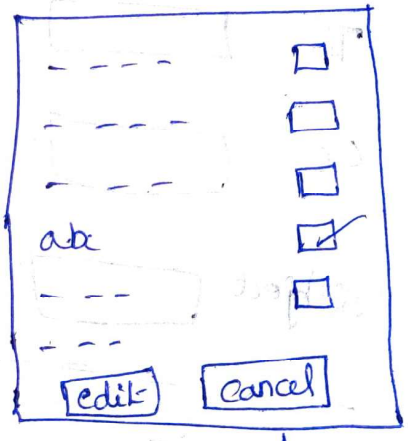


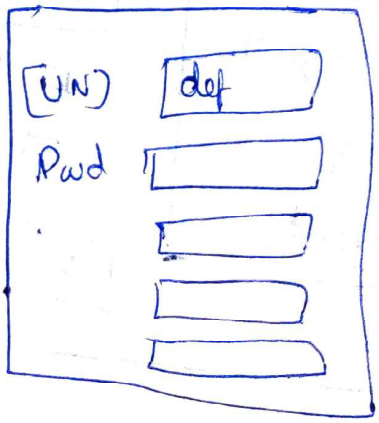


delete User

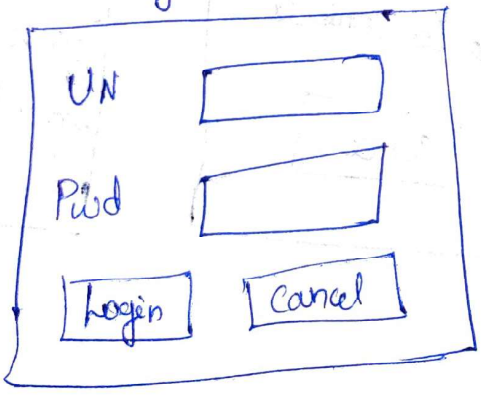


edit User





Login

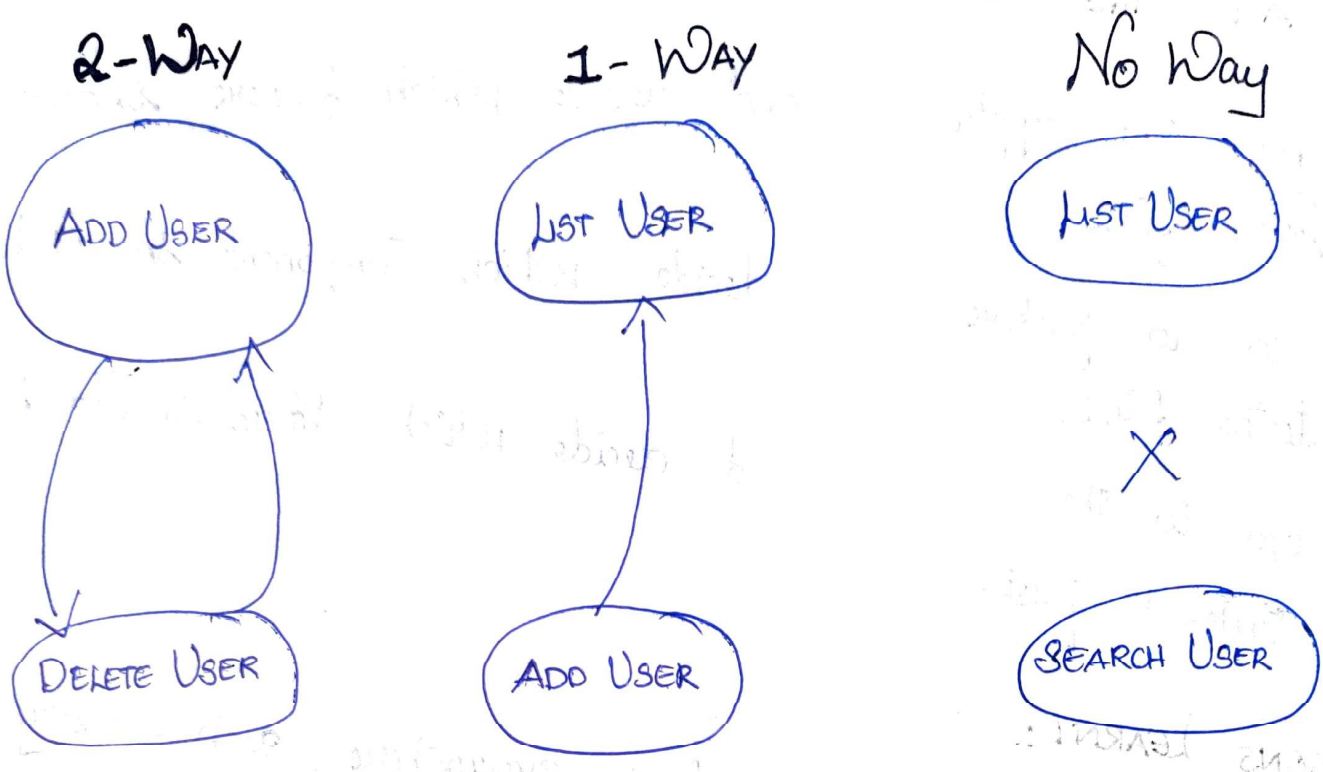


PRIORITIZING IS VERY IMPORTANT AND IT SHOULD BE DONE IN ALL THE STAGES:

1. Open the application and decide which feature should be tested first.
2. Go to feature and decide which Component should be tested first.
3. Go to the Component and decide which Value should be Entered first.

LESSONS LEARNT:

1. Don't apply same logic everywhere, each & every feature has different logic.
2. Completely test 1 feature, then Only move to other feature.
3. First do through functionality testing and then do Integration testing. directly don't start with Integration testing.



Between ~~two~~ ② features we might be doing both +ve and -ve testing or we might be doing only positive integration testing it just depends on how each and every module is related.

Between 2 features we might be doing integration testing in 1 way or in 2 ways or we may not do integration testing, it Completely depends on how Each and Every feature is related.

How They Allocate Work in Real Projects:

1. Compose -> C
2. Sent Items -> Critical
3. Inbox -> Critical
4. Drafts -> Major
5. SignUp -> Critical
6. Calendar -> Minor
7. Trash
8. Logout
9. Outbox.
10. Spam
11. All mails -> Critical
12. Contacts -> minor.
13. Settings
14. Starred -> Major.
15. Attachments
16. Start Meeting
17. Bin
18. Archives
19. Feedback
20. Drive.
21. Schedule -> minor
22. Remember Pwd -> Critical
23. Forgot Pwd -> Critical
24. Manage Accounts ->
25. Help
26. Tasks.
27. Notes
28. Unread
29. Unsubscribe.
30. Important
31. Login

(1) Group all the related features and keep the independent features for balancing.

(2) Assign the grouped features to the test Engineer.

(3) Monitor them for few cycles.

INTERVIEW QUESTIONS:

① What is Integration testing ??

- - -> def
- - -> how to do integration (6 points)

② How they allocate work in real time projects ???

TASKS:

1) Identify 100 scenarios on Any Payment Application.
2) Identify 100 scenarios on Any Ecommerce Application.

3) Identify 100 scenarios on OLA, UBER

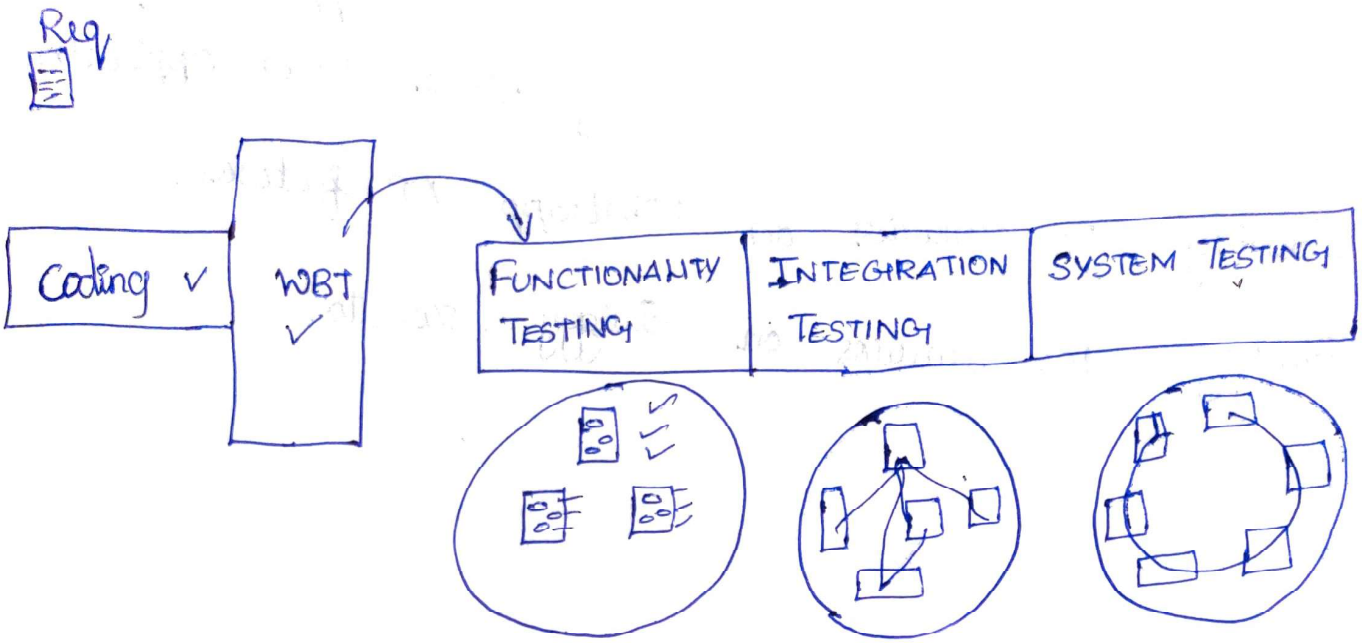
4) Identify 100 scenarios on Any travel applications.

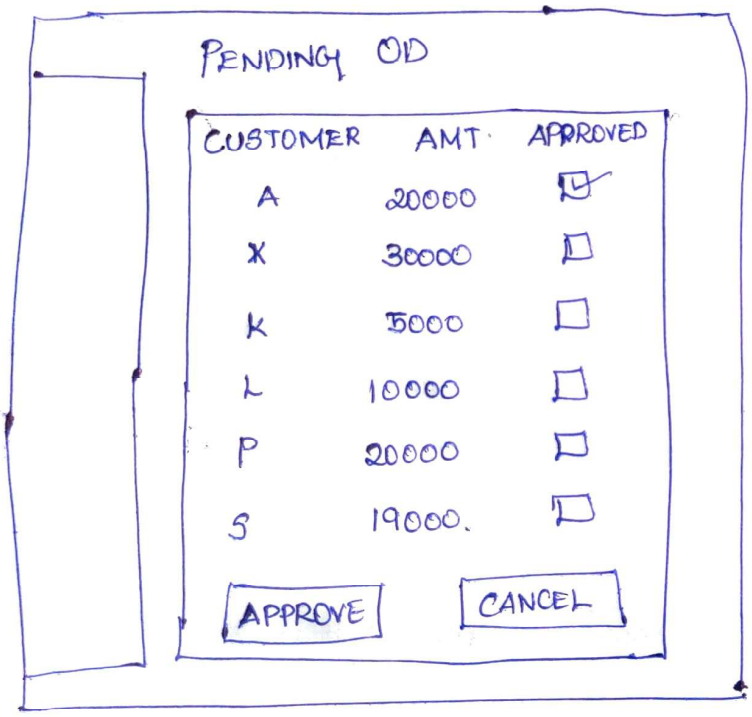
5) Identify 100 scenarios on Any social media applications.

6) Identify 50 scenarios on whatsapp DP features.

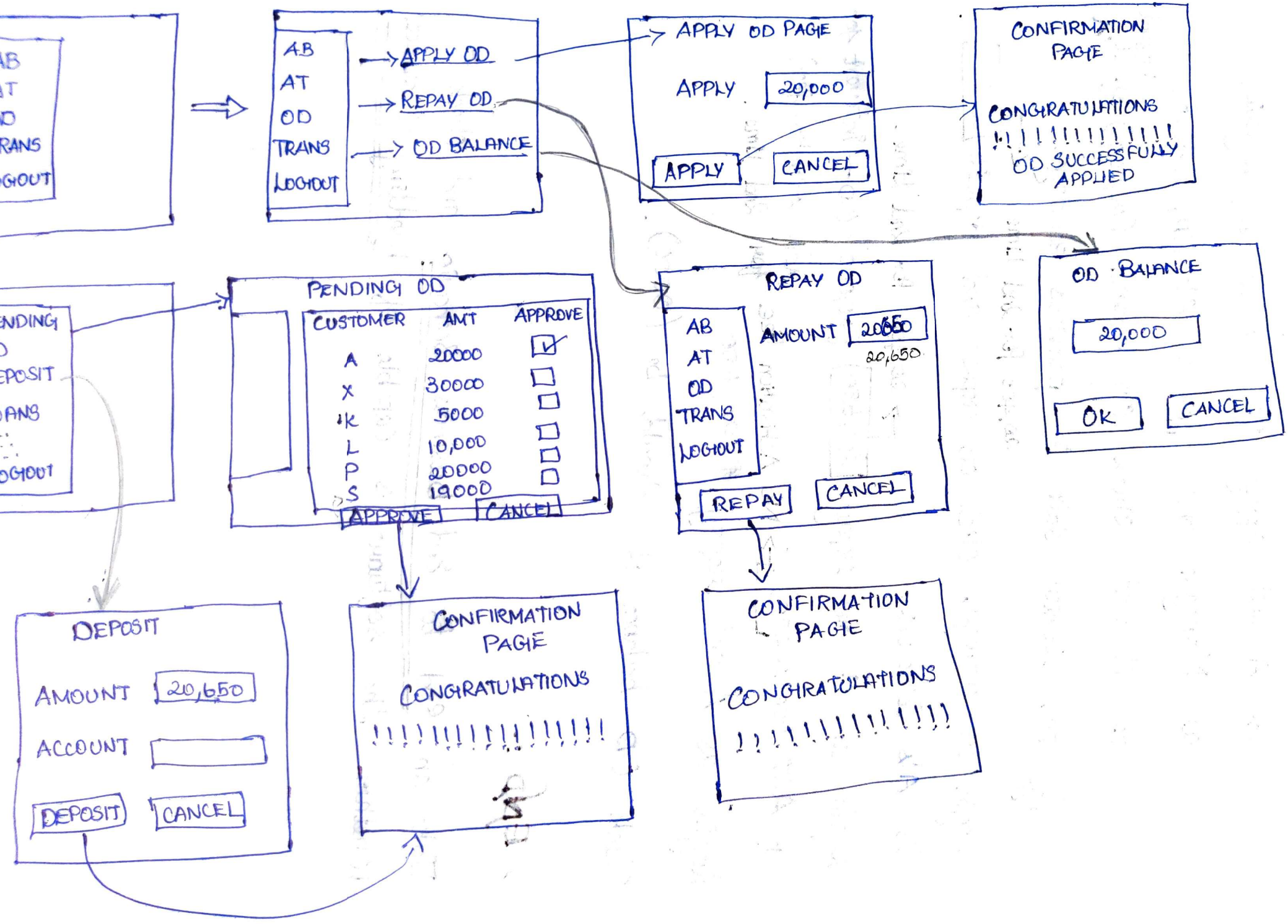
7) Identify 100 scenarios on Swiggy, zomato.

System Testing:-





Def. It is an End to End testing wherein test Envr is just similar to production environment.


AB
AT
OD
TRANS
LOGHOUT
APPLY OD
REPAY OD
OD BALANCE
APPLY OD PAGE
APPLY
20,000
APPLY
CANCEL
CONFIRMATION PAGE
CONGRATULATIONS
!!!!!!!!!!!!!!!
OD SUCCESSFULLY APPLIED
OD BALANCE
20,000
OK
CANCEL
REPAY OD
AB
AT
OD
TRANS
LOGHOUT
AMOUNT
20650
20,650
REPAY
CANCEL
PENDING OD
CUSTOMER
AMT
APPROVE
A
20000
X
30000
K
5000
L
10,000
P
20000
S
19000
APPROVE
CANCEL
DEPOSIT
AMOUNT
20,650
ACCOUNT
DEPOSIT
CANCEL
CONFIRMATION PAGE
CONGRATULATIONS
!!!!!!!!!!!!!!!!!!!
CONFIRMATION PAGE
CONGRATULATIONS
!!!!!!!!!!!!!!!

END TO END SCENARIOS :-

1. Login as A and apply OD for 20,000, Logout.
2. Login as Manager and approve A's OD, Logout.
3. Login as A and check OD balance (20,000) of A, Logout
4. Change the server date to previous 30 days.
5. Login as manager click on deposit and deposit 20,650 (20,400 for 2nd time)
   to A's Acccount.
6. Login as A and repay 20,650 (20,400 for 2nd time.
   (20,000 OD + Interest 400 + ACT Fee :- 250) (ACT- Activation fee)
   → No Activation Fee for 2nd time.
7. ~~Login as A and repay 20,650~~
   Check OD balance it should be (0000).

END TO END :-

Def :- Take an End to End business Scenario and check whether the software is capable of handling it or not

ASSIGNMENT:-

on Monday

End - to - End Scenario.

Req

1st time :-

OD
act fee
int

2nd time :-

OD greater than 50,000
then
activation fee should be refunded

10,000
250
400
10,650

60000
+ 400
60,400

60,400
- 250
60,150

Integ

REQ 2:

1. Manager can set the activation fees, from that moment onwards, if anybody applies for OD the latest activation fees should be charged.

(I)

AM

1) login as manager set Activation fee as 250 & logout.
2) login as User A and apply for OD 40,000 & logout
3) login as manager and approve A's OD and LOgout.
4) change the Server date.
5) login as User A and Click on Repay OD (40,650)
6) login as manager and deposit 40,650 to A Account.
7) login as User A and repay the OD amount (40,650).

(II)

① login as ~~User~~ Manager & set activation fee 50 & Logout.
② login as User A & apply for OD 20,000
③ login as manager and set activation fee 50
④ login as manager and approve A's OD and logout.

(5)

(6) login as User A and Click on Repay OD 20,450 logout.

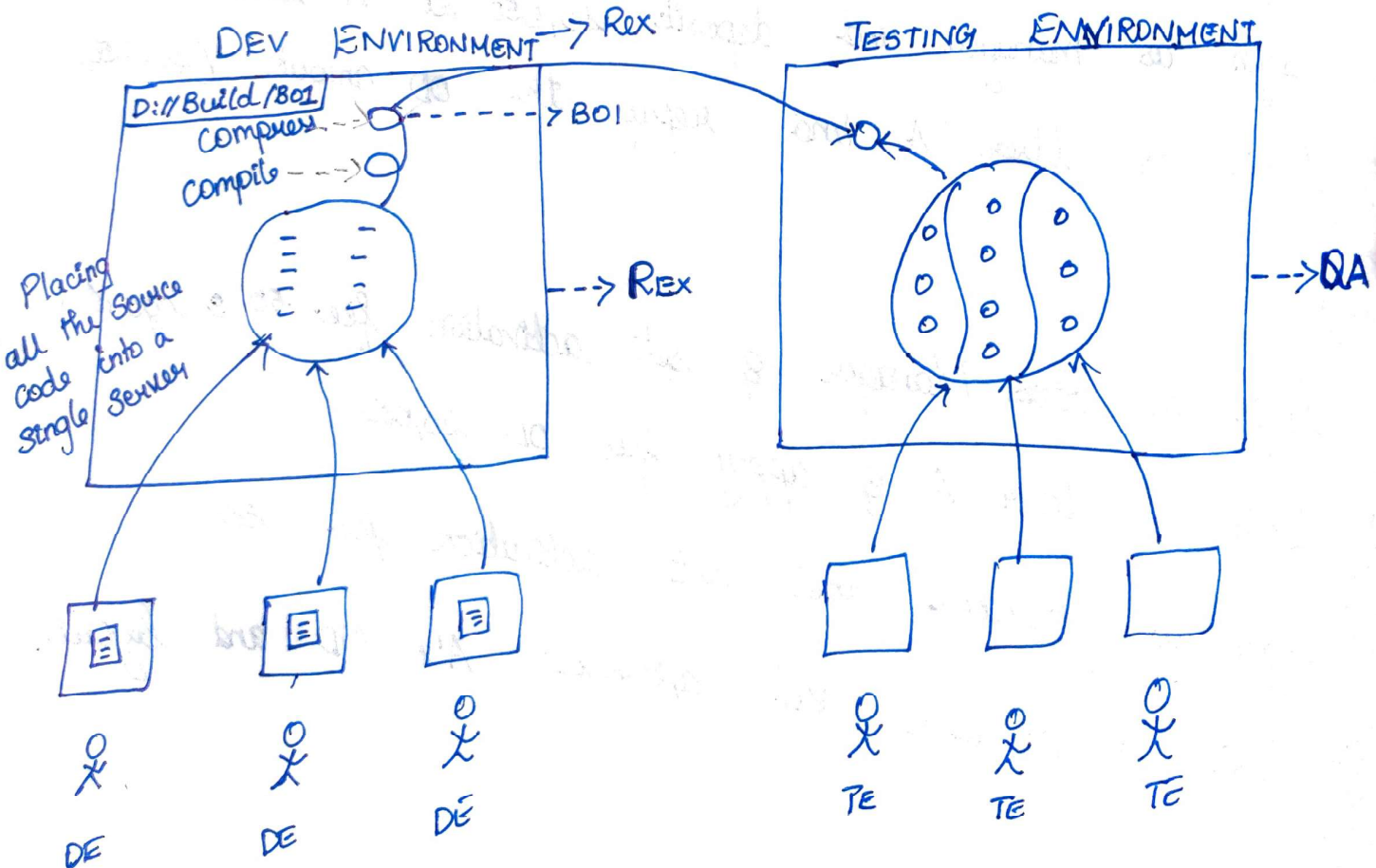
(7) login as manager and deposit 20,450.

(8) login as User A and repay the OD amount 20,450.

LESSONS LEARNT :

* Never try to assume the requirement.
* Never try to give suggestions regarding the features to the Customer.

DEVELOPMENT ENVIRONMENT :

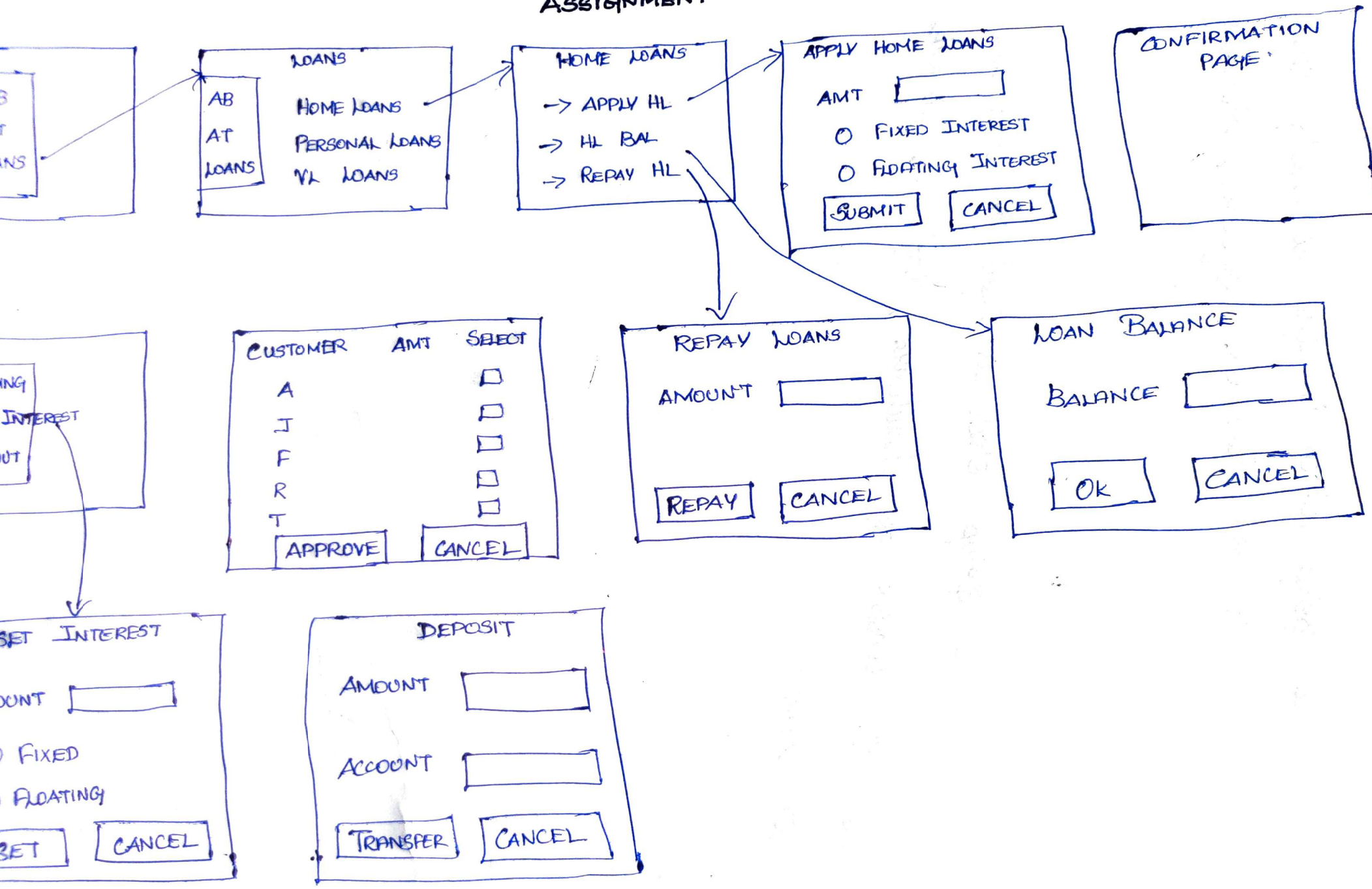


It is a setup which is used to, Develop the software. It consists of hardware, software, Networks and servers.

TESTING ENVIRONMENT:

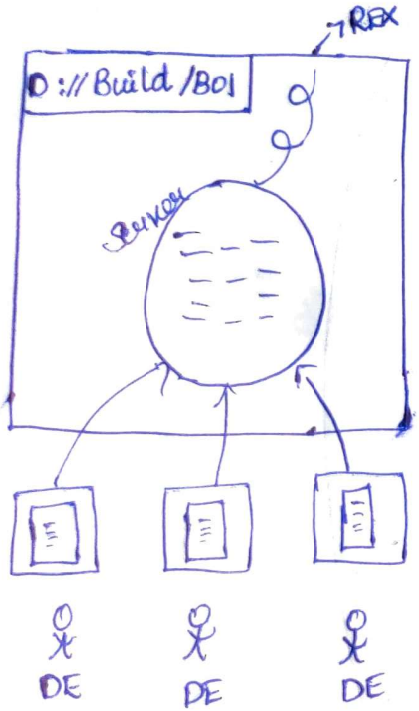
It is a setup which is used to test the software. It consists of hardware, software, networks and the servers.

ASSIGNMENT


LOANS
AB AT LOANS
HOME LOANS
PERSONAL LOANS
VL LOANS
HOME LOANS
-> APPLY HL
-> HL BAL
-> REPAY HL
APPLY HOME LOANS
AMT
FIXED INTEREST
FLOATING INTEREST
SUBMIT
CANCEL
CONFIRMATION PAGE
CUSTOMER
AMT
SELECT
A
J
F
R
T
APPROVE
CANCEL
REPAY LOANS
AMOUNT
REPAY
CANCEL
LOAN BALANCE
BALANCE
Ok
CANCEL
INTEREST
FIXED
FLOATING
CANCEL
DEPOSIT
AMOUNT
ACCOUNT
TRANSFER
CANCEL

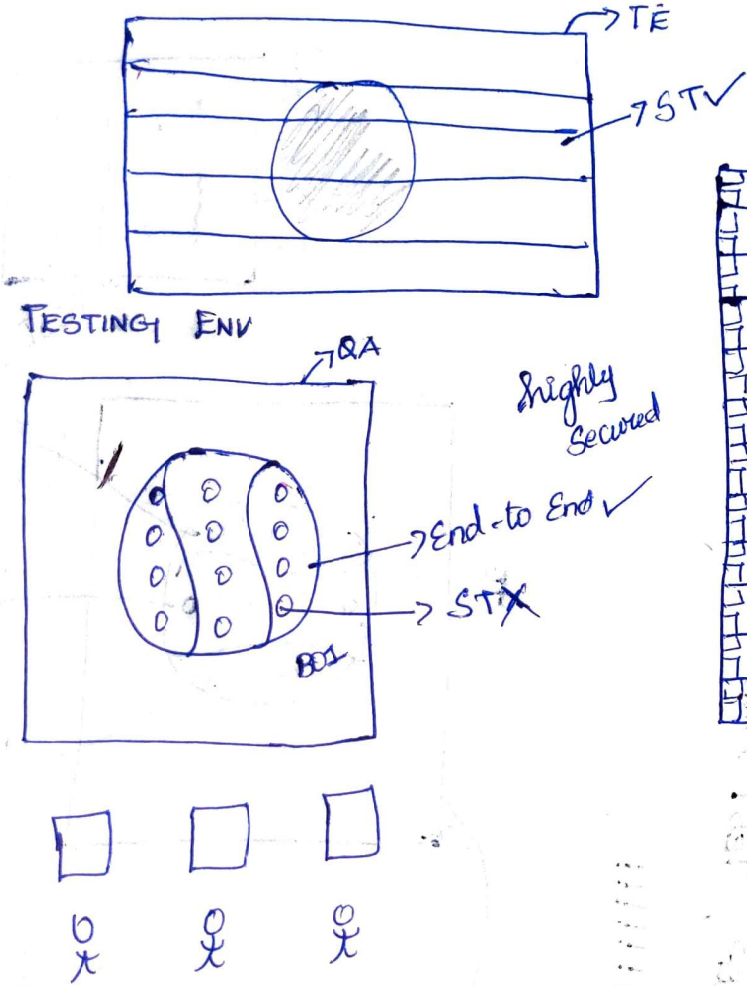
18/08/2022 (12:30pm - 2:30pm)

DEV ENV



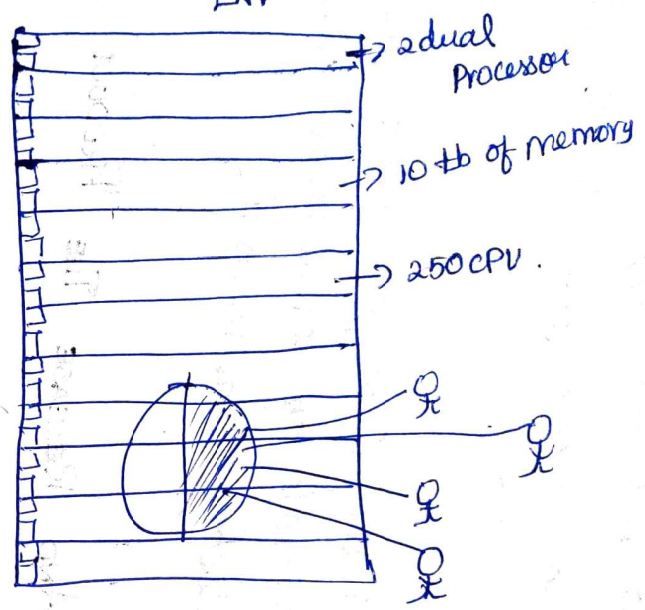
- It is an setup which is Used to develop the s/w.
- It Consists of ~~N/w~~ hardware, s/w, Networks, Servers

TESTING ENV



- It is an setup which is Used to test the s/w.
- It consists of hardware, s/w, Networks, servers.

PRODUCTION ENV



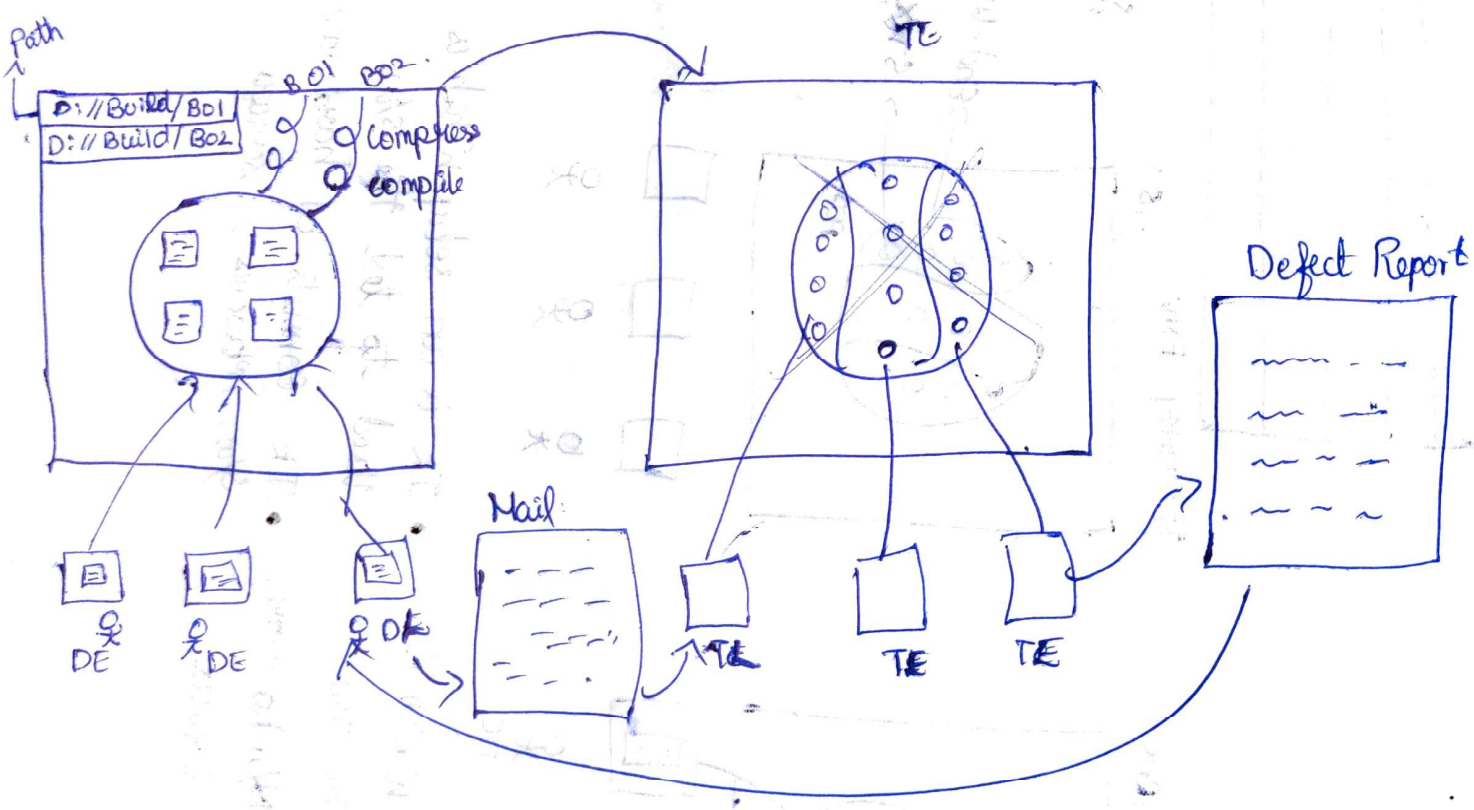
- It is an setup where we run the software.
- It consists of hardware, s/w, Network, servers.

Lessons Learnt :-

(1) The setup in which we test the application is different and the setup in which you launch the application is different.

(2) Customers cannot provide their own production Environment. so they will provide an environment which is similar to the production environment.

PROCEDURE TO DEVELOP, TEST AND RELEASE THE PRODUCT IN REAL TIME PROJECTS!!!!!!!



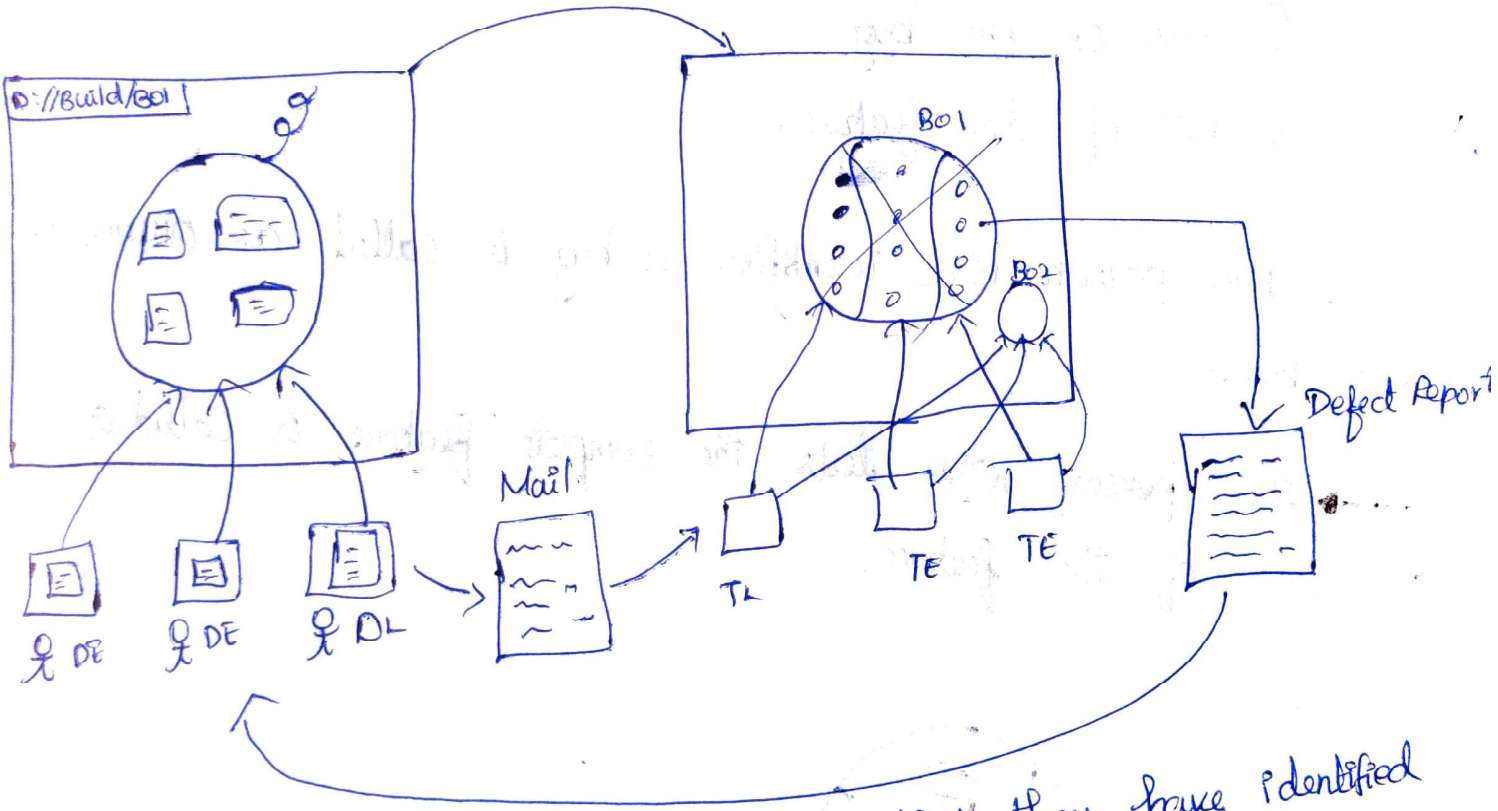
Lessons Learnt:-

① He Catch the bugs in the Testing Environment.

② He fix the bugs in the development Environment.

Note:-

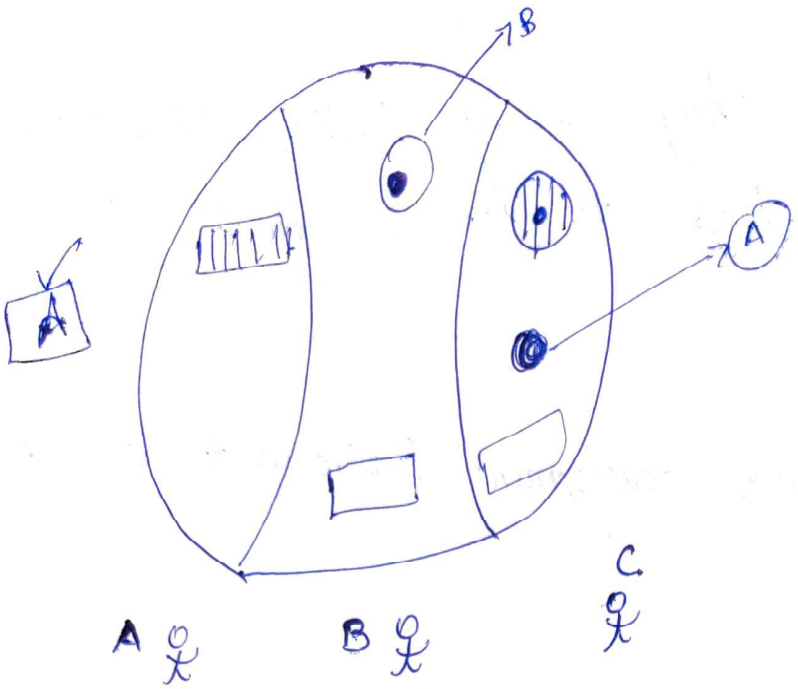
Never try to mix development Environment and Testing Environment.



* TL will send a mail to DL saying that they have identified 100 bugs.

* Now DL will reply saying that they have fixed Only 60 bugs.

Note:- test only the bugs which is been fixed.



19/08/2020 (12:30pm - 2:30pm).

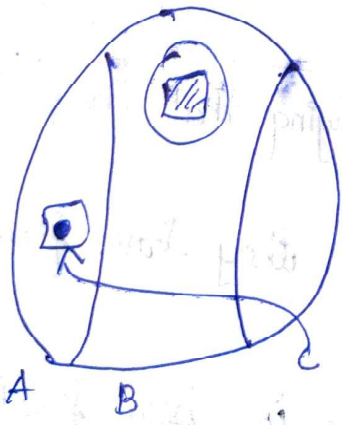
---> Owner of the bug

---> Owner of the feature.

---> The person who identifies a bug is called as Owner of the bug.

---> The person who tests the Complete feature is called as as Owner of the feature.

eg:



Owner of the Bug : C

Owner of the feature : B

WHAT HAPPENS IF A TEST ENGINEER DOESN'T SEND A DEFECT REPORT IMMEDIATELY???

---▶ You might forget the bug

---▶ If you the test Eng is not sending the bug report immediately then others might send your defect.

---▶ Development team might not get sufficient time to fix the bugs.

NOTE:-

1. We should retest the fixed bugs.
2. You should retest Only your bugs, don't worry about Others bugs.
3. As soon as you find a bug send it to development engineer immediately (so that dev eng will have sufficient time to fix the bugs)

MAILS

① What are the features there in the particular field. → (A, B)

② Build Number. → B01, B02.

③ path → D://Builds/B01

④ Name of the development Envi (Rex)

⑤ Bugs that are fixed —> 60

⑥ New features that have been added

↓

B

① Whenever new build comes we should test the new feature, first because probability of catching bugs is very high.

② We should do integration Testing between New and old module.

③ Retest the bugs.

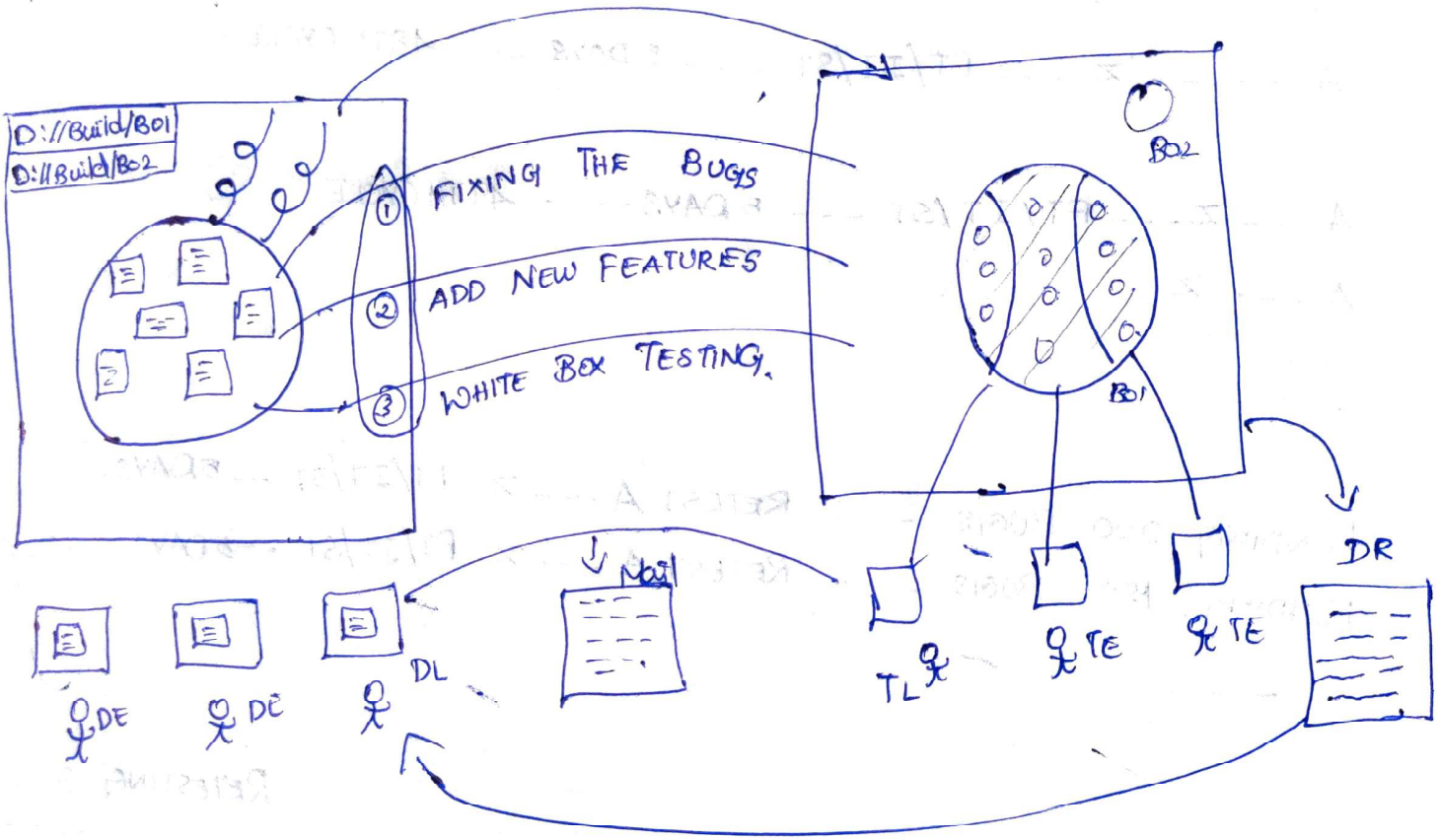
④ Test the old module.

WHY DO WE FIND NEW BUG IN OLD MODULE??

① Chances are there fixing one bug would have introduced one more bug.

② Chances are there adding new module may introduce new defects in the old module.

③ He would have missed few bugs in the Previous Cycle and He might be finding it now.



A ---- FT --- 5 DAYS ------ 1 CYCLE
A B ---- FT/IT --- 5 DAYS --- 2 CYCLE
A B C ---- FT/IT --- 5 DAYS --- 3 CYCLE
} FT/IT

A ---- P ---- FT/IT/ST ---- 5 DAYS ---- 31ST CYCLE
A ---- Q ---- FT/IT/ST ---- 5 DAYS ---- 32 CYCLE
A ---- R ---- FT/IT/ST ---- 5 DAYS ---- 33 CYCLE
} FT/IT/ST

A ---- Z --- FT/IT/ST --- 5 DAYS --- 48TH CYCLE.

A --- Z --- FT/IT/ST --- 5 DAYS --- 49TH CYCLE

A --- Z

PENDING 200 BUGS --- RETEST A --- Z FT/IT/ST --5DAYS-- 50
PENDING 180 BUGS --- RETEST A --- Z FT/IT/ST --5DAYS-- 51
}

RETESTING 3-8 CYCLES

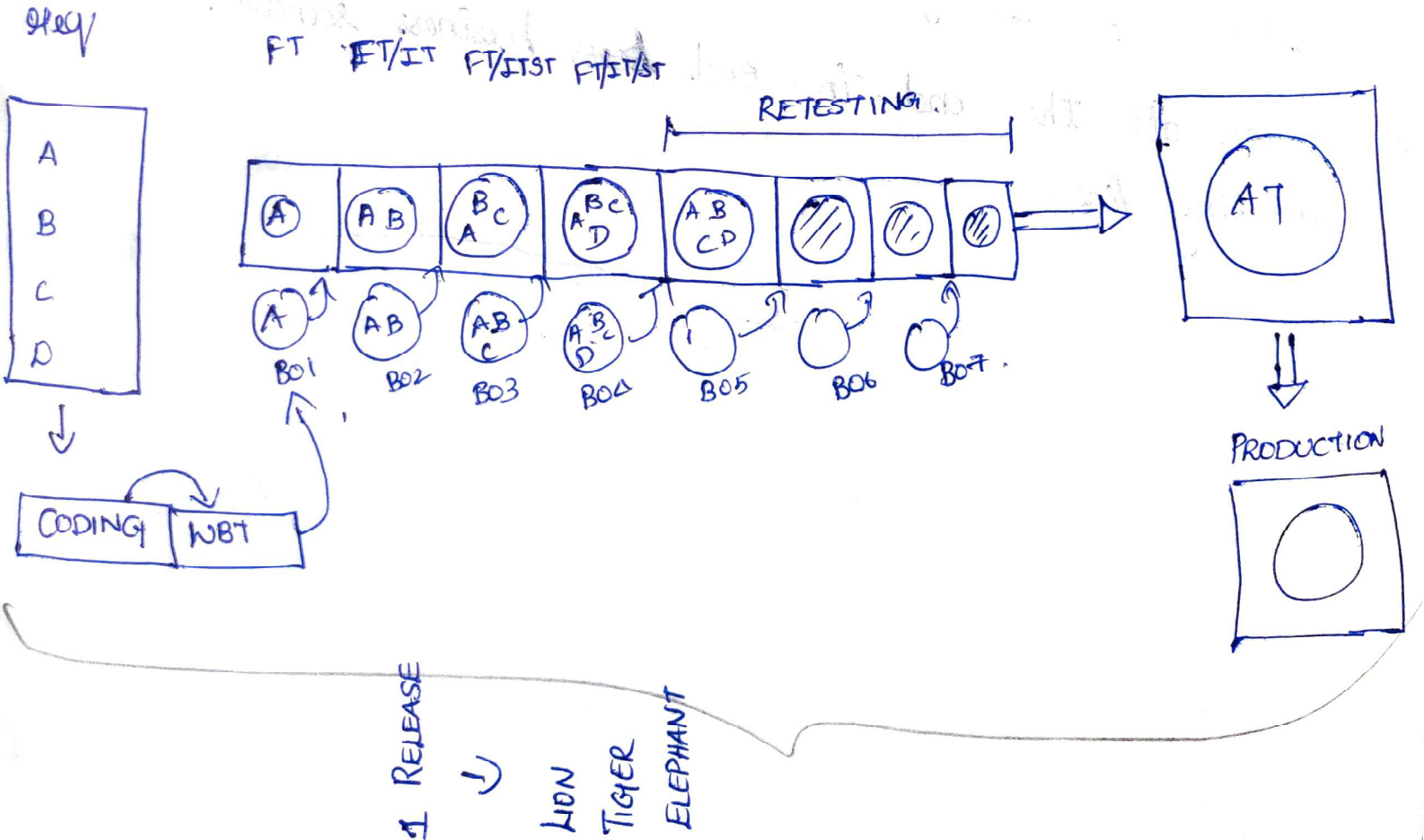
When Should We Do System Testing???

1. When all the basic features are stable.
2. When minimum bunch of features are ready.
3. If testing Environment is similar to the production Environment.

Note: Even though all the modules are ready we still continue with some more cycles because developers must fix all the pending bugs and the test Engineers should retest the fixed bugs.

20/08/2020 (12:30pm - 2:30pm)

Release:



Starting from Collecting the requirement, Developing & testing the software for many cycles and until we release the product to the production we call it as ONE RELEASE

WHEN DO WE RELEASE THE PRODUCT TO THE CUSTOMER???

1. If all the features requested by the Customer is ready.
2. If there is no critical or blocker bugs.
3. There are few bugs left out but all the bugs are minor and less than the limit set by the Customer.
4. If you have tested the product in the environment similar to the production environment
5. When all the end-to-end ~~bugs~~ business scenarios are working fine.

TEST CYCLE :.

* It is the duration or efforts spent to start & finish the complete testing.

* One test cycle might take 3 days, 5 days, 15 days

It just depends on.

1. Size of the application.
2. Complexity of the application.
3. Number of test Engineers in the product.

HOW MANY TEST CYCLES WHERE THERE IN UR RELEASE ???

while asking in Interview

1 TC → 5 days - 1 week.

1 month → 4 TC.

10 Month → 40 TC [±3] < 37 / 43

~~1 week~~ ?

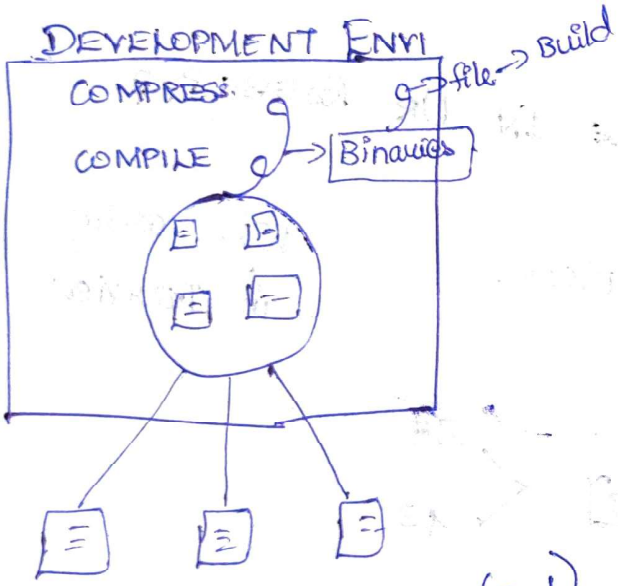
RESPIN::

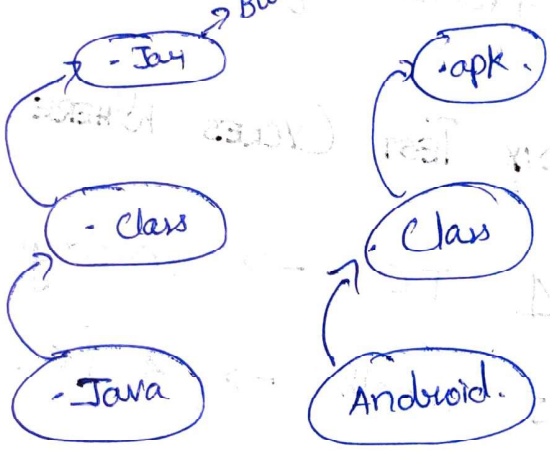
Getting more than One build in a single test cycle is called as Respin.

NOTE::

Avoid respin as most of the respin will cause more wastage of time in installation.

BUILD::





COMPILE :→ Binaries (0, 1).

```
IF
{
UNPWDD == 1
SOP("SUCCESS");
}
ELSE
{
SOP ("UNSUCCESS");
```

ASCII

A - 65
B - 66
C - 67
D - 68
E - 69
F - 70
G - 71
H - 72
I - 73

2 | 73
2 | 36 - 1
2 | 18 - 0
2 | 9 - 0
2 | 4 - 1
2 | 2 - 0
1 - 0

2 | 70
2 | 35 - 0
2 | 17 - 1
2 | 8 - 1
2 | 4 - 0
2 | 2 - 0
1

If ⇒ 1 0 0 1 0 0 1 , 0 1 1 0 0 1
I F

BUILD:-

All the programs are Compiled, then we get binaries, all the binaries are Compressed then we get Build.
(archive)

IN WHAT FORMAT YOU WILL GET THE BUILD ???

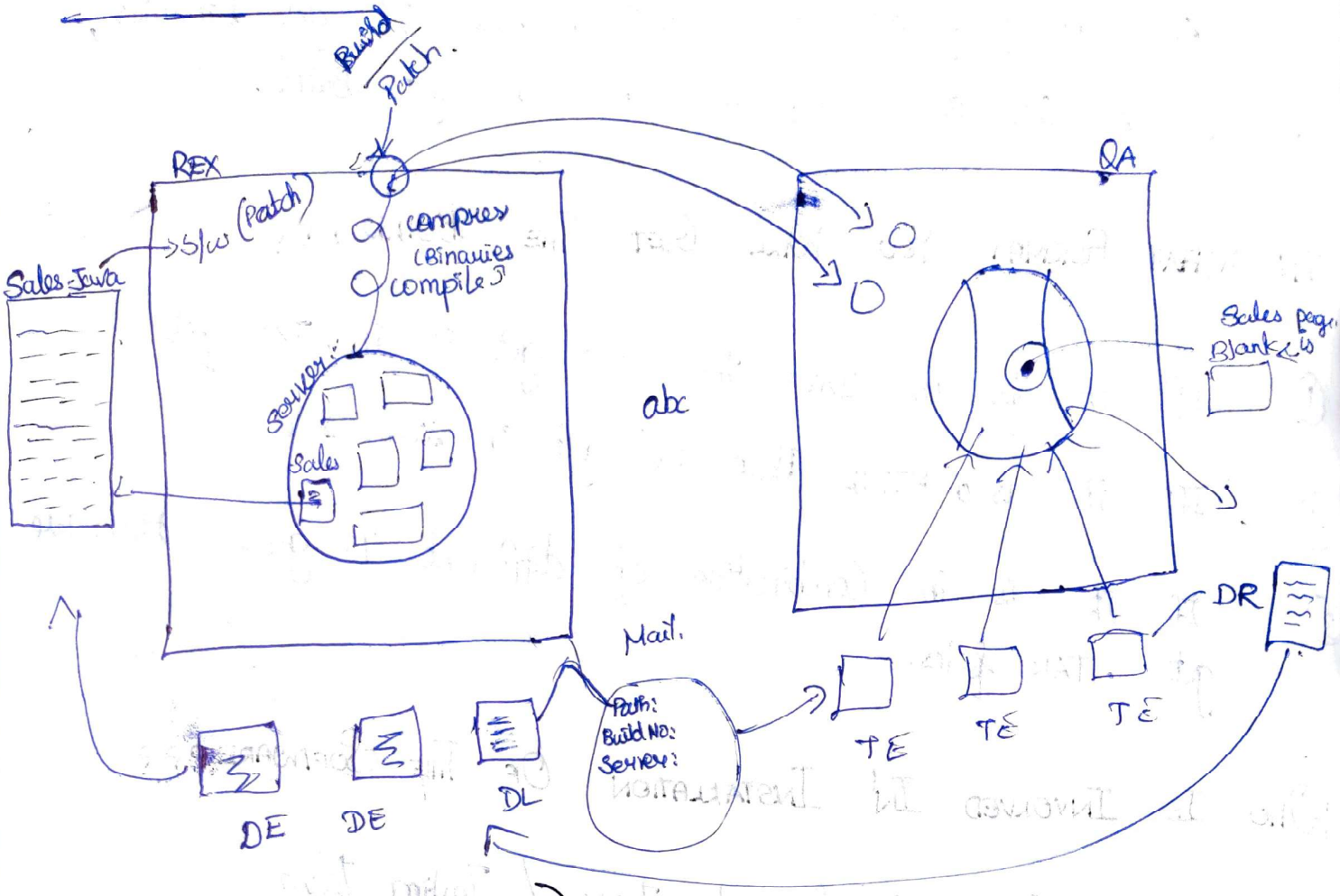
1. If it is in JAVA then we get it in .JAR files.
2. If it is in HTML then we get in .WAR files
3. If it is in Combination of different Programs then we get .TAR files.

WHO IS INVOLVED IN INSTALLATION OF THE SOFTWARE ???

* Anybody from Development team / Testing team
* Release Engineer or Build Engineer.

~~Anybody from~~

PATCH:



21/08/2020 (12:30pm - 2:40pm).

It is a small ~~Core~~ Software which Contains modified programs new program and a record of deleted programs.

Build	Patch
① Def	① Def.
② Build is a entire large Software	② Patch is a small S/w.
③ To install B02 (new Build) we need to uninstall B01 (Previously installed Build.	③ No need of Uninstallation of Previous Build in Patch.
④ By installing New Build we loose previous data	④ By installing the patch we will not loose our Previous data

Note:

Always in real project from 2nd release (Sprint) generally we give patch to production Environment. because by installing Patch we will not loose previous data.

Flow

Software Development Life Cycle:

* Tell What
* Tell different stages
* Tell different types of models
* Explain Each model in detail.

Software Testing:

* Tell What
* Tell Why
* Tell Types.
* Explain each type quickly.

White Box Testing:

* Tell What
* Tell Types
* Explain Each type quickly.

Black Box Testing::

* Tell What
* Tell Types
* Explain each type Quickly.

Functionality Testing::

* Tell What

Integration Testing::

* Tell What
* How do we do Integration testing
* Tell Types.
* ~~Each~~ Explain Each type quickly.

System Testing::

* Tell What
* Tell When
* Tell what is Build
* Tell what is Patch
* Tell what is ~~Pat~~ Test Cycle
* Tell what is Release
* Tell what is Continous Integration.
* Who is releasing the application.

ACCEPTANCE TESTING:

* Tell What (Approach - 1)
* Tell Why
* Tell different approaches.

SMOKE TESTING:

* Tell What
* Tell Why
* Tell When.

ADHOC TESTING:

* Tell What
* Tell Why
* Tell When

EXPLORATORY TESTING:

* Tell What
* Tell Why or When
* Tell drawbacks.
* Tell how to Overcome the drawbacks.

COMPATIBILITY TESTING:-

* Tell What
* Tell Why
* Tell When
* Tell different types of defect found in Compatibility Testing

GLOBALIZATION TESTING :-

* Tell What is I18N testing and what we check in I18N testing.
* Explain how we do I18N testing.
* Tell what kind of bugs we get in I18N testing
* Tell what is L10N testing and example of L10N testing.

USABILITY TESTING:-

* Tell What
* Tell how do we Use Usability testing
* Tell When
* Tell for which type of application we do Usability testing.

Accessibility Testing :-

* Tell What

Performance Testing :-

* Tell What
* Tell Types
* Explain how do we do performance testing.

Test Case :-

* Tell What
* Tell Why
* Tell When
* Explain different test Case design diagram techniques.
* Explain procedure to write test Case.

Regression Testing :-

* Tell What
* Tell types.
* Tell how to find impact area.
* Tell when we do full regression testing.

Test Life Cycle :-

* Tell What is the difference between Software test life cyce and software development life cycle.
* Explain Test life cycle in detail

Test Plan :-

* Tell What
* Explain Each Section of Test plan in detail.

Traceability Matrix :-

* Tell What
* Tell Advantage

DEFECT TRACKING::

* Tell what is defect
* Tell why we get defect.
* Tell difference between defect, bug, error, failure
* Explain defect life cycle in detail
* Tell what are different status and why we get those status
* Tell what is severity and levels of severity.
* Tell what is priority and levels of priority.
* Tell difference between severity and priority.
* Tell what is defect triage.
* Tell what is defect masking.
* Tell what is defect seeding.

ADDITIONAL QUESTIONS

SOFTWARE DEVELOPMENT LIFE CYCLE :-

1. What are the different types of models available in SDLC?
2. What are the advantages and disadvantages and application of each model.
3. Can you explain module in detail (waterfall, Spiral, V, Hybrid, Prototype & Agile model).

SOFTWARE TESTING :-

1. What are the characteristics of good test Engineer.
2. What are levels of testing.
3. Difference between Verification and Validation
4. Difference between static and Dynamic testing
5. Difference between Functional and Non functional testing.
6. Difference between QA & QC.

WHITE BOX TESTING :-

1. Why test Engineer is not involved in fixing the bugs?

FUNCTIONALITY TESTING

1. What is positive Functionality Testing
2. What is Negative Functionality Testing
3. What is Over Testing
4. What is Under Testing
5. What is Optimist Testing.

INTEGRATION TESTING:

1. Tell about Stub and Driver (or)
   When one module is ready and another module is not ready how you do integration testing.
2. How work allocation happen?

ACCEPTANCE TESTING

1. What are the different types of Acceptance testing.
2. What is Alpha and Beta testing.
3. What is UAT

SMOKE TESTING:

1. What is the difference between Smoke and Sanity Testing.

COMPATIBILITY TESTING:

1. How do you test web Application.
2. In real projects how do they choose platform to perform Compatibility Testing.

PERFORMANCE TESTING:

1. Tell for what kind of application we do performance testing.

TEST CASE:

1. How do you ensure your test coverage is good.
2. Difference between test Case and test scenario.

REGRESSION TESTING:

1. Drawbacks of Manual testing and Advantage of Automation
2. What is progression Testing.
3. Difference between Regression Testing and Retesting.

TEST PLAN:

1. Who writes test Plan.
2. Who reviews test Plan.
3. How long do you ~~know~~ need to prepare test plan.
4. what is release note and what is Contain
5. what are matrices?

TRACEABILITY MATRIX :-

1. What is forward traceability matrix
2. What is backward traceability matrix.
3. What is bi-directional traceability matrix.
4. Whenever requirement is not given how do you prepare traceability matrix.
5. What is the difference between test case & traceability matrix.

How you do dash-dash testing (-*-)

24/08/2022

Types Of White Box Testing:-

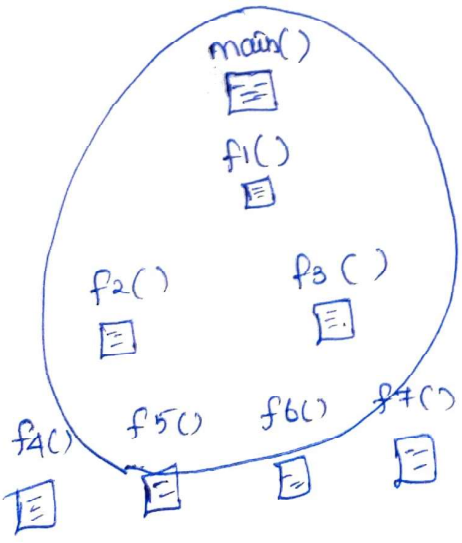
1. path testing
2. Condition testing
3. Loop testing
4. Whitebox testing memorypoint of View.
5. Whitebox testing performance point of view.

Path Testing:-

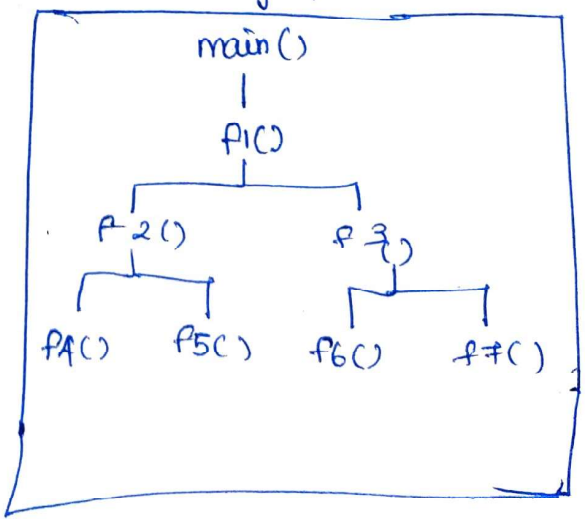
Write flow graph and execute independent paths

We Write flow graph:-

1. Which are all the programms not yet checked.
2. It prevent repeated testing of same code.



Flow graph



CONDITION TESTING:

Here we test the all the condition for both true and false value.

SSLC exam %.

more than 80% Science

less than 80% Commerce

if dinga 82

78 %.

if and else statement

If (greater than 80)

science

else

Commerce.

```
dinga = 82 /78
------
if (dinga > 80)
{----}
else
{----}
```

i agree to term & condition.

☑ ☐ not installed properly.

LOOP TESTING:

Test loop for all the cycle.

Write a program to print from 1 to 10.

```
a=1;
print (a)
print (a+1)
print (a+2)
:
print (a+9)
```

1
2
3
4
5
6
7
8
9
10

Loop:: If you are performing same actions again and again we go for loops.

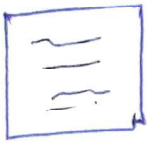
Types:: While loops, do while, for loops.

```
a=1;
While (a<11)
{ print (a)
 a=a+1
```

White Box Testing Memory Point Of View :-

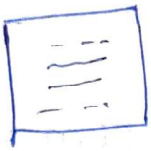
What are all the mistakes done by the developer because of which size of code increases unnecessarily.

① Because of not Using Proper Logic :-



Dinga

200 lines of code

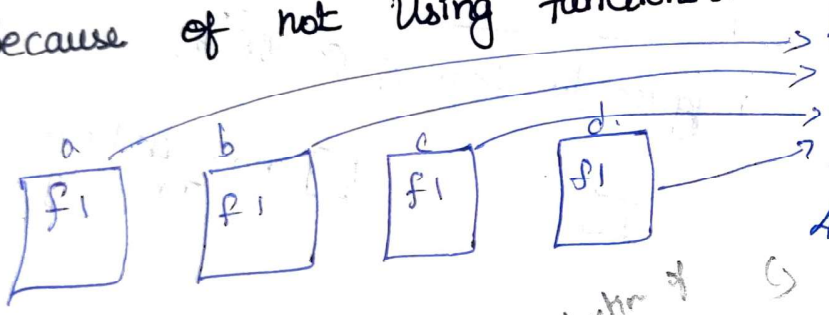


Manga

300 lines of Code

Instead of typing the full code we can use to loop concept, so that we can reduce the memory size.

② Because of not Using functions :-



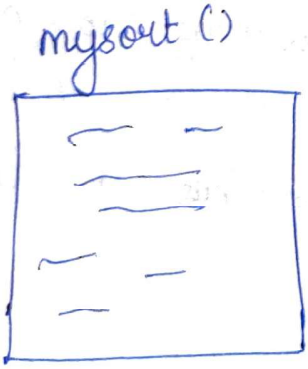
repeating

40 lines of code

10+4=14 line of code.

reduction of code.

③ Because of not Using in-built functions :-



→ 500 lines.

→ 1 Line.

Sort () --- inbuilt.

④ Because of Unused Variables and Unused functions...

```
a=10
b=20;
mysout ()
sout ()
...
...
{
print (a)
}
Called Sout {}
```

If there are repeated lines in a code make it seperately and file it as f1 so whenere you want to use it you call the f1 file.

WHITE BOX TESTING PERFORMANCE POINT OF VIEW:

What are all the mistakes done by the developer because of which the code / program is taking more time to run / execute.

① Because of improper logic.

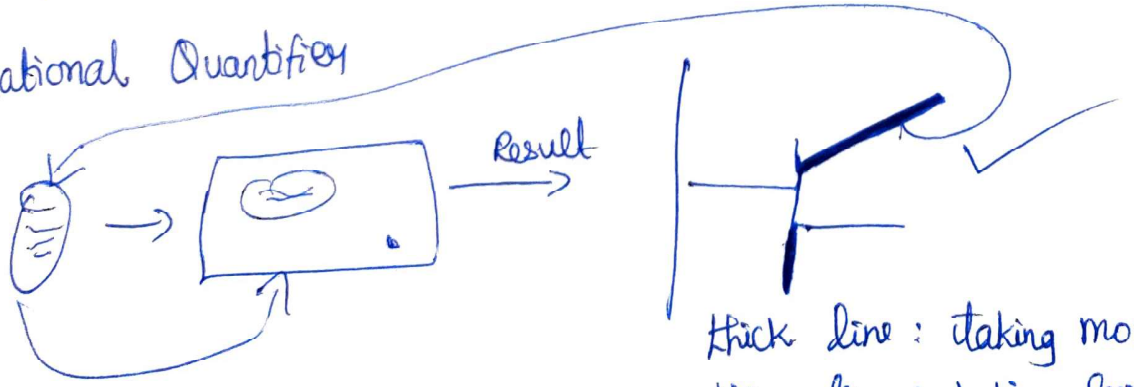
```
if (a OR b)
{}
else
{}
```

(a) If you have OR condition in the program make sure that 1st operand value should become true max number of time.

(b) If you have AND condition in the program make sure that 1st operand value should become false max number of time.

modifying a code / program with intent of increasing the speed of execution of code / reducing the time taken to execute the code is called performance tuning.

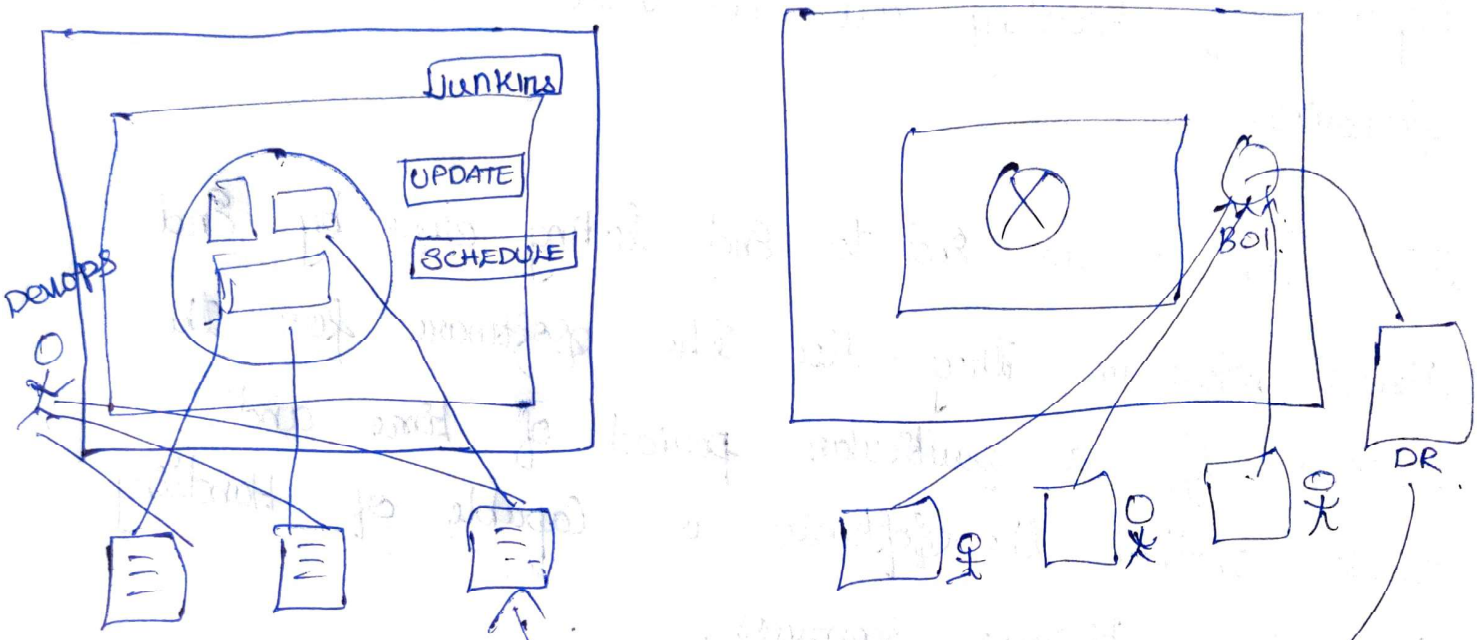
Rational Quantifier

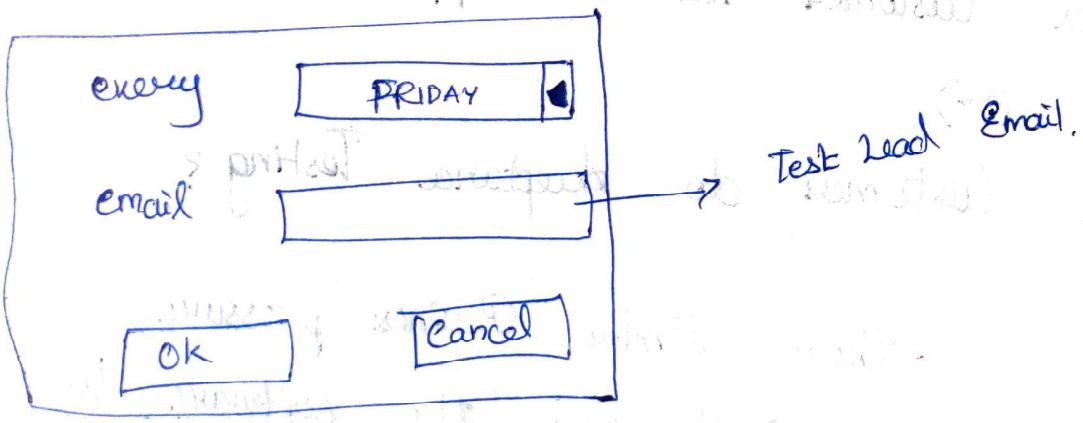


Thick line : taking more time
Thin line : taking less time.

CONTINUOUS INTEGRATION:

The Continous Synchronization between the Development team and testing team. is called. Continuous Integration.





ACCEPTANCE TESTING::

APPROACH ::

① It is an End to End testing, Testing is done by Test Engineer sitting at Customer place where in they check whether the software is capable of Handling all the Real time Business Scenarios.

② It is an End to End testing done by End Users where in they use the software for the Business for a particular period of time and check whether the software is capable of Handling Real time Business Scenarios.

Why Should Customer Test the application?

(Or)

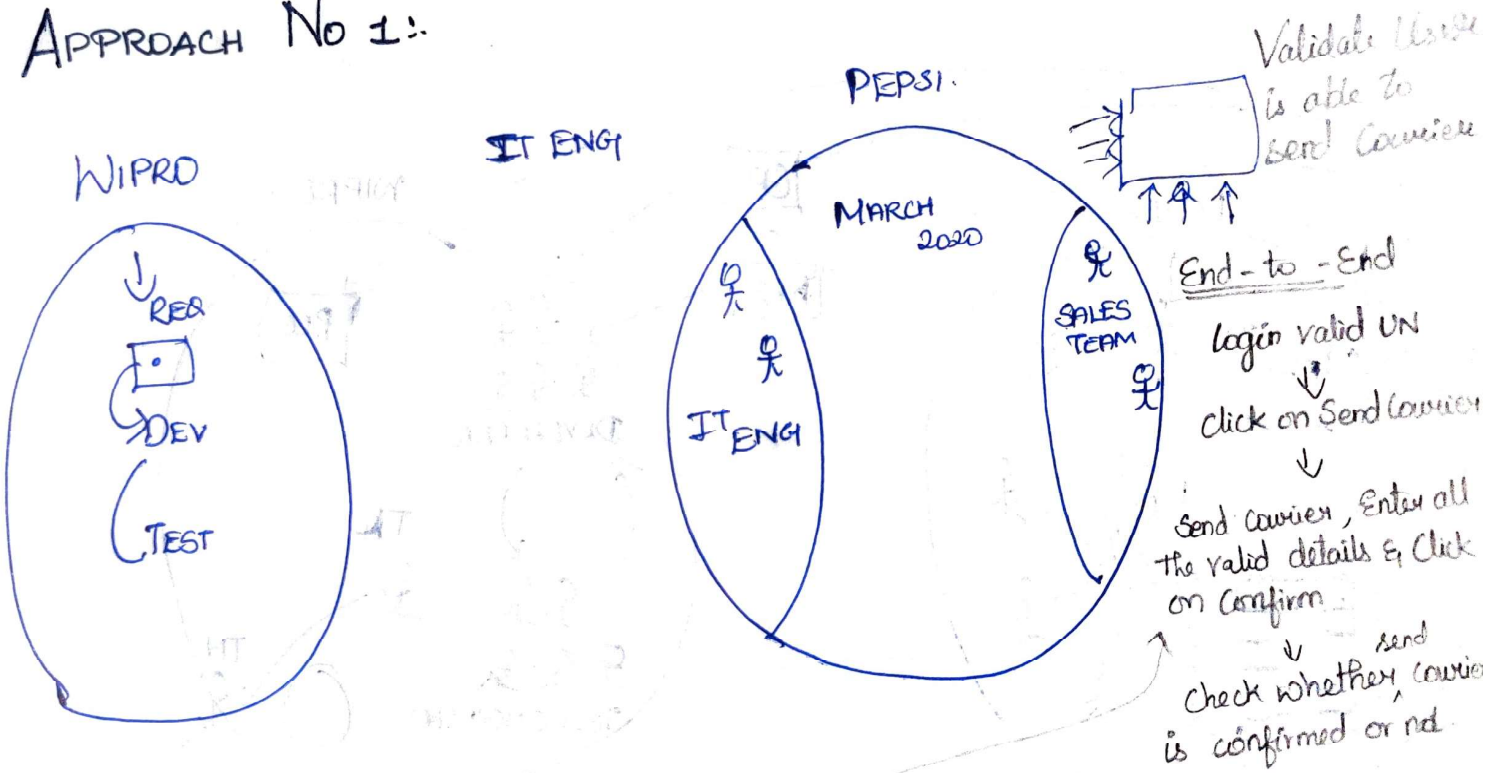
Why should Customer do Acceptance Testing?

① Chances are there Under Business pressure software Company might push the software to the Customer with lot of Bugs. To avoid this we do acceptance testing.

② Chances are there where ... misunderstand the requirement and develop lot of wrong features to avoid this we do acceptance testing.

③ If we launch the product with lot of critical bugs to the production then the customer will undergo severe loss.

APPROACH No 1:



DEF I:

It is an End-to-End testing done by IT Engineers sitting at Customers place and check whether the software is capable of Handling all the Real time Business Scenarios.

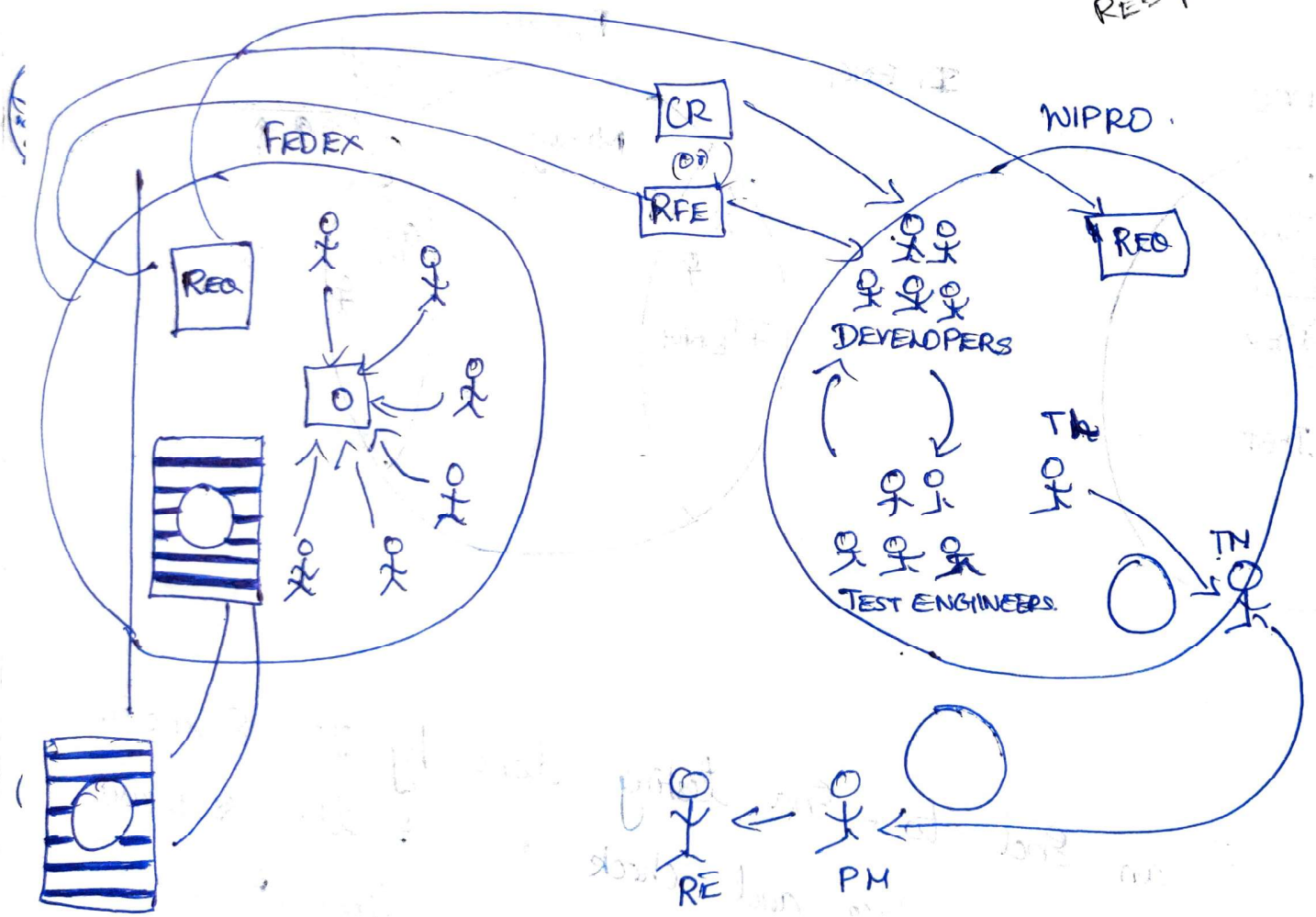
26/08/2020 (12:30pm - 2:40pm)

UAT - User Acceptance testing.

APPROACH NO : 2

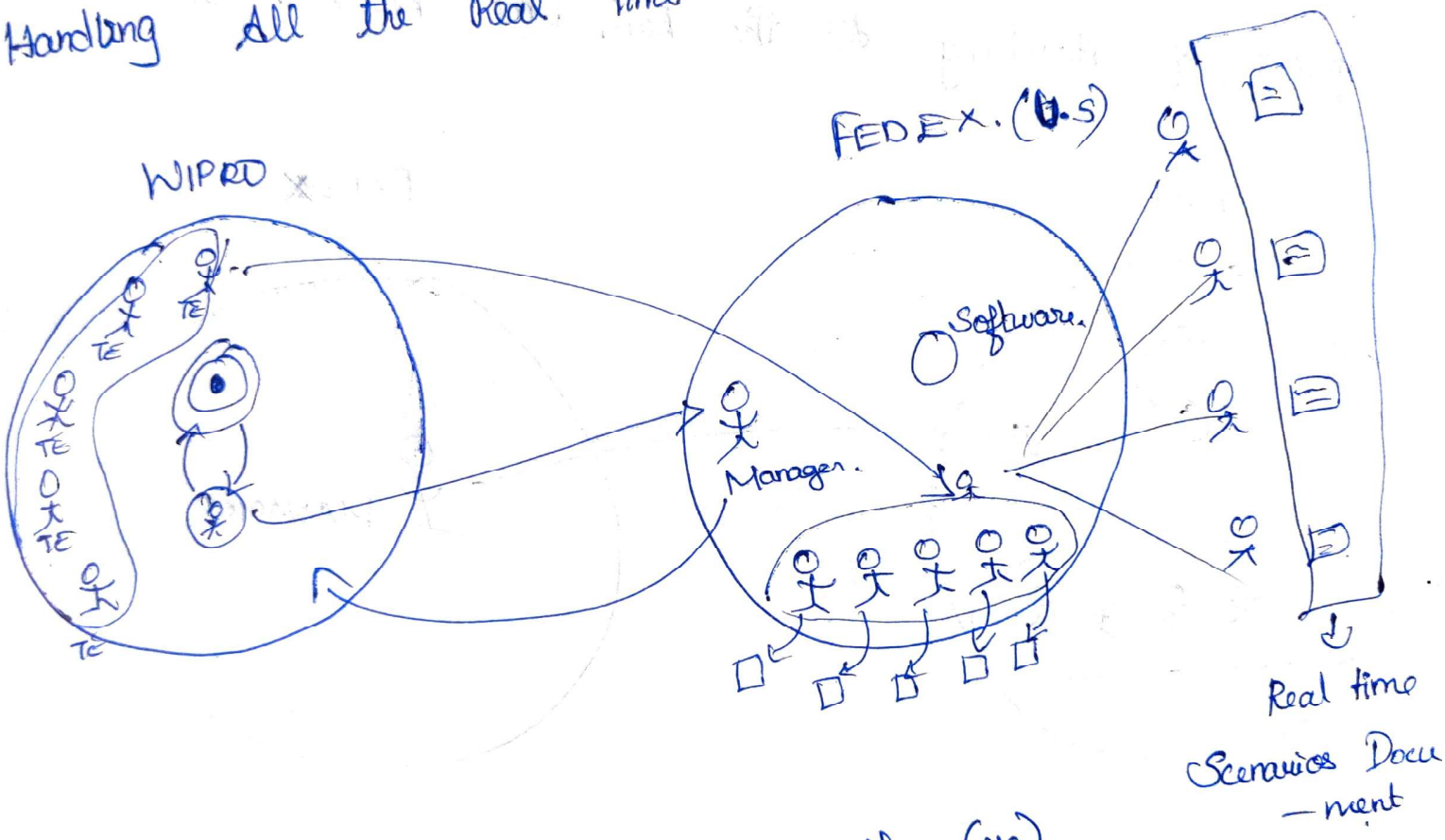
It is an end to end testing done by end users wherein they use the software in the business for a particular period of time and check whether the software is capable of handling all the real time business scenarios.

RFE = Request A Enhancem
CR = change Request
RE = Release Engg



Approach No : 3

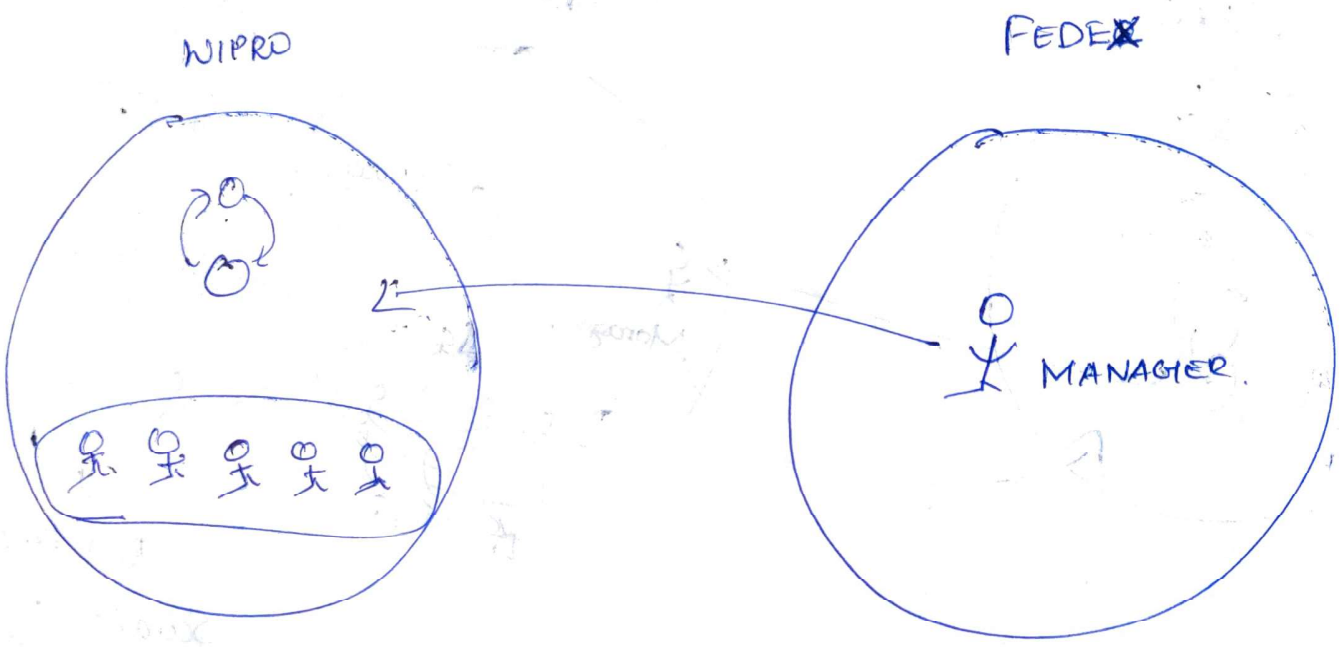
It is an end to end testing done by our own test Engineers sitting at Customers place wherein they refer the Real time business scenarios given by the Customer and check whether the Software is capable of Handling All the Real Time Business Scenarios



Five TE's will go to the fedex office (US) to test and The TE in fedex company of (wipro) will test with the Real time Scenarios Document. If

APPROACH NO :- 4

It is an End to End testing done by our own test Engineers sitting at our own place wherein they refer the real time business scenarios ~~by~~ given to by the customers and check whether the software is capable of Handling all the Real time business scenarios.



ASSEMBLY DONE FOR AGILE METHODOLOGY