Manual Testing

classmate
Date 19/11/2020
Page

* **Software** - Set of programs given as an instruction to any machine which will respond to user action.

* **Software Testing** -

Defination 1 - Verifing or cheaking application, Features Functionality is working as expected or not.

As a Tester we need to verify every option usage in software.

Def^n 2 - Testing a software with an intention to identify Bugs

* **Bug** :

Def^n ① → Deviation bet^n expected and actual result.

Def^n ② → Feature Functionality not working as expected.

* **Error** -: Mistake in a Program.

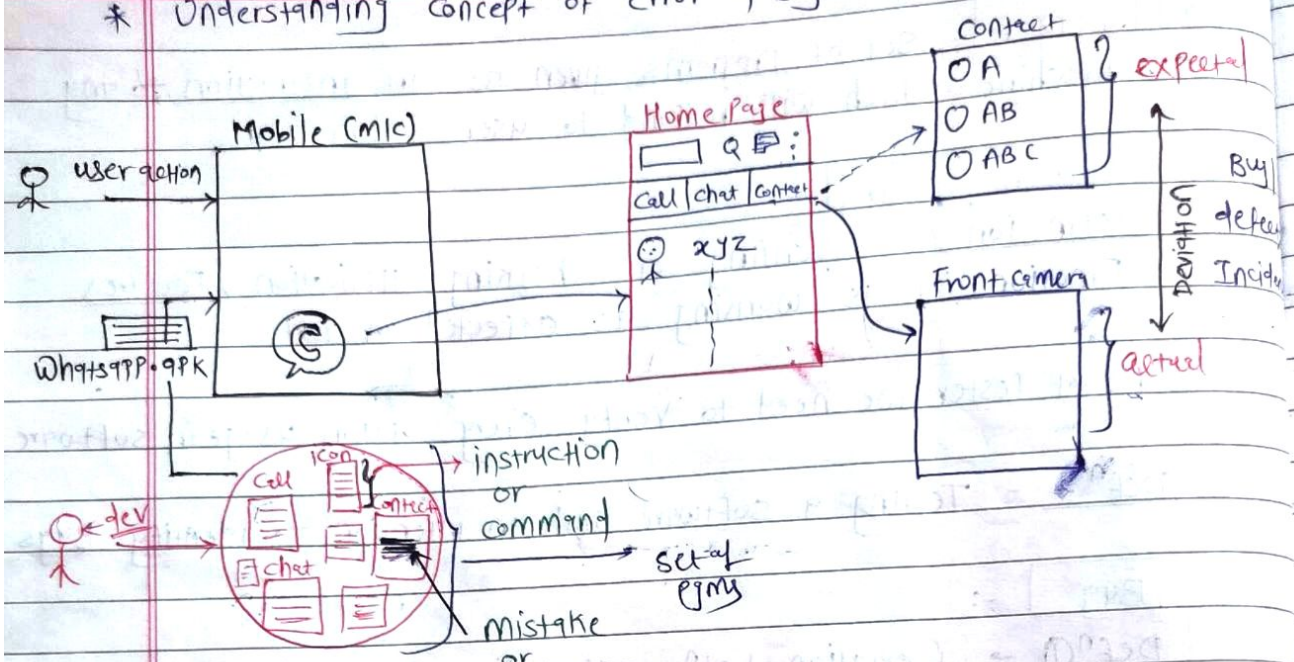
* Types of error -: 1) Compile time error (syntatical mistake)
2) runtime error (Program executable mistake)

Note - If developer do mistake in programs tester will identify the bugs while testing.

S.T Defination 3 -: Testing an appl^n by executing Programs and identifing mistakes

2021/10/21 19:25

* Understanding concept of error & Bugs →

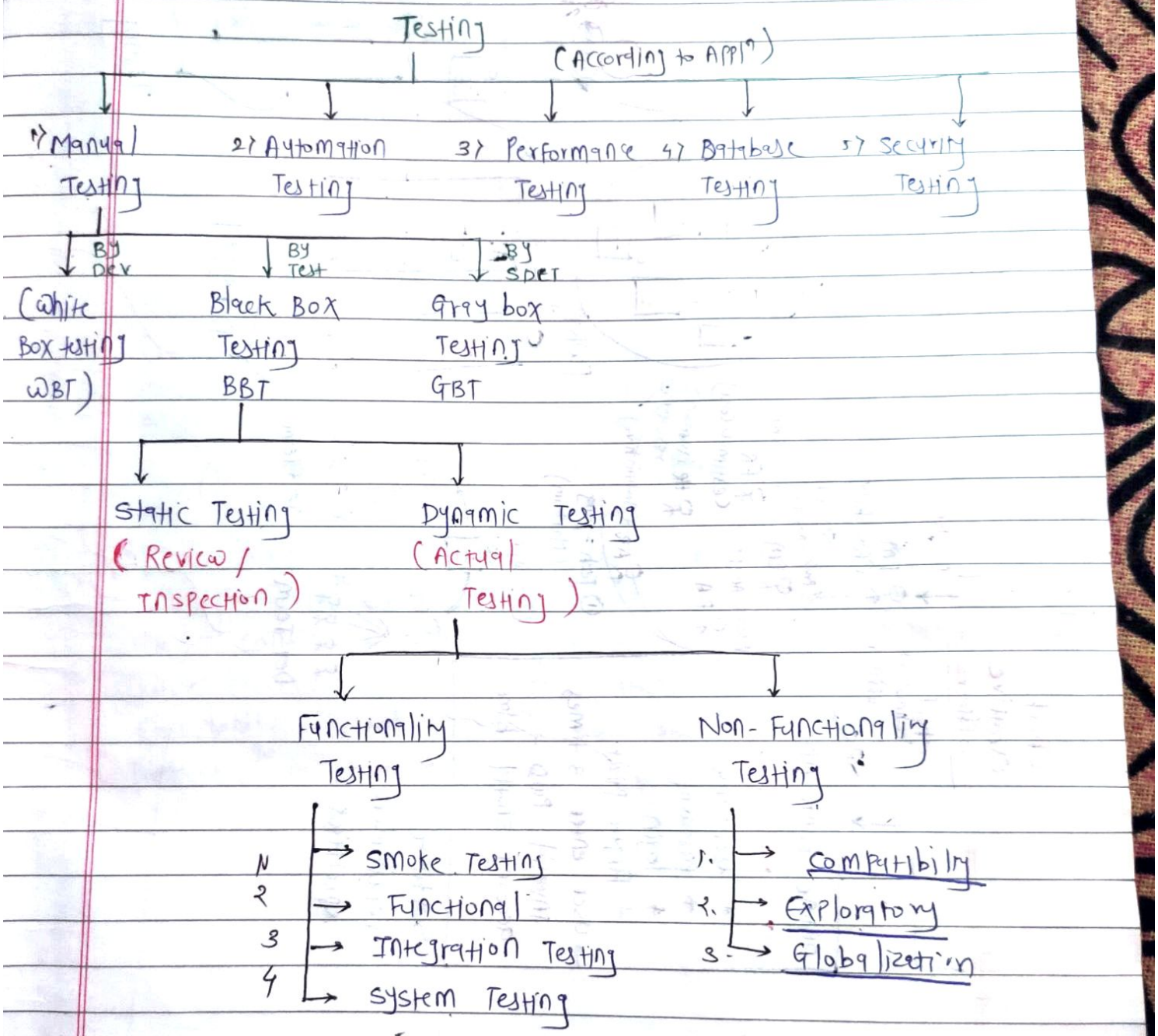


ST Defn 4 : Testing an appln with an intention to improve its quality by identifing more number of Bugs.

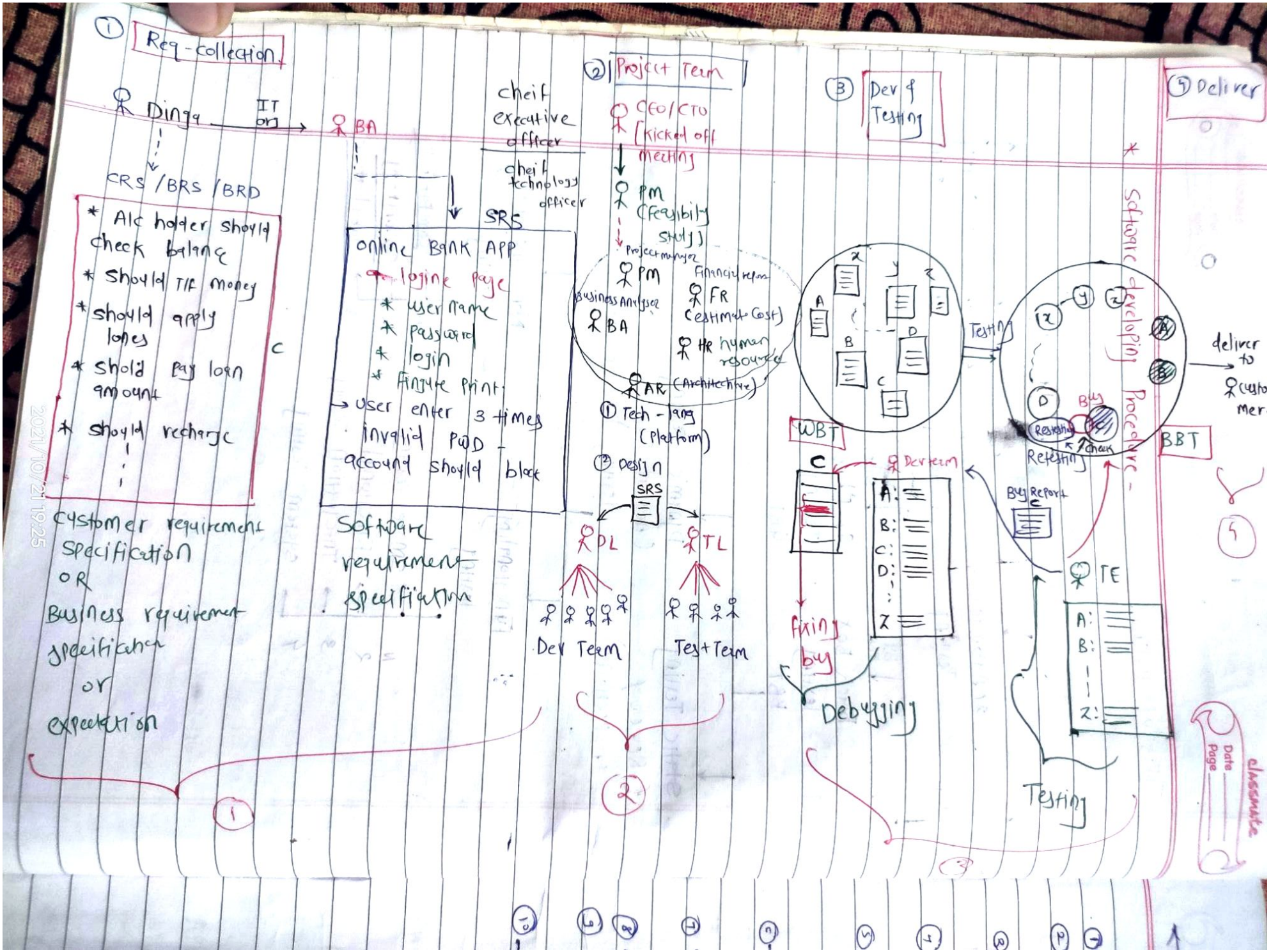
→ Manual Testing
By Dev
(White Box testing WBT)

7-2-20

Types of testing.



2021/10/21 19:25


① Req-collection
Dinga
IT or
BA
CRS / BRS / BRD
* A/c holder should check balance
* should TF money
* should apply loney
* shold pay loan amount
* should recharge
customer requirement specification
OR
Business requirement specification
or
expectation
chief executive officer
chief technology officer
SRS
online BANK APP
login page
* user name
* password
* login
* finger print
→ user enter 3 times invalid PWD - account should block
Software requirement specification
② Project Team
CEO/CTO
[kicked off meeting
PM
(feasibility study)
Project manager
PM
Business Analyser
BA
Financial rep.
FR
(estimate cost)
HR human resource
AR (Architechture)
① Tech - lang (Platform)
② Design
SRS
DL
TL
Dev Team
Test Team
③ Dev & Testing
A
B
C
D
Testing
WBT
C
Dev team
A:
B:
C:
D:
Z
fixing bug
Debugging
* Software developing Procedure -
deliver to customer
BBT
Bug
Retesting
check
Bug Report
TE
A:
B:
Z:
Testing
④ Deliver
classmate
Date
Page
2021/10/21 19:25
4

← Explaination of given digram

① - Customer will Prepared requirment and send it to Project team
② - BA (Business Analyst) will converts CRS to SRS Document
③ - once project manager will selected feasibility will be conducted to analyse customer requirement.
④ - Business A will share the requirement with both developing and testing teams
⑤ - By refering to requirement document developers develops the application and perform the white box testing (WBT) before send to testing team
⑥ - Testing team by refering to requirement Performs Black Box testing (BBT)
⑦ - If identifies the any bug prepared bug report and send it to the developers.
⑧ - By refferiny to bug report Developer will fixed the bug.
⑨ - Testing team will retest developer fixed bug and continoey the testing.
⑩ - This process will continey until on feature are completely tested, appln is bug Free and relised Product to the customer.

8 / 20

Interview Que

1) diff betⁿ CRS & SRS.

	CRS	SRS
1)	It is prepared by customer	1) It prepared by buginess analysy
2)	It explain customer needs or specification	2) It explains software specification how applⁿ should be developed
3)	To prepared CRS customer buyiness is a base	3) To prepared SRS, CRS is a base document.

2) * Responcibilities of BA.
→ 1) Interacting with customer
2) converting CRS into SRS
3) explaining customer needs to software team

3) what is kick of meeting ?
→ initial meeting conducted to developed a software

4) what is fixing of bug ?
→ Modifing Program or removing wrong code and entering correct code

5) diff. betⁿ debbuging & testing
1) debugging - 2) Testing
① It performed by developer. ① Performed by tester.
② Identifing mistake & fixing the mistake ② identifing the mistake.

2021/10/21 19:25

① build dass (BC)

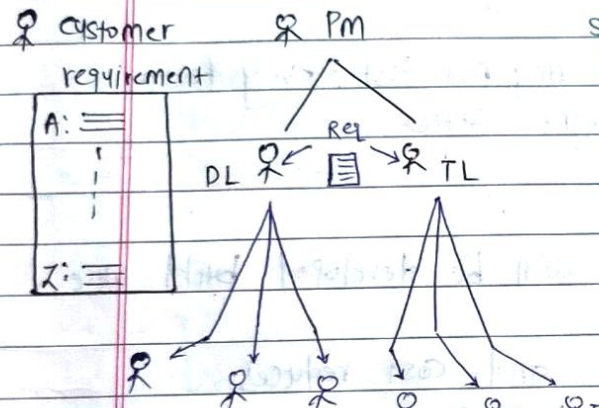
dev lead
Test lead

classmate
Date
Page

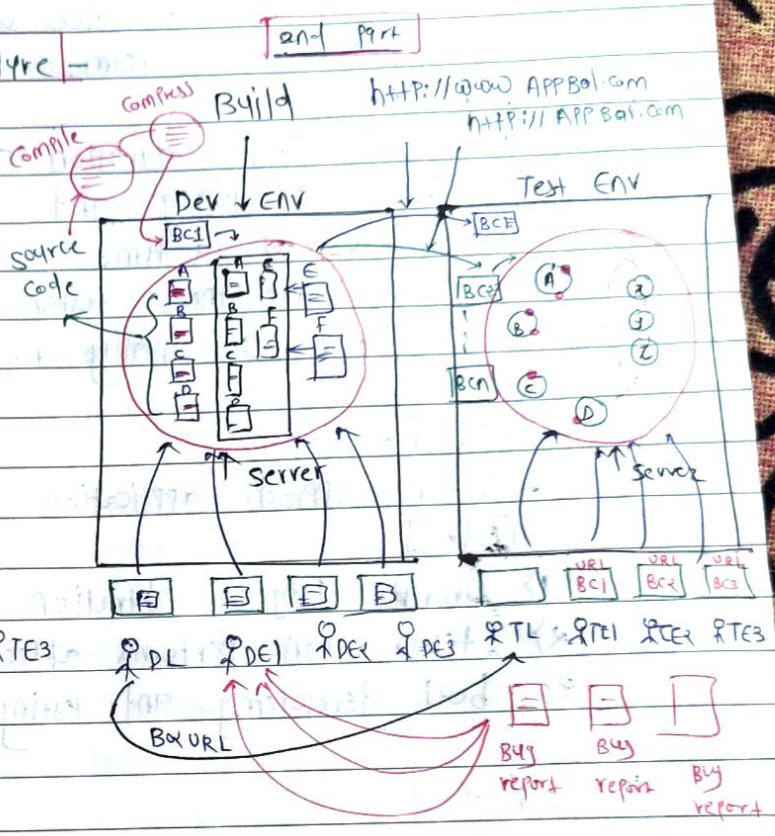
c) What is Retesting
- It is perform after fixing bug.
- Retesting means verifing or checking bug is really fixed or not.

* **Software testing Procedure** –

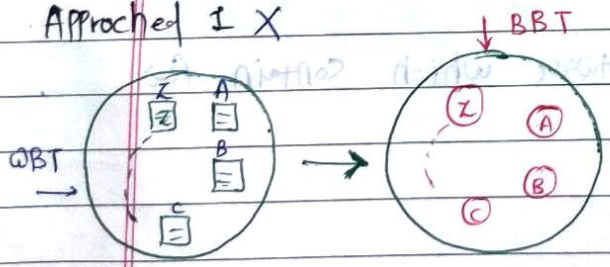
1st Part



2nd part



Approched 1 X

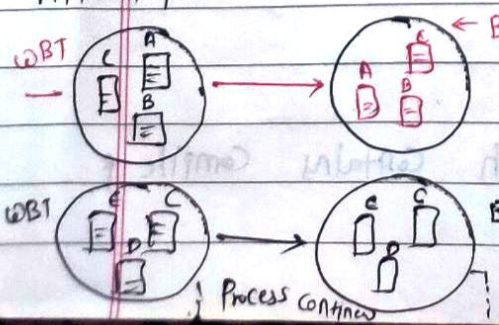


Compress

Compression X
1) multiple file to single
2) size reduced (↓)
3) extention (.zip)

Archened ✓
1) my multiple file to single
2) size remain same
3) exten on

Approched 2 ✓



2021/10/21 19:25

1) To developed an application developing team will used Following Approches.

* Approch 1
- in this Approch developers will developed complite appln and send to testing team.

Limitations
1) overall Project duration and cost might increased
2) both developing and testing team are not involved at a time.
3) chance are there testers may not test every feature complitely, so quality might affect

* - Approch 2
- In this approch application will be developed build wise.

Adv
1) Overall Project duration and cost reduces
2) tester can perform effective testing
3) both developeing and testing team are involved.

* Build

Def. ①
- It is the piece of software which contain few Features

Def ②
- Its a piece of software which converts Compile & Compress Program to features

Def ③
- Its a piece of software which contains Compile & Compress Programmy

2021/10/21 19:25

① once build 1 is developed developers will installed build in testing environment.

② while testing team build-1 developed, Features tested

③ development team develop build-2 which contain new features and bug Fixes.

④ after testing build 1 complitely new build will be send to testing team.

⑤ this process will continued uptil all Features are complitely developed. and relised product to customer.

I-Q when will you perform of retesting on fixed bug if you identified bug in current build

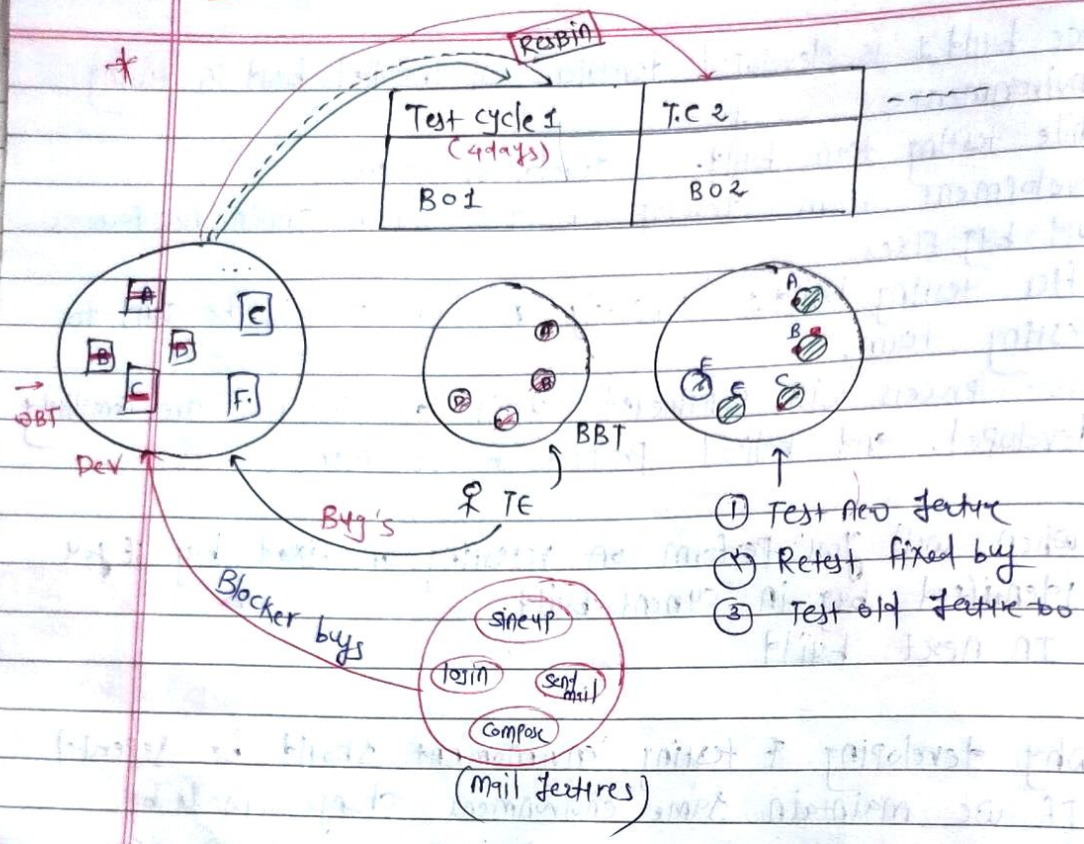
→ In next build

I-Q why developing & testing envirnomant should be seperated

→ - If we maintain same environment changes made by developers will impact on tsting
- tester will not able to perform effective testing

*

Test cycle



① — When a build come for testing test lead will decide Test cycle duration

② — Whithine the test cycle we should test all developed feature.

③ — developer will fixed bugs in 'Next build'

④ — For 1 test cycle testing team may receive only 1 build.

⑤ — chaces are there Testers may identified Blocker bug.

⑥ — developer should fixed immediatly and send Resbin in same test cycle.

* **Test cycle** - Time taken by testing team to test a build

Test cyde duration will be decided based on
- 1) No de of feature developed
- 2) No of tester available
- 3) & Complexity of features.

2) Blocker bug - It is a type of bug which will not allow further testing

3) Regbin - More than one build in same test cycle
- No of regbin depends on number of blocker bugs identified by tester.

Mayuri Mahendra Kawde

10-2-2020

* **Black Box testing**:-

① Verifing or checking application, Features, Functionality by refering to requirement document.

② Black Box testing also known as closed box testing because code is not visible to ~~developers~~. tester

③ in BBT we test each feature functionality so it is known as behaviour testing

④ Tester is a First person to used software.

⑤ To do black box testing testing team will used Following testing like Functional, integration etc.

* **Functional Testing**

Testing each and every component, Functionality Complitely or thoroughly by reffering to requirement

Functional requirement document Forsignup Page-

① When user acts as application by entering URL

1.1) Welcome Page should to display

1.2) It should contain Following Features

a) signup

b) login

② Wher user clicks on signup button

2.1) signup Page should display

2.2) It should contain Following Features

2.2.1) user name

a) It should accepts 8-12 charecter

b) It should be combination of Alpha + number

c) Special character, operator's, blank & space not allowed.

2.2.2) password

a) It should accepts 6-10 charecters

b) It should contain alpha + no.

ex→ URL given

2021/10/21 19:25

c) Password should starts & end with alphabate.
a) one special char allow (Optional))
e) blank space not allow.

2.2.3) email id
should accept valid email id.

2.2.4) Contact
a) should accept 10 digit no, b) alphabets, decimal, expression spe.char, blank space not allowed

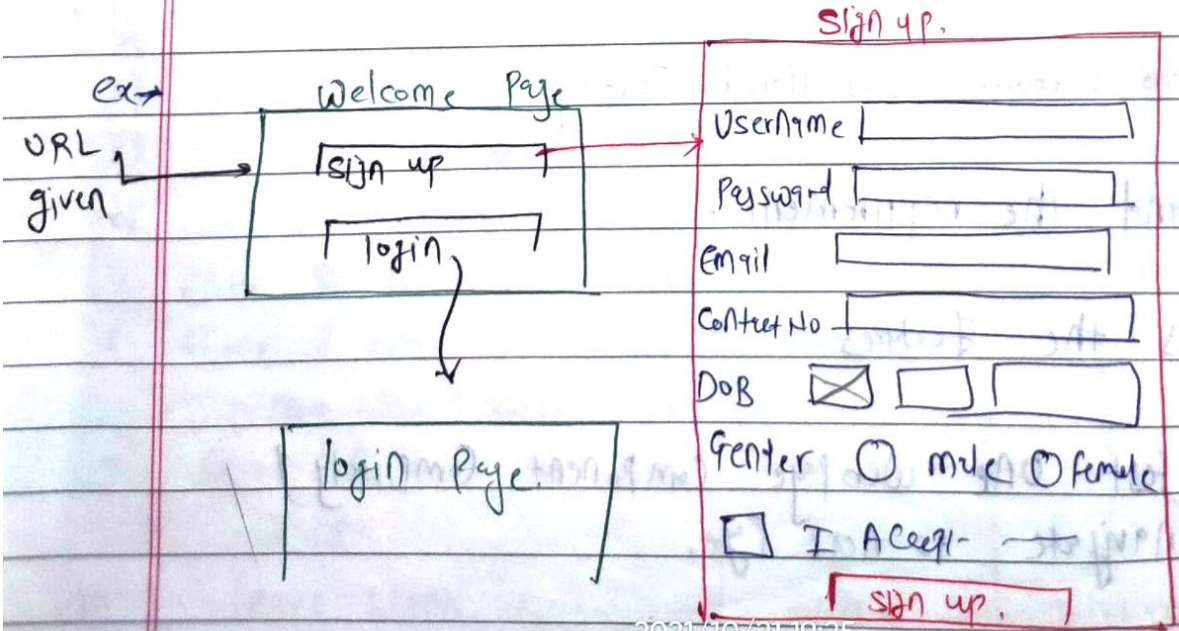
2.2.5) DOB
- should accept valid DOB.

2.2.6) gender
- should accept any one option from list.
a) male
b) Female

2.2.7) signup button
- when user click on signup
a) Home Page should display

(3) Login - when user click on login
3.1 - login page should display

* By refering to above requirement document developers dereloped application as follows.



→ enter text → I/P → Valid / Invalid } Req

① Test field
② list box →  (single i/p selected)
③ links, checkbox, Radio, button → used to go next page.

☑ Check 1 ○ R1
☐ 2 ○ R2
☐ check n ○ R3

multiple i/p selected — single i/p selected.

* In Functional testing we need to test each component by using all possible inputs. (Valids & Invalids)

* Procedure to Perform Functional Testing.

Step1) Identified all possible input.
Step2) Test Component with identified i/p.
Step3) If any i/p not working as exeepcted, Prepared bug report

* Rules to Perform Functional Testing

Rule1) Understand the requirement.
Rule 2) Priorities the features
Rule 3) always test one webpage component Complitely then navigate to next Pge.

Rule 4)
Rule 5)
Rule 6)
Exampl
Sr N
Valid
Invalid

2021/10/21 19:25

Rule 4) always test one component at a time.

Rule 5) always test valid input first.

Rule 6) If all valid are working then test for invalids

Example

Username [ ]

Requirement → a) it should be 8-12 character.
b) should accept alpha + no.
c) spl char, operator, blanks space not allowed

For checking this we make 1 chart as below.

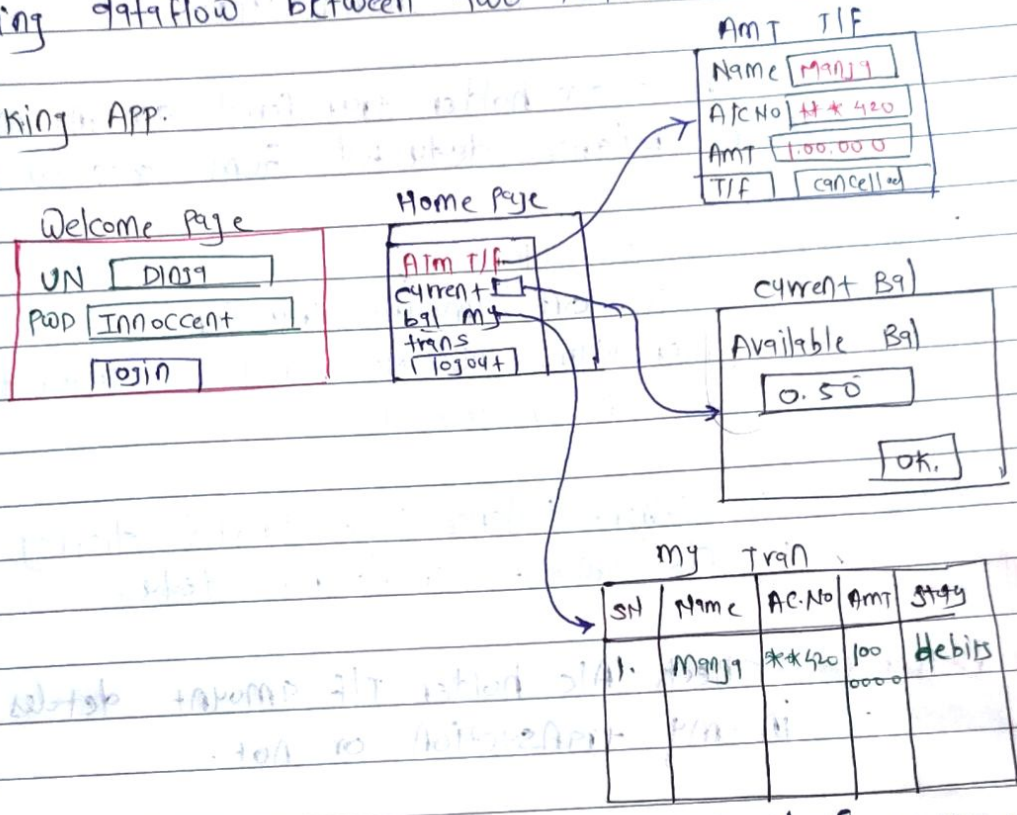
Sr No	Action.	I/p	Expected result	Actual result	Status
Valid cases	1) Enter 8 char UN with alpha + No.	Dinga112	should accepted	Not Accepted	Fail CBug
	2) Enter 12 char UN with alpha + No	Dinga1234567	should accepted	Not Accepted	Fail CBug
	3) Enter 10 char UN with alpha + No.	Dinga12345	Should accept	Accepted	Pass
	4) Enter <8 UN with alpha+No	Dinga1	Not accepted		
	5) Enter >12 UN with alpha+No	Dinga12345678	Not accepted		
	6) Enter 8 char UN with alpha	abcdefgh	Not accepted		
	7. Enter 8 char UN with No	12345678	Not accept		
	8. Enter 8 char UN with alpha + No + special char.	Dinga@R3			
	9. Enter 8 char UN with Space	------			
	10. leave blank.	Nothing	Not accepted		

Integration Testing

Def^n 1) Testing relation between two Features

Def^n 2) Testing dataflow between two Features based on relation

eg. Banking App.



1) Action1- check A/c holder T/F AMT deducted from current balance

- Steps - 1) login to app
  2) click on Amt T/F
  3) Enter Name, A/c No, Amt, and click on T/F button
  4) click on current balance

- Expected result - a) current Bal. Page should display
  b) T/F amt should deduct.

2) Action-2 - check Acc. holder T/F amt details displayed in my tran or not.

- Steps 1) login to app
  2) click on amt transfer
  3) click on my transaction.

exp. result. a) current bal page should display
b) Tlf Amt should deduct.
a) my transaction page should display
b) Tlf amount should update.

3) Action3 - check acc. holder transfered amount more than current balance. deducted from acc or not

steps. 1) login to app
2) click on amount transfer
3) enter Name Alc No. amt is greter than available
4) click on available balance.

ex result - a) current Balance page should display
b) Tlf amount should not deduct.

4) Action4 - check Alc holder Tlf amount deteles displayed in my transaction or not.

steps. 1) login to app
2) check on amt Tlf
3) Enter name/ Alc No/ Amt more than available balance
4) click on Tlf
5) click on my transcation.

ex result - a) My transcaction page should display
b) details should not updated

5) Actions - check account holder cancelled transcation amt deducted from acc or not.

Steps - 1) login to app
2) click on amount transfer
3) Enter name, A/c No Amount & click on cancled button
4) click on current balance.

Exp. result - a) current page should display
b) Amt should not deduct.

6) Actions - check Acc. holder cancelled transaction details updated in my transcation or not.

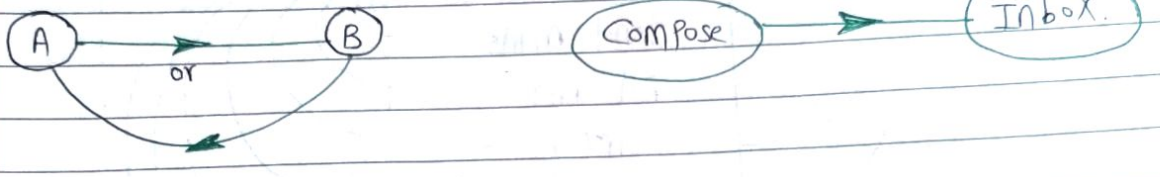
Steps - ① login to app
2) click on AMt Tlf
3) Enter name A/c No, Amt & click on cancelled button
4) click on my tran.

Exp. result - a) my tran. page should display
b) details should not update my transcation

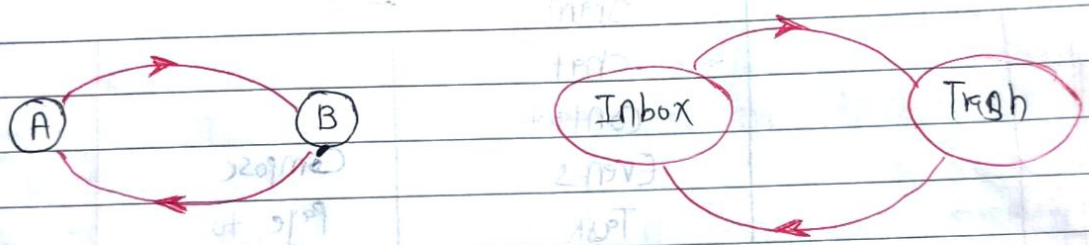
* Integration testing on whatsapp -

* Relation Types -

1) Monodirectional -
Relation betn two Features will be in one direction, either From A to B or B to A.



2) Bidirectional -
Relation betn Fatures will be in both direction i.e A to B and B to A also.

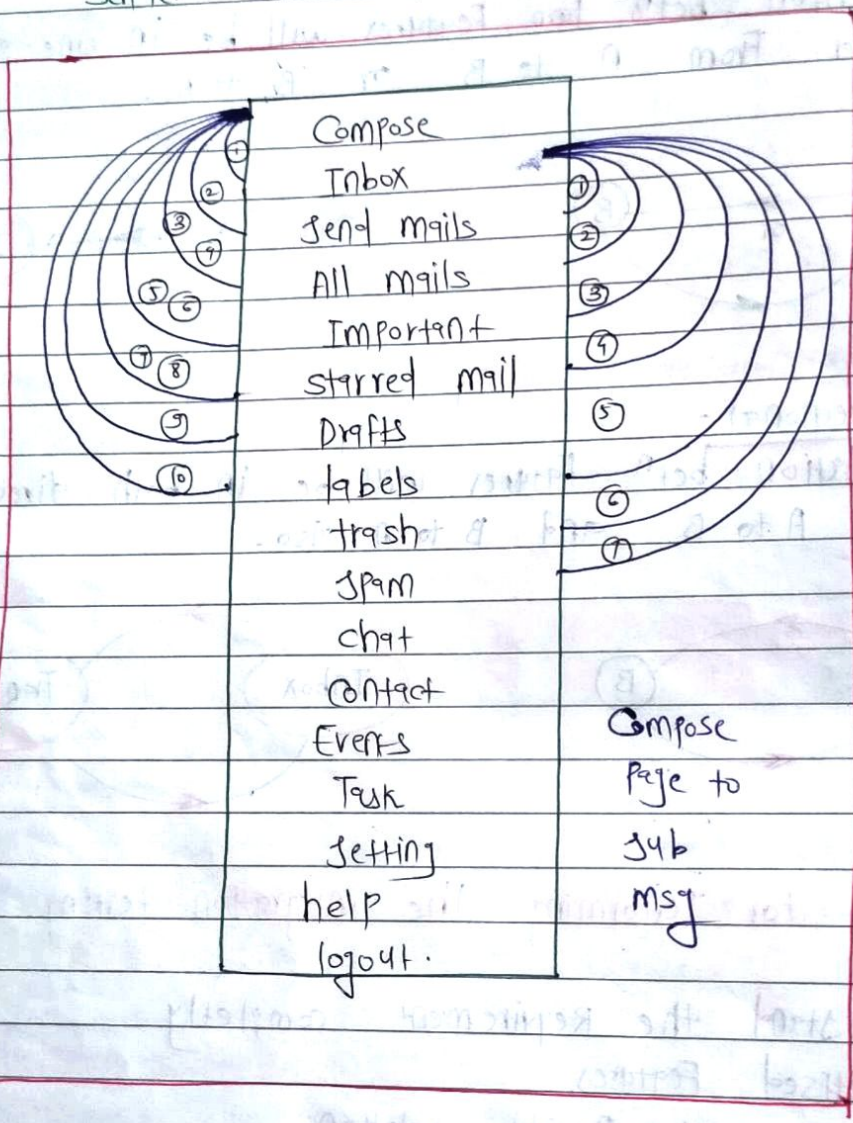


* Rules For performing the integration testing

1) Understand the Requirement completly
2) Prioritised Features
3) Identify all possible relations
4) Always test valid relation first.

* Integration testing on Gmail

Sender Gmail / Home page.



* Integration Testing between Compose with other features

1) It between Compose & Inbox → check if we send any email From Compose it will display in inbox or not.

2) In between Composed and send mails - check all the sent mails From Composed it will display in send mails or not.

3) It between Composed and all mails :-
- Check if send emails is display in all mails or not
- If we cancelled the email it will also display in all mails or not.

4) In between Composed and important -
- check whether all send emails are display in important or not.
- If we cancelled the send mail it will display or not.

5) it between Compose and starred Potat mail -
- If we make an email as starred it will display or not.
- If we cancelled the send mails it will display or not.

6) It between Composed and starred mail -
- If we make any emails as starred it will display or not
- If we cancelled the starred mail it will display or not in starred Mail.

2021/10/21 19:25

7) It between Composed between Draft.
If we Cancelled any sent emails check whether emails display in draft or or.

8) It between Composed and labels
If we send any mails with label it will display in label or not.

20-2-2019

Integration testing is classified into two types based on how tester will combine features to perform integration testing.

1) Non-Incremental Integration Testing
Testing relation by Combining features which are having one set of relation by performing one action check all the relations.

Limitation:
1) To perform non-incremental Integration tester should have good knowledge on appln.
2) It consumed more time
3) Tester needs to spends more efforts.
4) chances are there might miss one or two features grouping which might contains bugs.

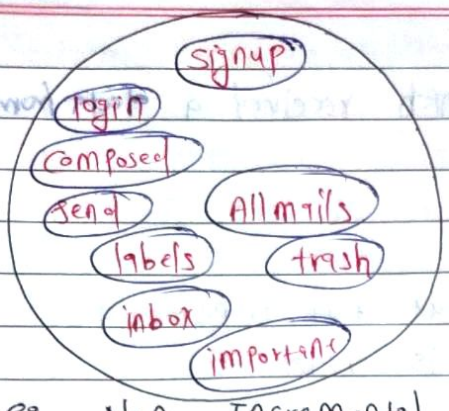
- It is also known as Big-Bang Integration Testing.

2. Incremental Integration Testing
Testing Relation bet two features by incrementally adding one by one features.

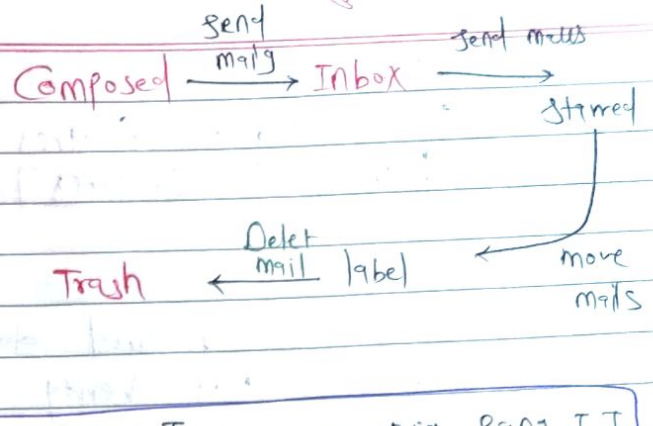
eg. ✓ Integration Testing Incremental

2021/10/21 19:25

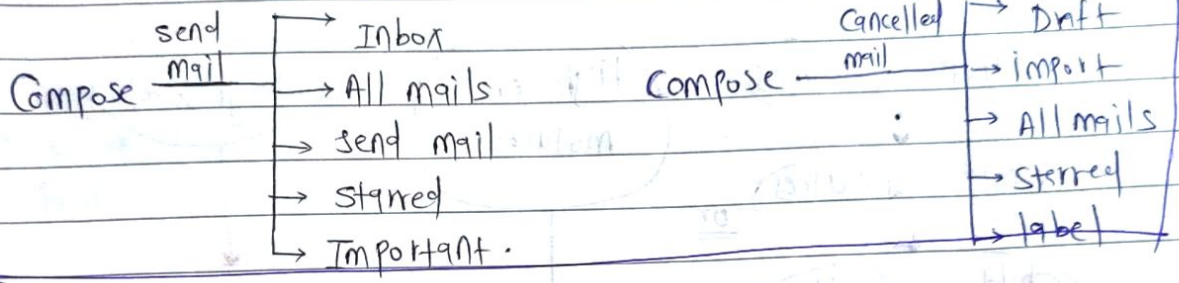
Non Incremental



Incremental



eg. Non-Incremental Intgration Testing or Big Bang I.T.



* Incremental intgration is classified into two types based on dataflow bet^n features.

1) Top down incremental Integration :-

1. Testing relation bet^n parent to child or high level fatures to low level features
2. To perform top down relation should be monodirection

Note- If developed had developed all the functionality then component will be Considered as a features

3. Chances are there developers may developed high level Modules or features but not low level module.
4. In order to perform testing on high level feature tester may required low level features, so developers will create a dummy features [Stubs]

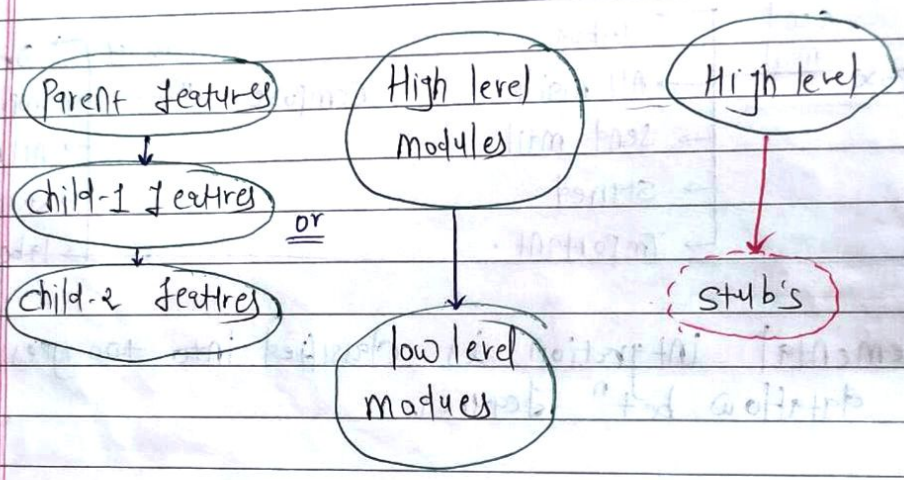
2021/10/21 19:25

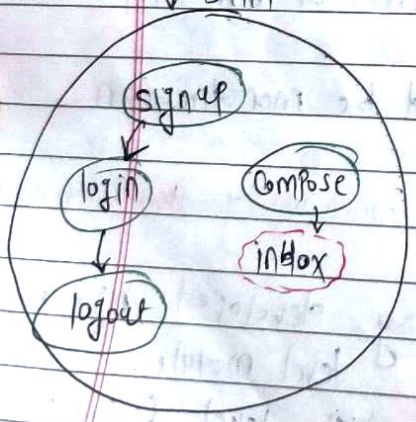
Stub's

Its a dummy features which received a data from high level features

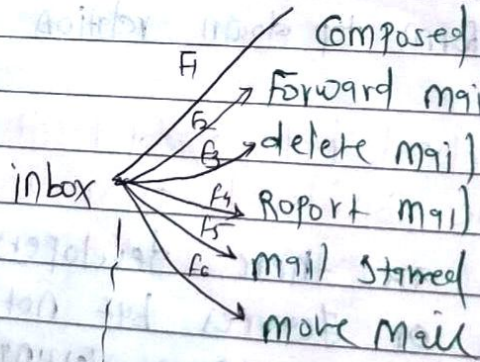
Limitations.

1) Stub's will received data but not transfer data
2) Stub's will not verify data is valid or invalid.





Features

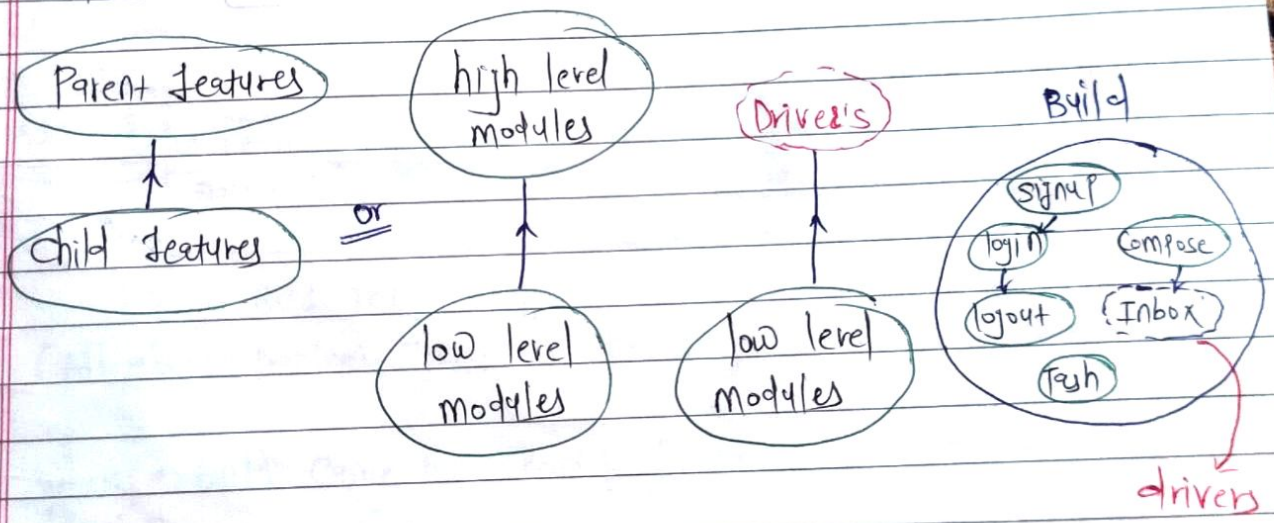


2021/10/21 19:25

2. Bottom up integration Testing
Increment

1. Testing data flow from child to Parent or low level modules to high level module
2. chances are there tester may developed low level features but to high level features
3. In order to perform testing on low level modules tester might requires high level modules so developed will developed dummy features drivers

Drivers
Its a dummy features which receives data, analysis data and transfered data



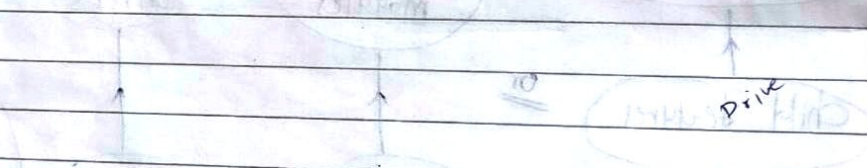
Interview Qye. What is sandwich incremental integration testing
→ 1) Its Combination of top down & buttom up incremental integration
2) In sandwich developers can developed either stub's or driver's based on relation.

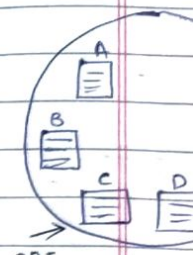
2021/10/21 19:25

2. Difference between Stub's & Driver's

Stub's	Driver's
1) it receives data	1) it receives or transfers data
2) used when low level modules not developed	2) used when high level modules not developed
3) mainly developed in top down approches	3) mainly developed in bottom up approches
4) Developed when rel^n is monodirectional	4) Developed when relation is Bidirectional
5) Stub's are sub Program	5) Drivers are main Prog
6) Stub's are "Called function"	6) Drivers are "Calling Function"

Called fun^n Calling fun^n





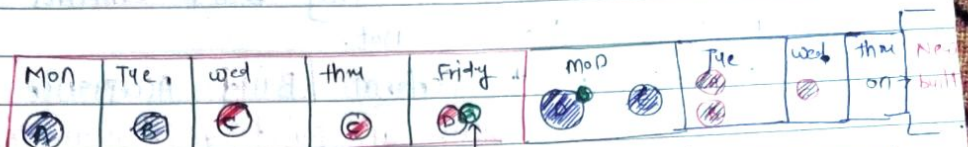
Instead of

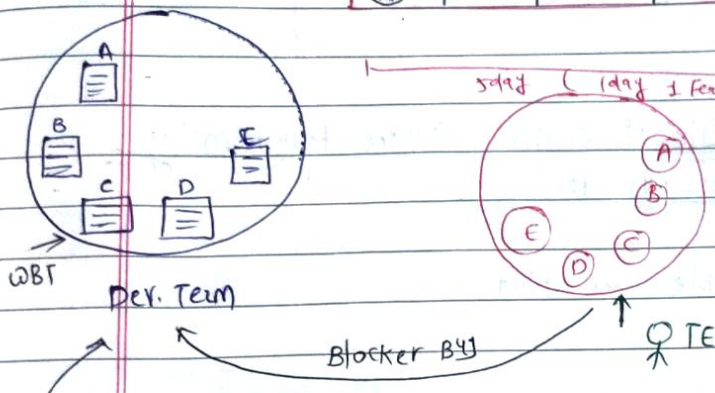
ST

2021/10/21 19:25

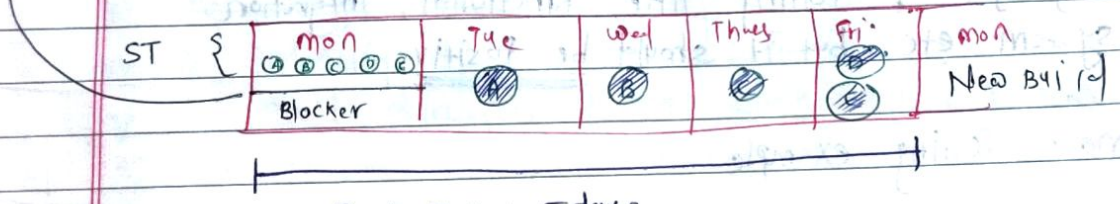
classmate
Date 21-2-20
Page

3. Smock Testing / Build verification Testing (BVT) / Build Acceptance Testing (BAT)





Instead of doing this Following strategy wed ↓



(All above problem Tester is the only Person responcible)

① When a build came for Testing TestLead will decide Test cycle duration.

② within the Test cycle Testing Team need to test all developed Features

③ Chance are there testing team may identifies blocker bug at the end of test cycle. to Fixed that Blocker bug developing team might required sufficient time.

④ After fixing bug ~~risk~~ resbin will be send to the ~~testiy~~ team

⑧ To test the resbin testin team will required sufficient time. because of this delay might be happens in testing new build.

(c) To avoid this delay before accepting the build testing team need to verify build contains any blocker bugs are there are not.

(d) so we should perform Build Acceptance Testing or Smock Testing or Build verification Testing

* Defination of S.T:-

1) Verifing or checking basic & critical Feature functionality before performing thorough testing

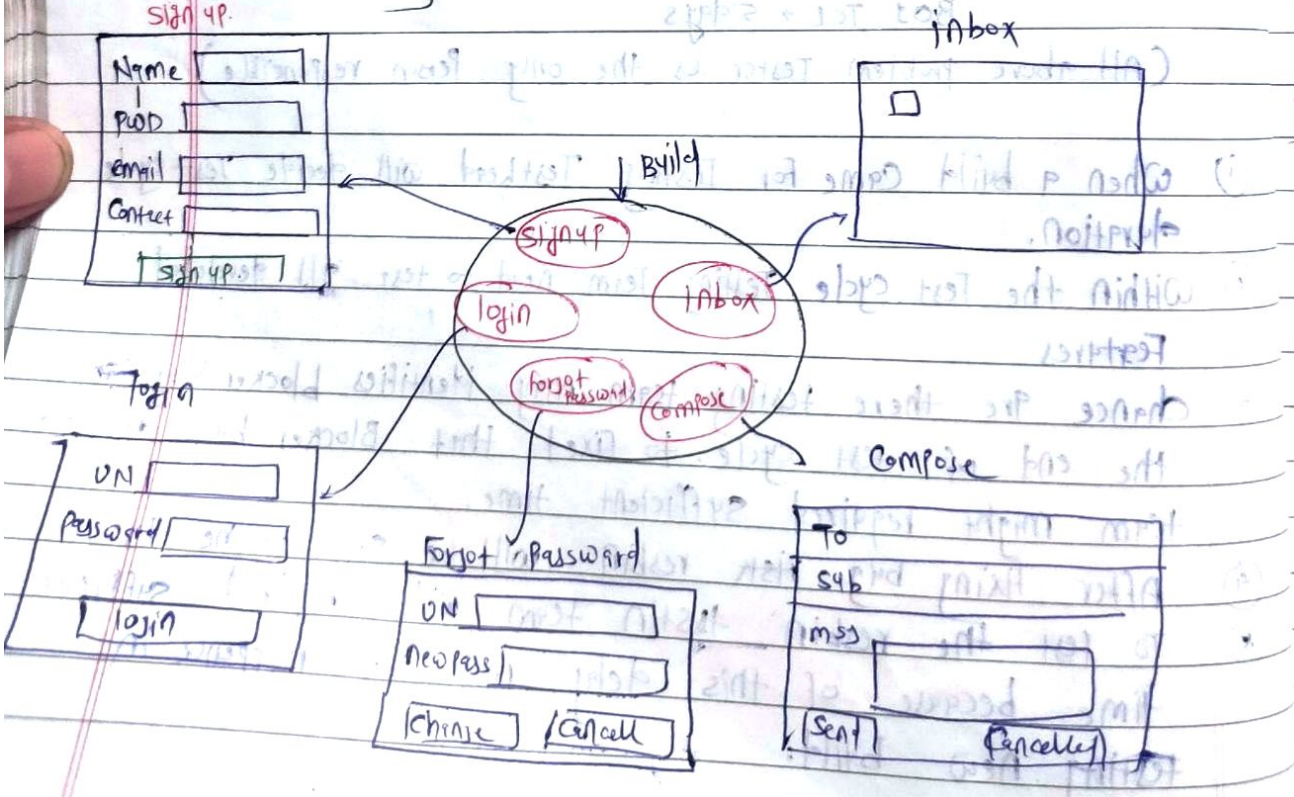
or

2) Checking build is testable or Not.

Q. what type of testing will be perform in smock?

→ Any type of testing like Functional, integration system etc. but it should be positive.

* Smock Testing example.



* Smock Testing Senario -

1) signup - check whether the user can create Account or not

2) login - check user can login with register username & password

3) Forgot password - check whter user can creat new password by using forgot password.

4) Compose - check user can send email or not.

5) inbox - check user can received email or not.

* Why we should perform Smock Testing ?

1) To avoid delay in new build Testing
2) To avoid increasing of Project duration and cost.
3) To indentify blocker by at initial stage
4) To avoid developers fixing blocker by efforts.

imp / Practically industry

5) It act as a entry criteria for developing team to develop new build
6) To verify build is installed properly in testing environment or not

21/2/20

* Compatibility end User Testing

Compatibility testing

12/7/20

* ① Customer will prepared requirement & sent to Project Head
② Project team will decide Base platform in which operating system (os) and browser. Software need to be developed & tested.
③ Archítech will set up developing & testing environments as per base platform.
④ once appln is complitely developed & tested, relised Product to Customer.
⑤ chances are there endusers might used application in different platforms
⑥ Features which are working in ~~one~~ Base platform may not workes in other platforms so endusers might face difficulty
⑦ To overcome this difficulty before relising Product to the customer testing team need to test software in multiple platforms i.e. Compatibiling testing

Definations

- Testing a software in multiple os with different browsers in different versions

Que) What type of testing will Perform in Compatibiliy testing

→ we can perform any type but end to end testing will be Perform mainly

—|

* Compatibiliy testing —
  - Hardware Configuration → O/s, Processer,
    - N/w Configuration → bandwith response
    - Device → Printer & peripherals
    - slw version
  - Software Configuration →
    - Mobile configuration
    - Browser Compatibiliy

2021/10/21 19:25

Que 2) what type of defects tester will identify while doing compatibily testing

→ 1) changes in user interface (UI)
2) Fetures Functionality
3) Font size
4) screen resolution
5) Broken imges
6) css, style and colure.
7) Scrolling

* Types of Compatibiliy Testing :-

1) Forward compatibiliy testing -
Testing software using latest version of OS browser etc.

2) Backward compatibiliy testing -
Testing software using older version of os and browser etc.

Que 3) when we should perform compatibiliy testing.

→ once application is complitely developed and stable in base platform we should perform compatibiliy testing

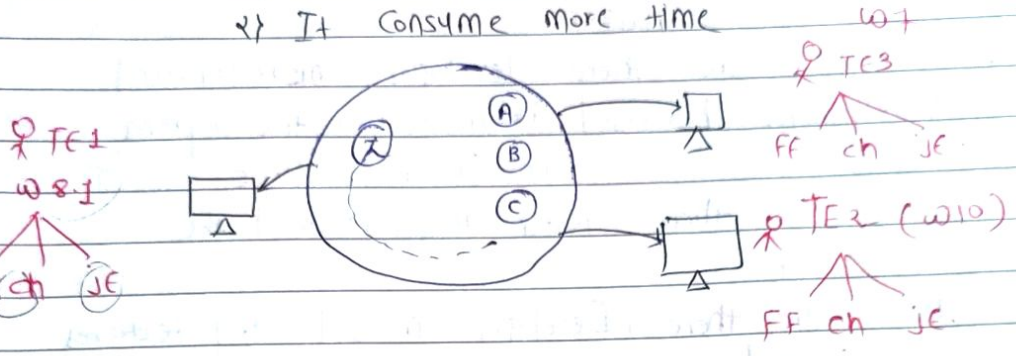
Que 4) Why we should do compatabiliy testing

→ 1) to avoid difficulties that can be faced by end users
2) to avoid loss in customer buiness
3) to avoid negetive impact on customer buiness
4) to improve appln accuracy & efficiency
5) to verify any changes required to make application user friendly

Forward

* How to Perform Compatibility testing
* Approch 1
- In this approch every tester will perform compatibility testing on assign platform.

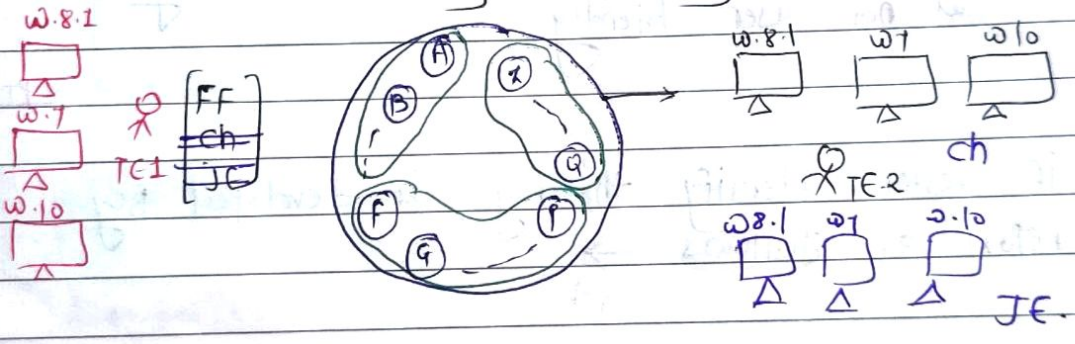
limitation - 1) Tester need to spend more efforts
2) It consume more time



* Approch 2

In this approch every tester will perform compatibility testing on multiple platform using assign feature

limitation - 1) Compatibility testing cost increase.



* Approch 3

By using C.T tools Browser Short, Vertual desktop, Browser stack.

multiple browser in only one system ie pc

2021/10/21 19:25

23/02/20

* Defect

* Deviation between expected and actual result is known as Defect.

Que. Why defect is happen ?

→ 1) chances are there developer misunderstand requirement. because i) mistake in the Program
ii) incorrect logic

2) chances are there, tester misunderstood the requirement

3) chances are there, Developer missed any features to developed

4) chances are there, Developer added the features which is not mentioned in requirement.

5) if tested fills any features functionality is not user friendly.

PTO

* if tester identify the bug it developed BUG report as follows →

Defect

Bug Report

1. Project Name
2. Feature Name
3. Build No
4. Severity
5. Priority
6. Reproducability
7. Platform
8. Bug Id
9. Bug Status
10. Bug Summary:
11. Bug Description:
12. Steps to Reproduce :
13. Expected result.
14. Actual result.
15. Attachment.

PTO

1) **Project Name**
this section explains in which application tester identify Bug

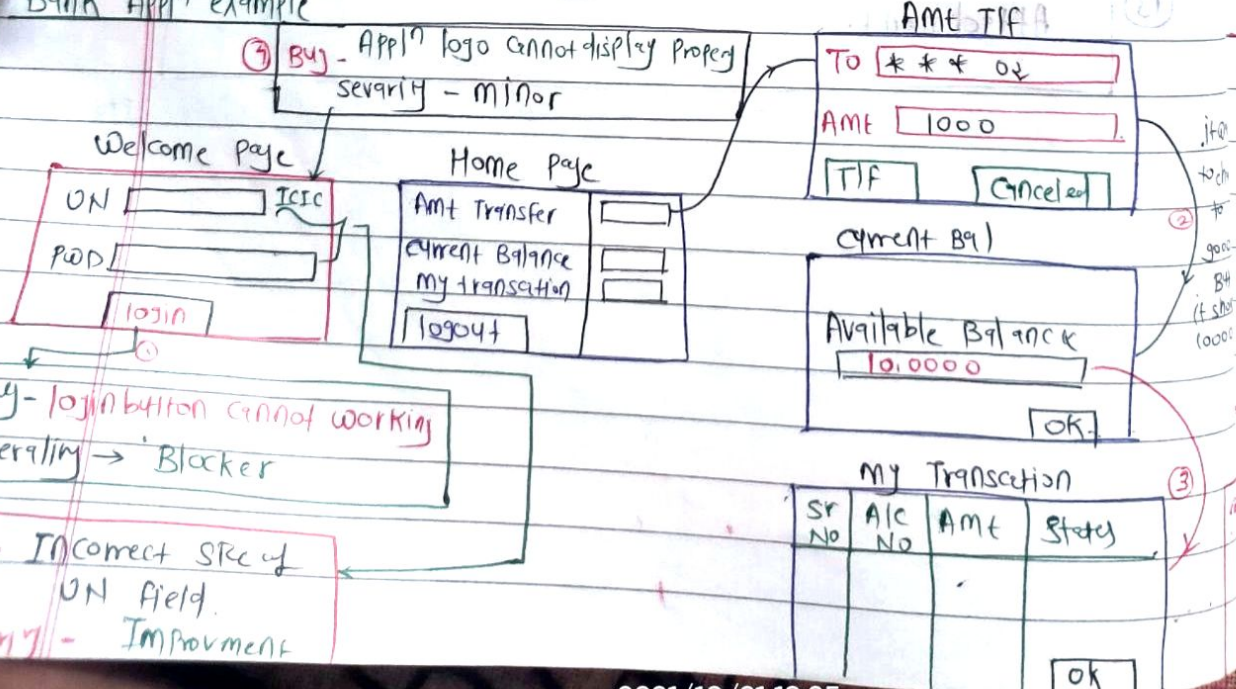
2) **Feature Name**
This section explains in which feature tester identify Bug.

3) **Build No**
This section explains while testing which Build tester found Bug

4) **Severity**
This section explain how identify the bug. which show impact on customer business

* **Types of severity -**
1) Blocker - Further testing Not possible
2) critical - huge loss in customer business
3) major - No loss but customer will face difficulty
4) minor - No loss no difficulty but appln will not look good.
5) improvement - Not a bug it's a suggetion to improve the or feature Functionality of the features

Bank Appln example



5) Priority
This
identi
* Types
1) im
2) hig
3) med
4) low

Note - In
be
set

5) **Priority**

This section explains how Fast developer need to fixed, tester identified Bugs.

* Types of Priority -

1) immediate - Bug need to be fixed within same test cycle
2) high - Bug need to be fixed in next test cycle.
3) medium - Bug can be fixed within 5 to 10 testcycle.
4) low - Bug can be fixed within any testcycle before releasing Product.

Note - In Bug report severity is set by Tester. & Priority will be set by Developers mainly, some time Tester can also set Priority

Severity	Priority
Blocker / (S1)	Immediate / (P1)
High / critical / (S2)	High / (P2)
Medium / major / (S3)	Medium / (P3)
Low / minor / (S4)	Low / (P4)

2)

Bug - Tlf amount Not deducted From current Balance

Severity critical

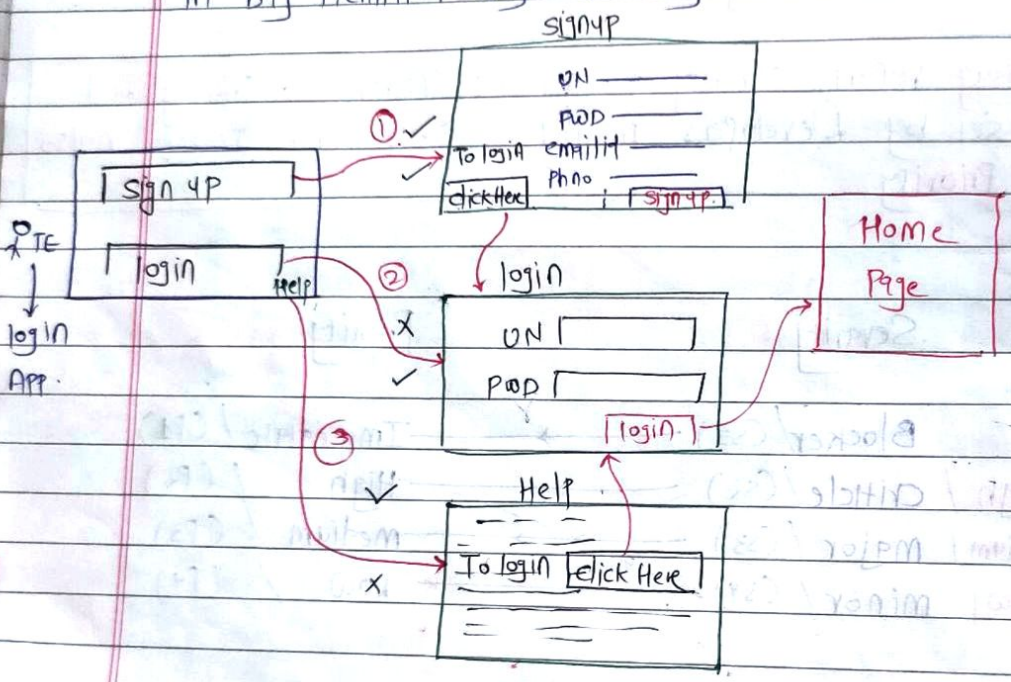
Bug - Transaction detail cannot display.

Severity - Major

25-4-20

1)* High severity high Priority bug :-
This status will be given if identified bug shows serious loss in customer business, no. of users uses application in bug identified way or more

2)* High severity low Priority -
This status will be given if identified bug shows serious loss in customer business, no. of users uses appln. in bug identified way or very rare.



ex 1) Bug - User enable to login through login button
Severity - Critical / high
Priority - high

2) Bug - User enable to login through Help button
Severity - critical / high
Priority - low

* low Severity high Priority -
This status will be given if identified Bug will ~~show~~ not affect customer business, no. of users uses application or more
ex- Spelling mistake in application logo

* low Severity, low Priority -
This status will be given if identified Bug will not show any affect on customer business and no. of users uses application or very rare.
ex. Spelling mistake in application Document. i.e help contaiint

6) **Reproducability**
This section explains tester identified bug is consistance or inconsistance.

* Consistance bug - if tester able to Reproduce bug in all attaimps.
status → Always

* Inconsistance bug - if tester unable to Reproduce the bug in multiple attempts.
status → 1) alternate - Bug Reproduce in alternate attaimpts.
2) Rarely - Bug reproduce twice or thrice.
3) only once - Bug reproduce only once.

Q1) Why we should log inconsistance bug to costomers
Chances are there at customer side inconsistance bug is become consistance to have a Proff document the testing team ~~have~~ a identified bug we should log inconsistance bug

Q2) How ~~we~~ you make to developer to fixed inconsistance bug.
→ By attaching log file

2021/10/21 19:25

* Log file - it is a document which contain records of every action performed on application

7) Platform - The section explains in which operating system, browser tester identified the bug.

8) Bug summery - overview of the Bug.

9) Bug descriptions - in detail about identified Bug.

10) Steps to reproduce - step by step procedure how tester identified bug

11) Expected result - How system should responce or what should happen

12) Actual result - How system is responding or what is happening.

13) Attachment - Bug related document.

* Bug report on Banking appln example. (Critical Serious)

Project Name - ICICI Net Banking App.

Feature Name - Current Balance

Build No - 05

Severity - critical.

Priority - High

Reproducability - Always.

Platform - OS - W7, Browser - chrome

Bug ID, Bug Status → teach onwards remaining

Bug summery - Amt non deducted from account.

Bug description - Payer T/F AMt < Available Bal. to Payee, T/F Amt credited to Payee but not deducted from Payer A/c.

Step to Reproduce - 1) Open app
2) login as Payer
3) click on amt Tf.
4) Tf amt to Payee
5) click on current Balance.

Expected result - T/f amount should deducted

Actual result - T/f amount not deducted

Attachment - log file, Screenshot

21-2-20

* Defect Lifecycle -

① If tester identified the bug, Prepared bug report with status [New / open] and send it to developer Lead (DE)

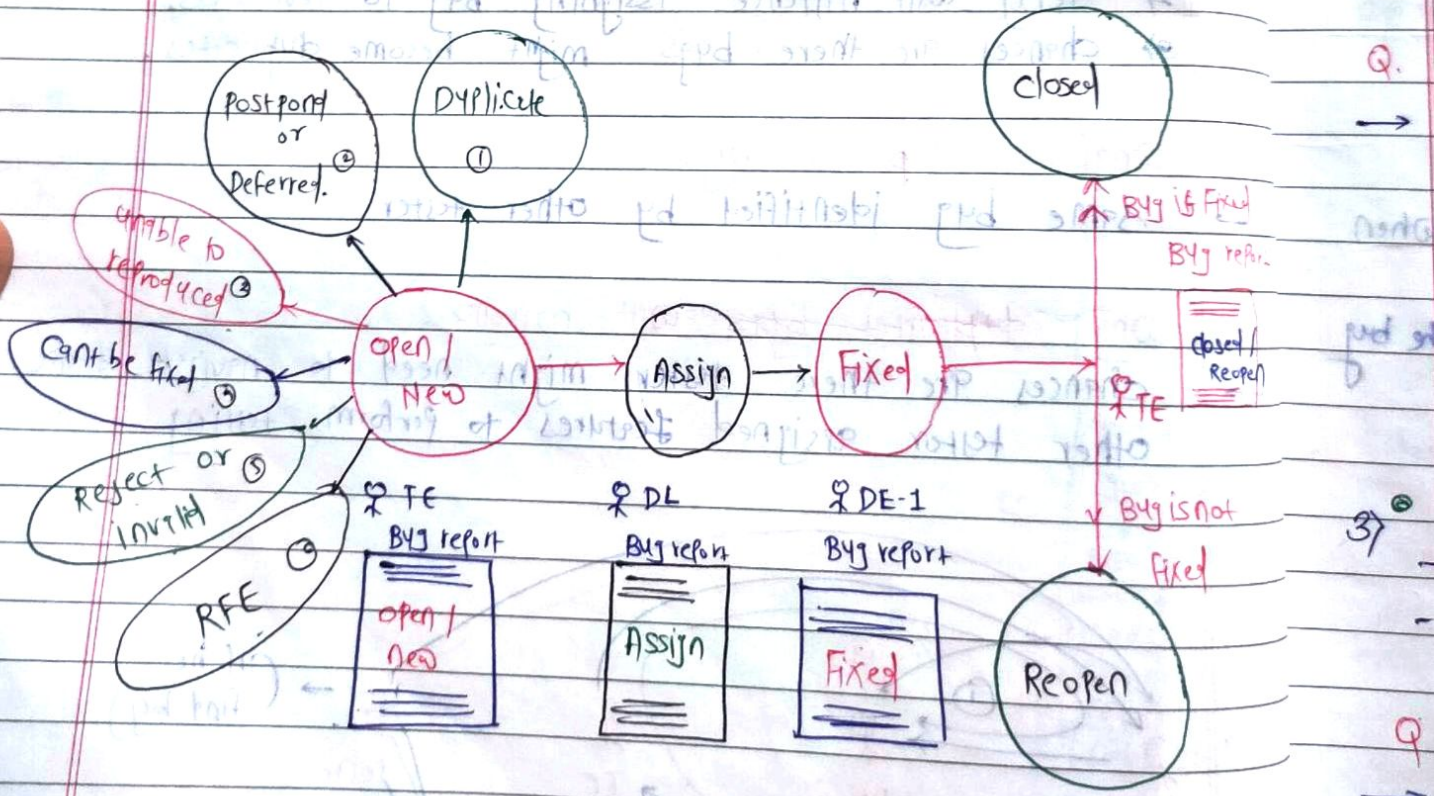
② DL will review the bug report, identified developer who developed that feature. and assign the bug by changing bug report status to [assign]

③ Developer will review bug, reproduced the bug, if bug is reproduced, accept the bug and fixed the bug, change bug report status to [fixed] & send bug report status to tester.

④ Tester will Performs retesting, if bug is Fixed change bug report status to [closed]

⑤ if bug is not Fixed, change status to [reopen] and send it to Dereloped lead.

⑥ This process will continey until every bug is closed.



2021/10/21 19:25

* Status →

1) • **Duplicate** -
- This status will be given by developed Lead (DL).
- Developed Lead (DL) not accepted your bug because similar bug already assign to Developer.

Q) Why Duplicate bug will happen ?
→ ① Because of common Feature navigation
② Because of related features
* Duplicate Bug Journey will be like this,

Open → Duplicate → closed

Imp • **Postponed or Deferred**
2) - This status will be given by Developer
- Developer accepted the bug, but it will be fixed After sometime.

Q. Why we get postponed status ?
→ ① might be bug sevarity is minor.
② might be bug priority is low.
③ if developer is busy in adding new Features
④ if costemer is ~~accepted~~ expecting ↑ changes in bug identifing features

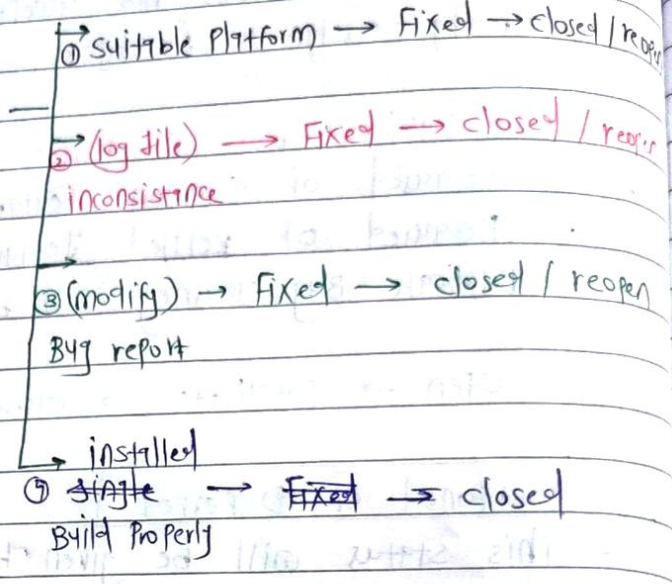
New → Assign → Postponed → Fixed → closed / reopen

3) • **Enable to reproduced** -
- This status will be given by developer
- IF developer enable to reproduced tester identified bug at developing side.

Q Why we get enable to reproduced status.?
→ ① chances are there developer using different platform to reproduce bug (IFF / Crome / etc)
② chances are there bug is inconsistance

2021/10/21 19:25

③ might be bug report is not clear
④ chances are there build is not properly installed in testing environment



4) **Can't be Fixed status** -
- This status will be given by Developer
- Developer accepted the bug but it can't be Fixed

Q) Why bug can't be Fixed
→ ① chances are there Cost of Fixing a bug is more when compare to loss due to that bug.
② technology issue. which is not supported to Fixed the bug
③ identified bug is minor which is located in base program

5) **Rejected status**
- This status will be given by developer
- Developer not accepted tester identified bug

Q) why bug will be rejected?
→ ① chances are there tester misunderstood the requirement
② Developer misunderstood the requirement

2021/10/21 19:25

Request for enhancement (Improvement) (RFE)

③ might be developer unable to reproduced tester identified required bug

④ tester is reffering to old required document which is not updated

New → Assign → Rejected
- 1) TE misunderstanding → closed
- 2) DE misunderstanding → Re open → Fixed → closed
- 3) Dev. unable to reproduced → Re open → Fixed → closed / Reopen
- 4) old requirement → closed.

6) RFE (Request For enhancement)

- This status will be given by Developer
- From testing point of view it's a bug. but development point of view its not a bug because bug related context not available in requirement.
- all RFE's are discussed with customers, if customer accepted, Fixed the bug otherwise, reject the bug.

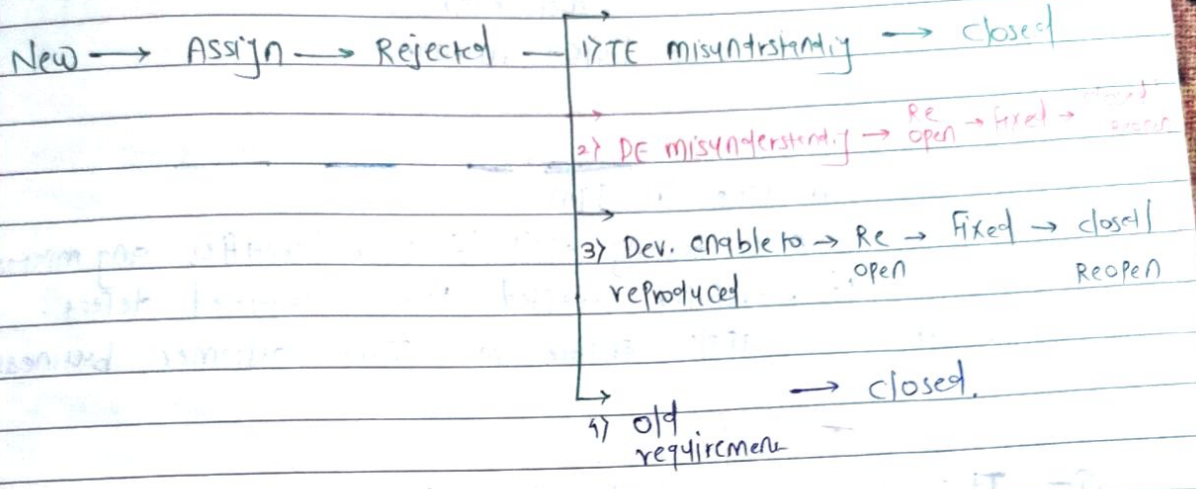
New → Assign → RFE
- customer accepted → fixed → closed / Reopen
- customer not accepted → Rejected → closed

2021/10/21 19:25

Request for enhancement (Improvement) (RFE)

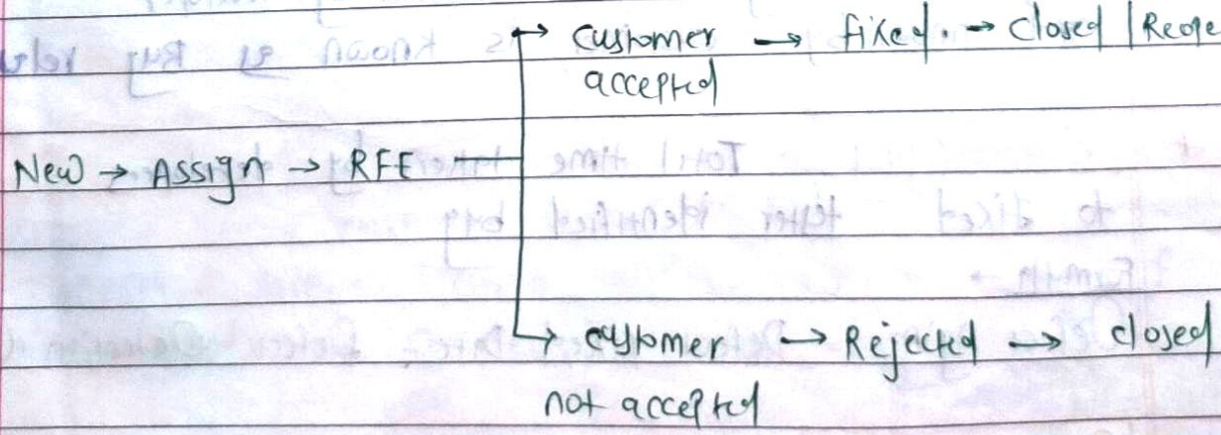
③ might be developer unable to reproduced tester identified required bug

④ tester is reffering to old required document which is not updated



6) **RFE (Request For enhancement)**

- This status will be given by Developer
- From testing point of view it's a bug. but development point of view its not a bug because bug related context not available in requirement.
- all RFE's are discussed with customers, if customer accepted, Fixed the bug otherwise, reject the bug.



2021/10/21 19:25

28-2-20

Q) How will you convince Developer to fixed Bug ? if, developer rejected your bug report.

→ ① Try to show SRS documents.
② show similar application
③ try to explain customer business
④ Forward issue to Project manager.

Q) Difference between Defect, bug, error, failure.

→ 1) error - mistake in pgm
2) Defect - while testing appln if tester identifies any mistake
3) Bug - if Developer accepted tester identified defect.
4) failure - if appln enable to support customer business

* **Defect Triage** :-

① It's a meeting conducted by Project Team when all Features are Completely developed but bugs are existing.

② In this meeting Project team will Priorities which bug need to be fixed first, which bug need to be fixed last & which bug need to be ignore

* **Bug Releas** :-
Bug identified by tester, not fixed by developer and known by customer is known as Bug released

* **Defect Aging** - Total time taken by developers to fixed tester identified bug

Formula →
Defect Aging = Defect Fixed Date - Defect Detection date

* **Defect density** It explain total no of defects present in a feature.

2021/10/21 19:25

Defect density = $\frac{\text{No of bugs}}{\text{No. of lines of code.}}$ <= 1

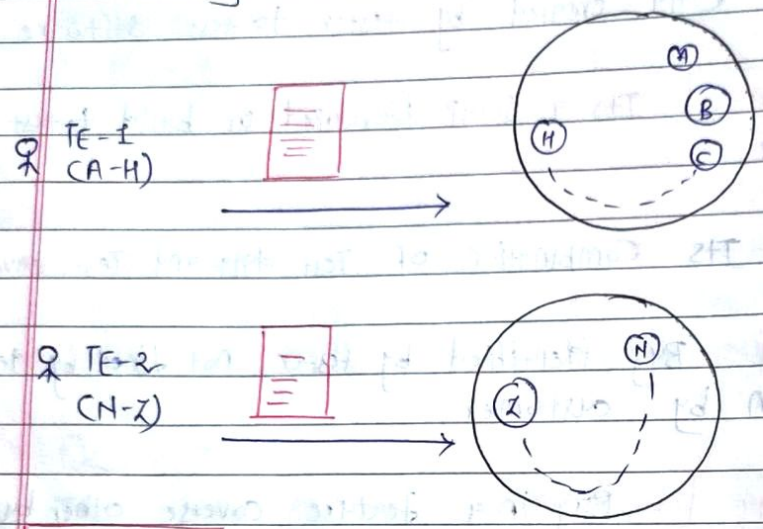
it should be less than or equal to 1

* **Test Data** - Data created by tester to test software
* **Test Environment** - Its a setup Developed or build to test the software.
* **Test Bed** - Its combination of Test data and Test environment
* **Latent Bug** - Bug identified by tester, not fixed by developer & not known by customer.
* **Defect Masking** - Bug in a feature coverse other bug identification until that bug is fixed rest of the bugs in that feature can't be identified.
* ~~**Adhoc Testing**~~ - ~~Ad-hoc of test~~
* **Bug leakage** → Bug identified by endyser while using Product.
* **Not Fixed** → Time taken by Project team to fixed customer identified bug.
* Not Fixed time duration will be 1hr to 1 day & its depends on complexity of the bug.

2021/10/21 19:25

* **Adhoc Testing** (Monkey / Gorilla)

In this type every tester need to identify creative scenarios on there assign features and perform adhoc testing

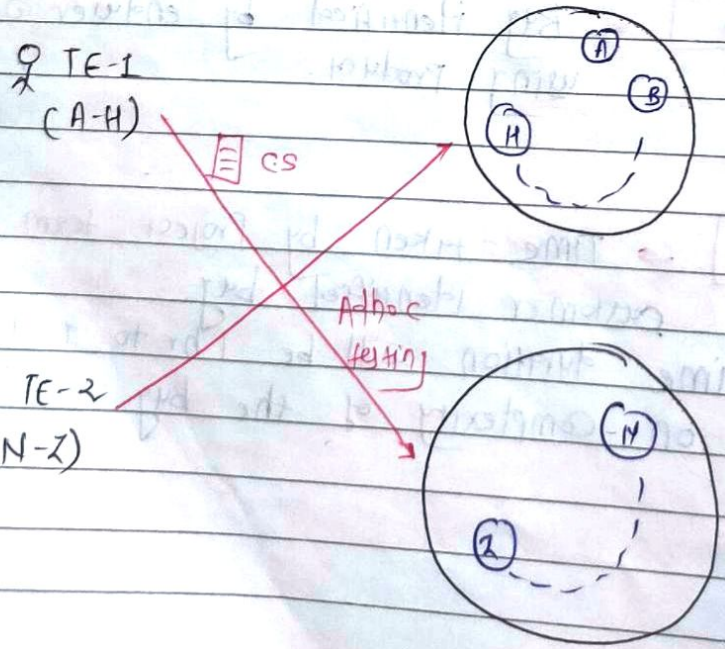


Definition - Testing an application without using any logic

It having 3 types →

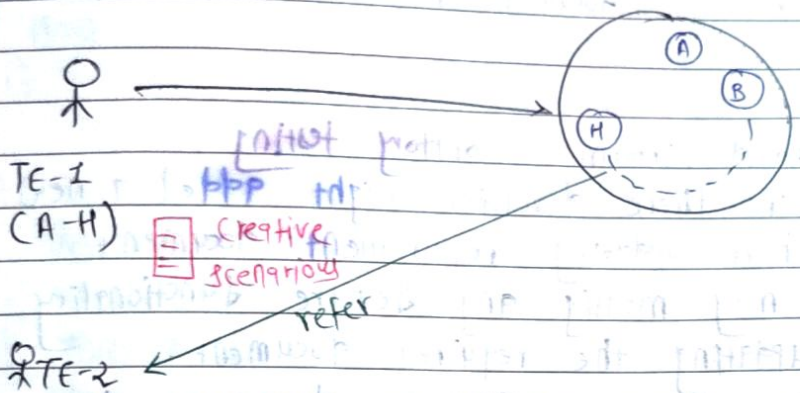
1) Interchange Testing -

In this every tester need to identify creative scenarios on other tester ~~identify~~ Feature and need to perform Adhoc Testing assigned



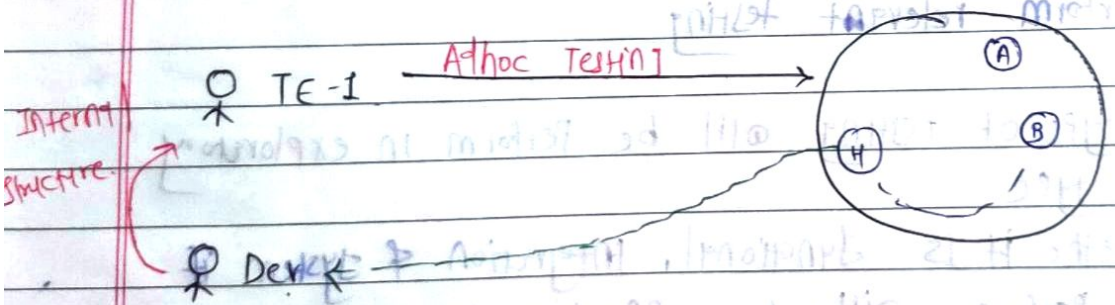
2. Pair testing -

In this type tester will identify creative scenarios. by referring to that creative scenarios other tester will perform adhoc testing.



3. Buddy Testing -

In this type developer will explain internal design & coding feature to tester based on that knowledge. tester will identify creative scenarios & perform adhoc testing



* Reliability Testing -

Testing a feature functionality continuously for a period of time.

- **Exploratory Testing**

Testing on appln based on understanding feature functionality without reffering to requirement document.

Que 1] When we should perform exploratory Testing?

→ When requirement document not sufficient to perform Testing.

Que 2] Why we should perform exploratory testing

→ 1. chances are there developer might added a new feature without updating requirement document.
2. customer may modify any feature functionality without updating the required document.
3. chances are there requirement document is to old which is not suitable to perform testing

* Procedure to perform exploratory testing
1. understand the feature functionality
2. Prepare test case or identify all possible solution
3. Perform relevant testing

Que 3] What type of testing will be Perform in exploratory

→ 1. Any type
2. whether it is functional, integration & system if it is perform without reffering to requirement then it is exploratory

- **Limitation**

1. To understand feature functionality Tester may required sufficient time. so testing duration might increased

2. Chances are there tester may understand feature functionality right as wrong & wrong as right.

3. In exploratory tester can't identify more no of bugs. So application quality might effect.

Que 4] How to make exploratory testing effectively?

→ 1] I will understand feature functionality by exploring.

2] I will try to explore similar type of appln to understand feature functionality

3] Interact with dev & explain each feature functionality, what I understood to the developers if any changes are there developer will update

4] If I have any doubt even after iterating will developer I will communicate with customer to resolve doubts.

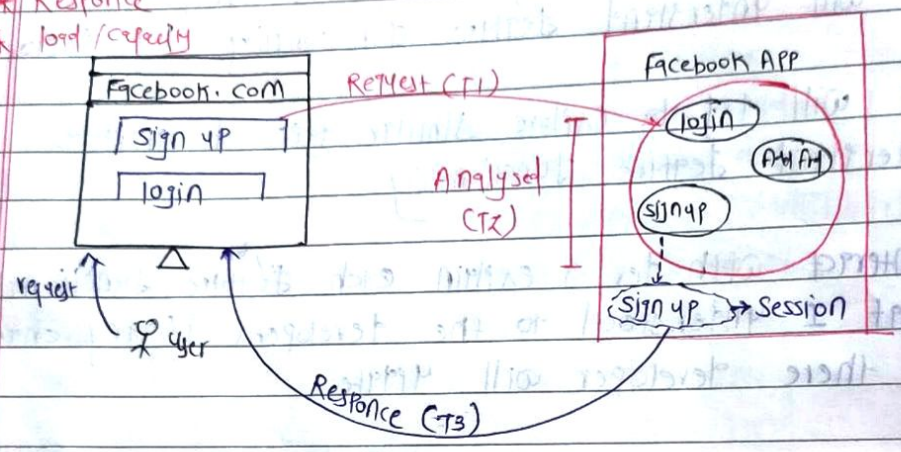
11-3-2020

* Performance testing - (Non Functionality testing)

Def 1 - checking or verifing response time of application Feature

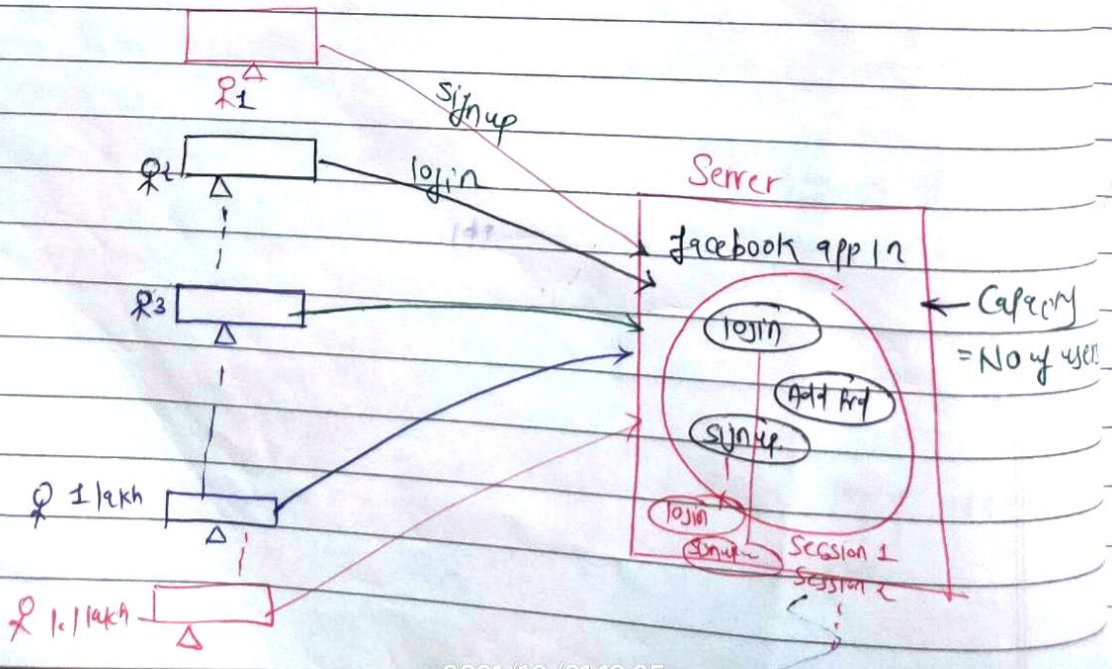
* response time = time taken by system to send request to server + time taken by server to analysed the request + time taken by server to send response to system

* Response
* load / capacity

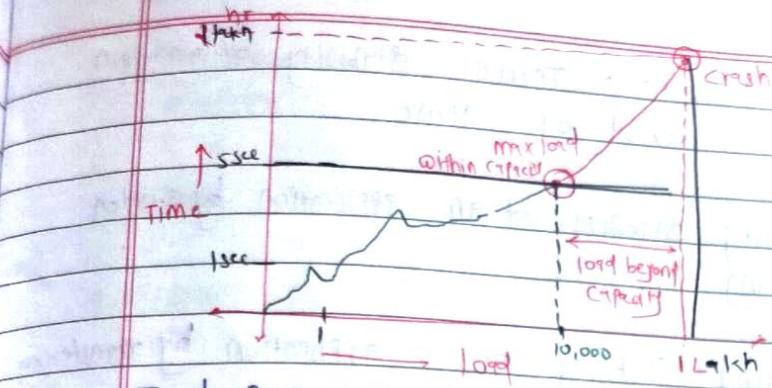


Response Time = T1 + T2 + T3
≤ 5 sec.

Def-2 Testing appln capacity by applying load.



2021/10/21 19:25



- To do performance testing we used automation testing tools like
  a) Load Runner
  b) Silk test
  c) Jmeter
  d) Selenium 3/4

* Why we should perform Performance Testing ?

→ ① To avoid loss in customer business
② To overcome difficulties that can be faced by enduser
③ To verify Speed - To determine to response of appln
④ To check scalability - To determine user load on traffic
⑤ To identify Stability - checking capacity at various load & situation

* When we should Perform Performance Testing ?

→ ① once appln is stable before relasing Product to Customer

Imp ● **Types of Performance Testing** -

1) Load testing - Testing Stability of an appln by giving load less than maximum load.

2) Stress or fatigue testing - Testing Stability of an appln by giving load more than maximum load.

2021/10/21 19:25

3) Soak or endurance Testing - Testing stability of an appln by appling load for a period of time.

4) Spike testing - Testing ~~stability~~ of an application by sudden increasing & decreasing of load

5) Volume testing - Testing stability of an application by transfering huge data.

* **Recovery Testing** -

① This type of testing will be perform if appln is crash at customer end, after rebuilding the software.

Defn - Verifing or checking appln is recovered from crash or not

OR

verifing or checking rebuilded software is able to work like stable software or not.

Need of test Case

① Required document shared with both development & testing teams
② Development team will start developing the app build wise (Few Features will be developed)
③ After performing UBT dev. will send Build to testing team
④ To test a build testing team will used following Approaches.

Approch 1 :-

①By refer req. tester will identify input & test the features

limitation -

1) testing depends on tester interest
2) Consistency will not be there in testing
3) App Quality might effect.

Approch 2 -

① By refering to req. tester will prepared test Cases.
② by referring to those test Cases tester will Perform testing

Define Test Cases :- its a Document which will contain all possible inputs that are used to perform testing

1) Why we should write test Cases -

→ 1) Consistancy will be there in testing.
2) App Quality will be improved
3) No depend on tester interest.
4) we can avoid training of new tester
5) we can avoid increse of project cost & duration

2021/10/21 19:25

P.T.O.

6) It will act as a proof document for automation testing
8) To depend on process rather than person

* test scenario (it will explain before)

Assignment

* ~~Bank Application~~
* Bank Employee Module (Create Account)

Project Name - ICICI Net Banking APP.
Module Name - Bank Employee module.
Feature Name - Create Account.
Type of Test scenario - Function test scenario
Req. No -
Severity - critical
platform - os window 10, Browser crome.
Brief Description - Documenting Function test scenario For create Account Features.

Sr No	Sub Feature	Test scenario Id	Test scenario	Status
1	Name	1) TS-ICICI-~~Creat Account~~ Bank employee - Creat Account - Name - FT-01	check 'Name' field accept valid data or Not.	
		2) TS-ICICI-Bank employee - Create Account - Name - FT-02	check 'Name' field accept invalid data or Not.	

2021/10/21 19:25

2. Account No.	1) TS- ICICI - Bank employee- Create Account - Account No- FT- 03	1) Check Account no. field accept valid data or not.
	2) TS- ICICI - Bank employee Create Account - Account No- FT- 04	2) check Account no. field accept invalid data or not
3. Contact No	1) TS- ICICI - Bank employee- Create Account - Contact No- FT- 05	1) Check Contact no field accept valid input or not.
	2) TS- ICICI - Bank employee- create Account - Contact No- FT- 06	1) check Contact No field accept invalid input or not.
4. UserID	1) TS- ICICI - Bank employee Create Account - UserID- FT- 07	1) check user ID field accept valid input or not.
	2) TS. ICICI - Bank employee Create Account - UserID - FI- 08	2) check user ID field accept invalid input or not.
5) Password	1) TS - ICICI - Bank employee create Account - Password FI- 09	1) Check Password field accept valid input or not.
	2) TS - ICICI - Bank employee create Account - password FI- 10	2) check Password Field accept invalid input or no

2021/10/21 19:25

6. create button	TS-ICICI-Bank employee -create account-create button-FT-11	check create account msg display or not.
7. Cancelled button	TS-ICICI-Bank employee -Create account-cancelled button-FT-12	check Cancell button field is work or not.

Author:
written date:
Reviewed by:
Reviewed date:
Approved by:
Approved date:

2021/10/21 19:25

* Integration testing on Bank employee create acc feature

Assignment

Project Name - ICICI Net banking App.
Module Name - Bank employee module
Feature Name - Create account
Type of Test Scenario - Integration Test scenario
Req No - 3.4 3.4.1 3.4.2 3.4.3
Serverity - Critical
Platform - OS window 10, Browser crome
Brief Description - Documenting IT scenario for create ACC. A/M.T.

Sno.	Valid invalid	Test Scenario ID	Test Scenario
1)	Valid Scenario	Ts-ICICI-Bank employe -Create Account-IT-01	check Bank employee Create ACC. then ACC holder should login or not.
2)	Valid Scenario	Ts-ICICI-Bank employee -Create Account-IT-02	check Bank employe Credited Money to Account holder acc. should credit to ACC. holder module or not.
3)	Valid scenario	Ts-ICICI-Bank employee -Create Account-IT-03	check Bank emplyee blockes the Account holder acc. then account holder shoule able to access Account or not.
4)	invalid scenario	Ts-ICICI-Bank employeed-Create Account-IT-04	check Bank employee fills incorrect information Cant ~~create~~ acc. of account holder then also acc. holder should login or not.
5)	invalid Senario	Ts-ICICI-Bank employee. Create account-IT-05	Bank employee Cancelled credit amount to acc holder then check

2021/10/21 19:25

			Whether it displayed amount credited to acc. holder module or not.
6)	invalid ~~ac~~	TS-ICICI-Bank employee-Create account-II-06.	check Bank employee can unblock acc. holder account then ~~also~~ he ~~faced~~ any problem ~~or not~~ to login or not.

Autor - Mayurik
Written Date - 29 JUN 2020
Reviewed Date -
Reviewed by -
Approved by -
Approved date.

Test Scenario Introduction

classmate
Date 26-6-20
Page

* Difference bet^n TS & TC

→ Test scenario → All possible ways to test requirement

ex.
Req. login to APP.
TS-01 - login to app using login feature
TS-02 - login to APP by using signup feature
TS-03 - login to APP by using help features

Test Case → Step by Step Procedure to execute test scenario
TS-01 - login to APP using Login feature
Step 1 - 1. open APP
2. Click on login button
3. Enter Mobile number & Password
4. click on login

* Where we should write test scenarios?

→ 1. WordDoc / Notepad
2. Excel sheet
3. spread sheet
4. Test Management Tools: Testlink, Qc
5. Project Management Tools: JIRA, Redmine, ALM (APP life cycle management)

* What type of test scenario need to prepared?

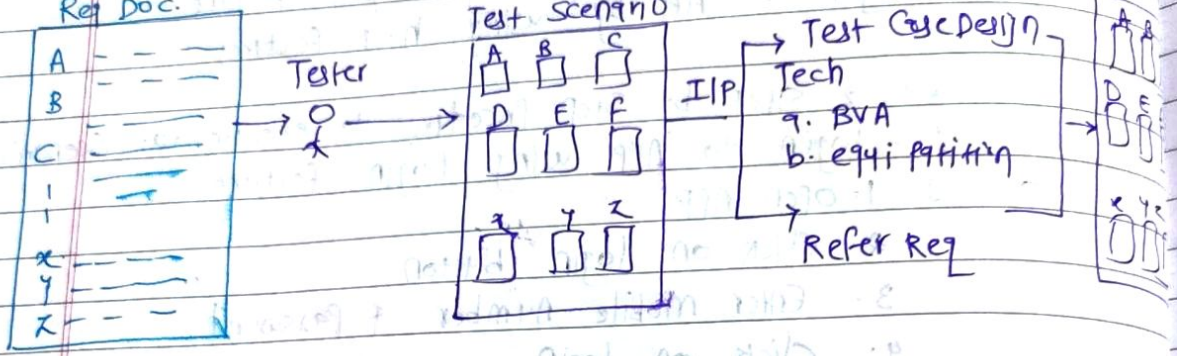
→ 1. Functional Testing T.S. Doc
2. Integration Testing T.S. Doc
3. System Testing T.S. Doc
4. Adhoc Testing T.S. Doc.

* To prepare Test Scenario Doc TL / TM we give Test Scenario Templete that need to be followed by testing team
  - Templete will Contain 3 section
    a) Header b) Body c) footer

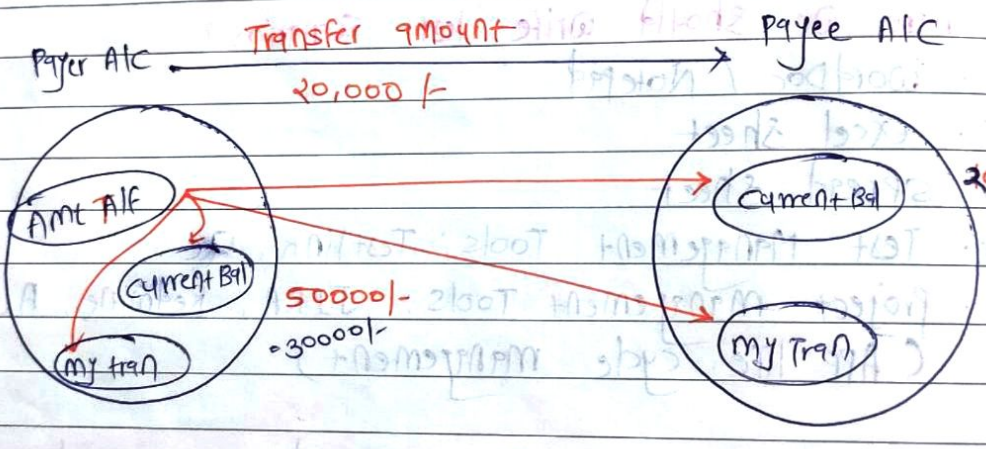
2021/10/21 19:25

* Account holder - Amt transfer Feature (Explain by sir save in Drive)

* Bank employee (done Priviou page)



27-6-20



* Test Scenario Review Procedure →

* Test scenario Review procedures

1) Tester by referring to req doc. prepares TS doc for assigned features
2) ~~TS~~ TS doc will be send to TL for approve.
3) TL will assign TS doc to other Tester (Reviewer) to review doc
4) Reviewer by refering to req doc reviews TS & identify mistakes
5) All mistakes will be doc. i.e Test Scenario Reviews Doc ~~TS~~ & it and send to the author.
6) By referring to review doc Modify TS doc & send modified Doc to reviewer
7) Reviewer will recheck modified doc & send confirmation to TL
8) TL will approve TS doc.

2021/10/21 19:25

((This test review document is in mobile google drive given by sir))

Q.) What are the possible mistakes that made by Tester in Ts doc?

→ 1) wrong scenario
2) Missing scenario
3) Modify scenario.

Q.) How TL will identifies Reviewer?

→ 1) changes are there based on relation between features
2) changes are there based on testing experience
3) changes are there based on domain knowledge

Q) What are the ethics that need to be followed by reviewer while reviewing doc?

→ 1) Always review certain but not author
2) Always identify mistake but not solution to that mistake
3) After Approving Doc if there is any mistake both Author & Reviewer are responcible

* Test Case design Technic

- this tech are used to identify input by referring to requirement

1) a) Boundary value analysis (BVA)
2) b) equivalence partition technic
3) c) Error Guessing
4) d) State Transition
5) e) Decision Table
6) f) Use Cases.

a) BVA Technic ⇒ If any Component accept ranges A to B, then to test that Component inputs are

A-1, A, A+1, B-1, B, B+1

ex)

ex)

1)

2021/10/21 19:25

Date
Page

2-1-20

b) Equivalence partition -

1) Pressmen Tech →

Tech 1) ranges ⇒ A - B -- to test Component inputs are one valid and Two invalides

Tech 2) Set of values ⇒ test Component for one valid and two invalid inputs.

Tech 3) boolean values ⇒ test anyone Component for both true & false.

2) practice Tech →

Tech 1) range ⇒ test Component for multiple valid & two invalids

- Multiple valid are identify by making sub ranges / part. each part is known as equivalence clys.

Tech 2) Set of values ⇒ Multiple valids & two invalids

Tech 3) boolean values ⇒ test multiple Components for both true and fawse.

2021/10/21 19:25

Date
Page 7-7-20

3) Use Cases - How system should responce to user action

Ex. ATM software:

Scenario :- Account holder withdraw amount From ATM.

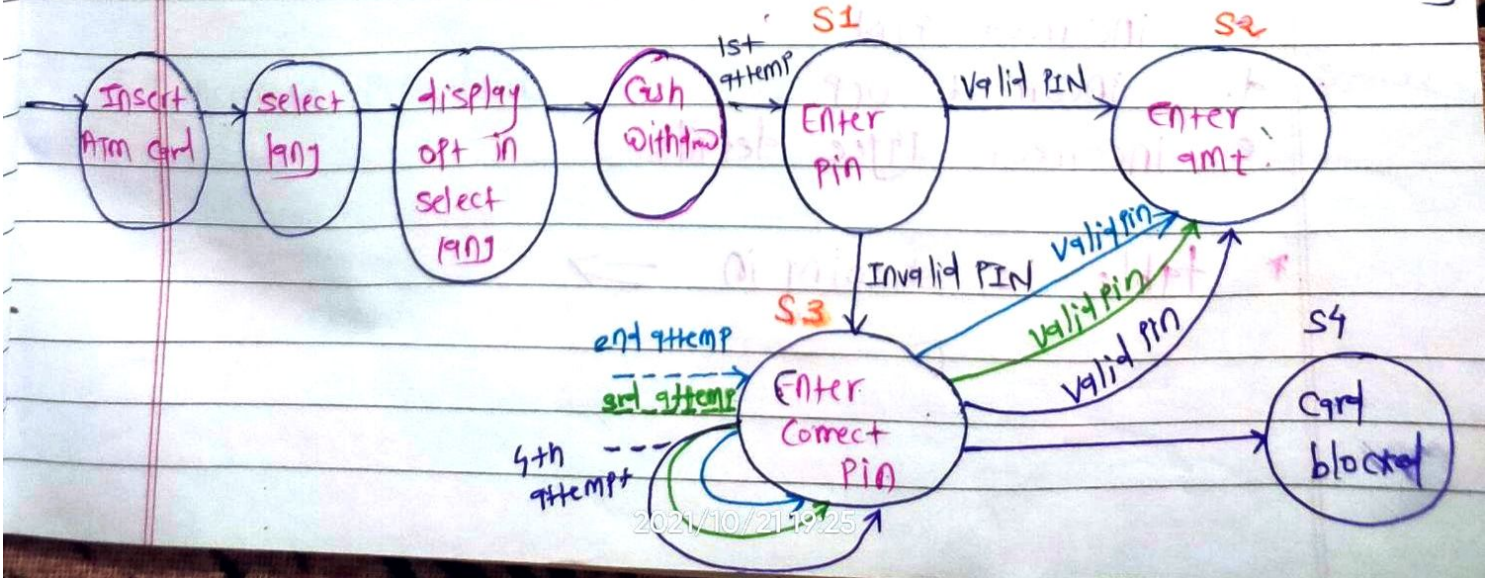
user action	system responce
1. Insert ATM card	1. should display lang select option
2. user select lang	2. should display operation option in selected lang
3. user select's cash withDraw options	3. should display Enter pin
4. User should enter valid PIN	4. should display withdraw type
5. user select's withdraw type	5. should display Enter withdraw Amount
6. user enter valid amount <= avail. Bal.	6. should calculate amt & amt should be relased

4) State Transition - This tech used to identify how system mode or state should change based on privious history

ex. ATM s/w -- Enter PIN number --

Requirement - if user enters incorrect PIN number for 4 time continously then

a. Account holder card should be blocked temporily



SNo	Attemp	Valid PIN	Invalid PIN
1.	1st attemp	S1 --- S2	S1 - -- S3
2.	2nd attempt	S3 - -- S2	S3 - - S3
3.	3rd attempt	S3 - - S2	S3 - - S3
4.	4th attempt	S3 - - S2	S3 - - - S4

5) **Decision Table** -

* it is also known as Cause effect Table.
* if customer requirement contains too many conditions (if else logic) to identify test case with this technic.

ex. Mobile Banking App.

Req. When Account Holder Transferring Amount -

Conditions

1. Payee details should be correct
2. Amt should be less than or equal to avail. Bal
3. Should enter correct PIN number
4. Enter OTP.

Actions

a. Amount transfer successfully
b. insufficient fund in ur account
c. incorrect PIN
d. incorrect OTP
e. incorrect Payee details.

* table is showing in ⟹

2021/10/21 19:25

Sl No	Condition	TC-01	TC-02	TC-03	TC-04	TC-05
1.	Payee details should be correct	True	True	True	True	false
2.	Amount should be less than or equal to availabe balance	True	True	True	false	N.P
3.	should enter correct PIN number	True	True	false	N.P	N.P
4.	Enter OTP.	True	False	Not possible	N.P	N.P
	Actions.					
1.	Amount Transfer successfully	Execute				
2.	insufficient fund in ur account				Execute	
3.	incorrect PIN			Execute		
4.	incorrect OTP.		Execute			
5.	incorrect Payee details					Execute

5) Error guessing -

- in this tech tester will identify i/p randomly
- main purpose of using this tech is to identify bug
- This tech will not follow any specific rules

Good Practice

1) this tech will used only by exp. Candidate
2) this tech will be used if tester have good knowledge of app (domain)
3) By analysing previous test execution dat

8/7/20

* How to write Test Cases -

1. Identify for which feature are we write test case (ex. Amount Transfer)
2. Identify type of test case are we write (ex. Functional Test Case)
3. Understand requirement for that feature (ex. Functional req)
4. Identify input for each req. using Test case design Tech
5. Always fill Header part
6. Start filling Body part. always write Navigation steps clearly upto feature
7. Document Valid input first and then invalid input is
8. for every feature we should prepare test case.

Ex.

1) Feature (Amount Tlf)
2) Type of Test Cases : Functional Test Cases
3) Req:
4) INPUTS:

Test case ex is in mobile Spreadsheet

3.3.1 : To
a) it should accept 10 char
b) only +ve no. are allowed
c) alpha, spl char, -ve no. operator, Blank & space not allow

inputs
a) 10 char +ve No - 1234567890
b) < 10 char +ve No -- 1234
c) -ve No --- -1234
e) splchar --- @#;@#$

3.3.2 Amount
a) it should accept Amount within range 500-50000
b) decimals allowed within range
c) spl char, -ve no's, blank, space, operators Not allowed

2021/10/21 19:25

Inputs
a) 499, 500, 501, 49999, 50000, 500001
b) 30000, 100, 60000
c) 499.99, 50000.01, 12345.67
d) @ 100, $ 50, one hundred, Rs 500.

3.3.3 Transfer button -
a) Amount Transfer msg should display
b) insufficient Bal msg should display
c) unreg Account Number msg should display
No. input

3.3.4 : Cancel button
a) Tran cancelled msg should display
No. input

9-7-20 Integration Test Cases -
1) identify feature that need to be documented.
2) understand Req. for that feature (IT)
3) Identify All scenario
4) Always fill header part first
5) start filling body part with navigation steps
6) Always document valid relation first & then invalid relation.

* Test Case Reprository: Its a common location where every tester will store all approved TC.

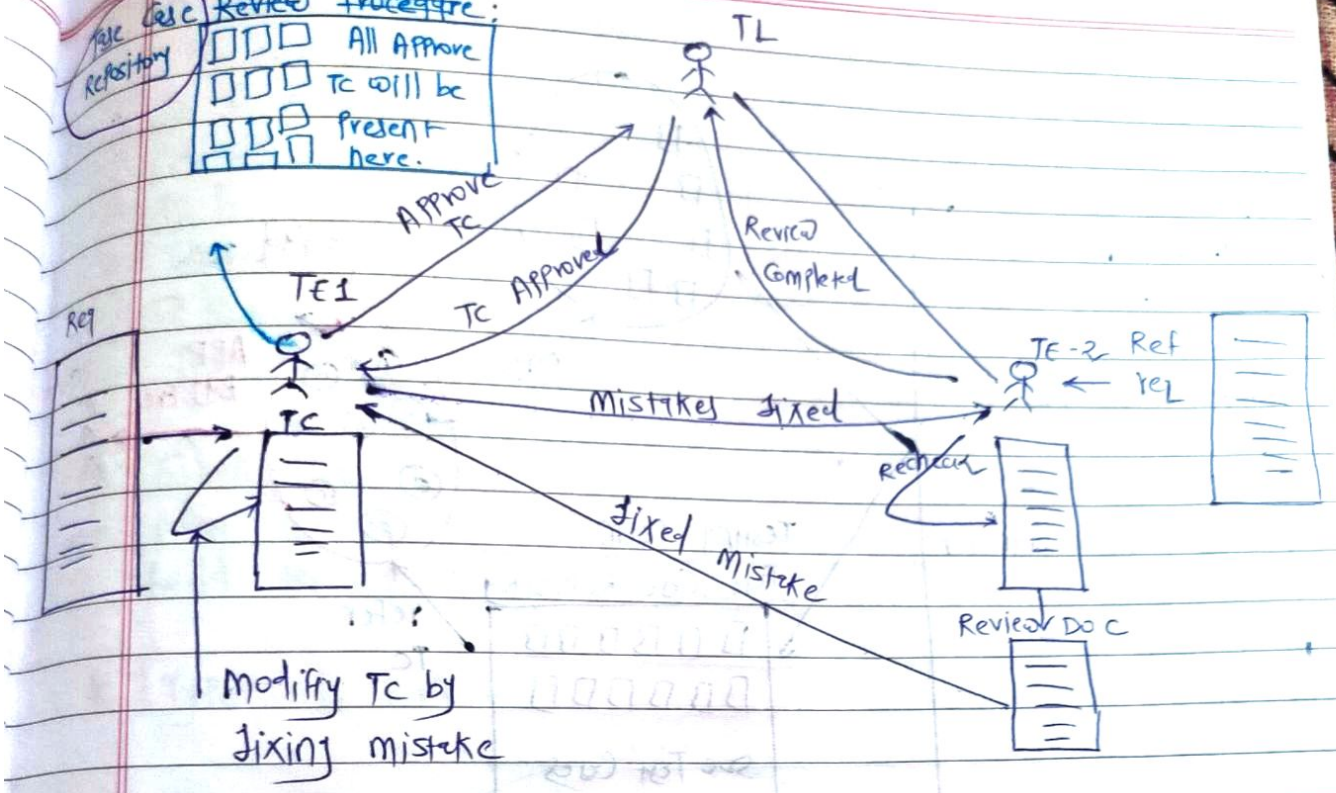
• which can act as Tc repository?
Ans) 1) Common shared folder
2) Test Management tool like QC / Test link
3) Project Management tool like JIRA, ALM

Test Case Review Procedure :-

Test Case Repository

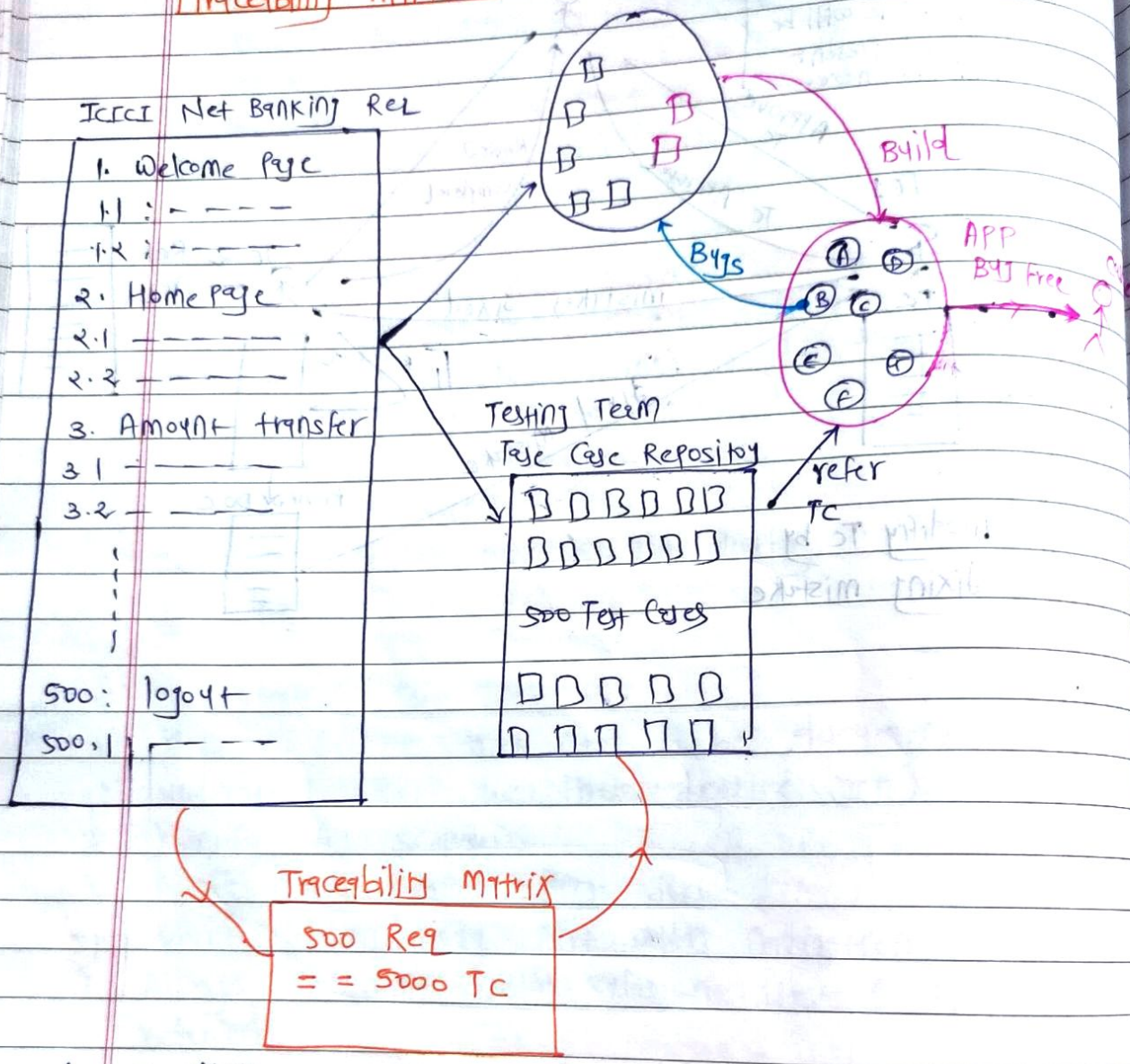
Test Case Review Procedure :

- All Approve
- Tc will be
- Present
- here.



11-7-20

Traceability matrix :-



* Traceability Matrix →

Its a document which will verify req given by customer is covered atleast once in Test Cases Prepared by testing team or not.

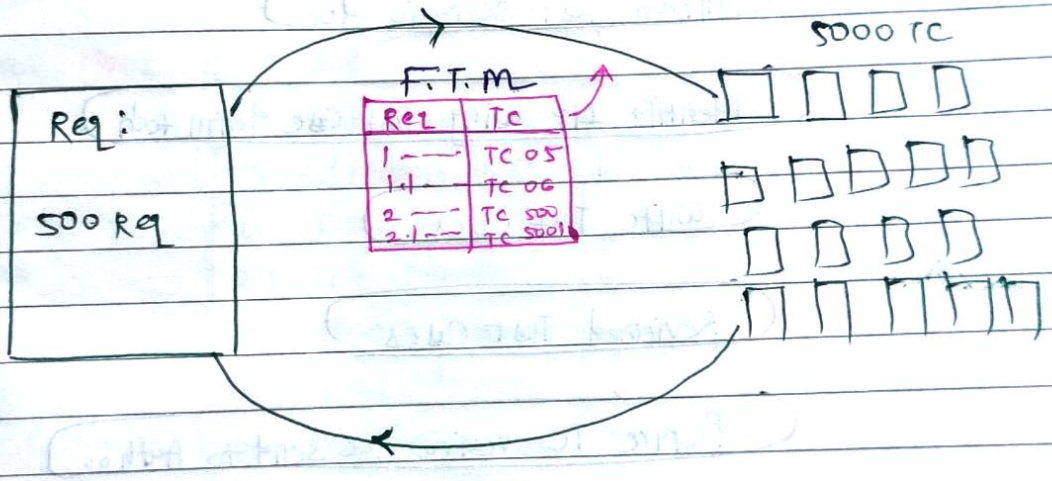
* Why we need to Prepared Requirement Traceability Matrix

→ ① Tester can easily identify whether all requirement are covered in his test cases or not.

② Tester can easily identify Test Cases that need to be modify if customer changes any requirement.
③ if customer removes any requirement tester can easily identifies Tc that need to be removed
④ it act as a Proof doc for customer that Testing team covered all requirement.
⑤ some of Tc can't be automated by Automation Tester -- those Tc need to be Tested Manually

* When we need to prepared T.M?
→ After storing Tc in test case repository & before testing the build testing team will prepare T.M.

* **Type of T.M →**

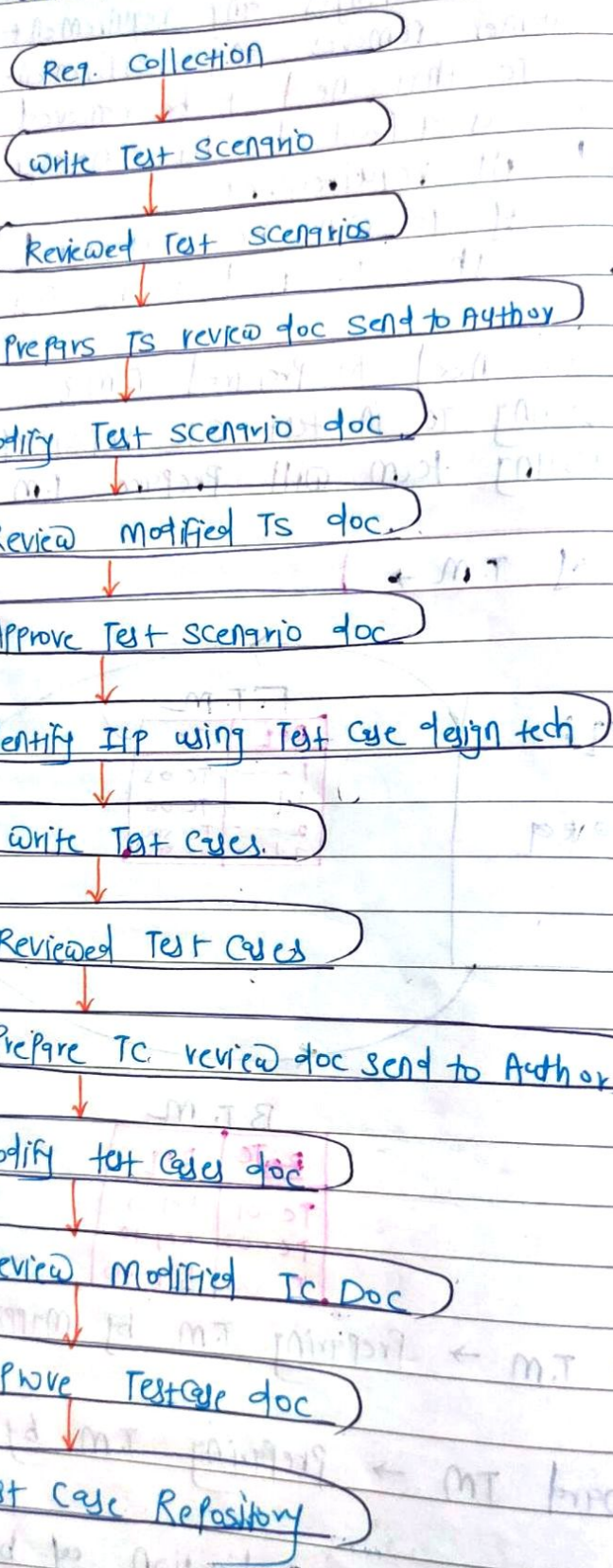


B.T.M

Req Tc	Req
Tc-01	Req 01
Tc-02	Req 10
Tc-5000	Req-400

1) Forward T.M → Preparing T.M by Mapping from req to Tc
2) Backward TM → Preparing T.M by Mapping from Tc to req
3) Bidirection TM → Combination of both FM and BTM.

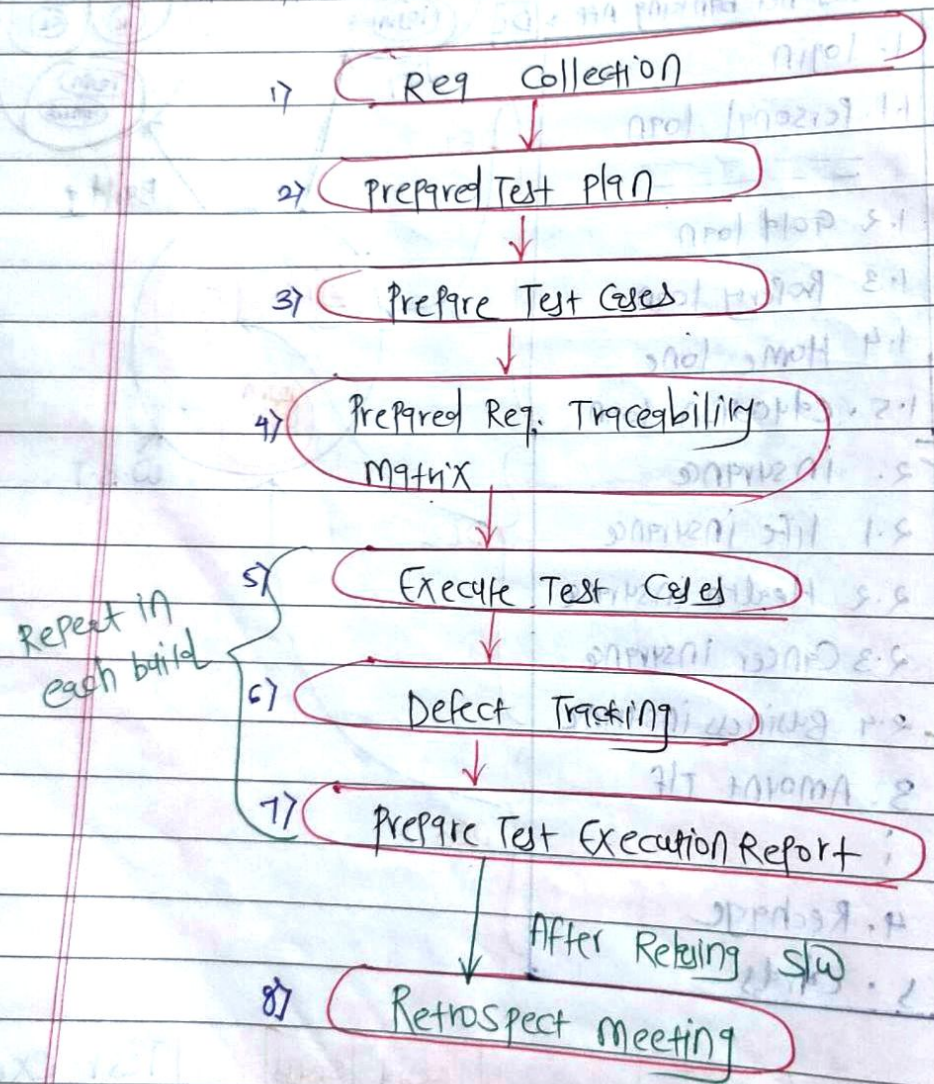
* Procedure to write test cases →



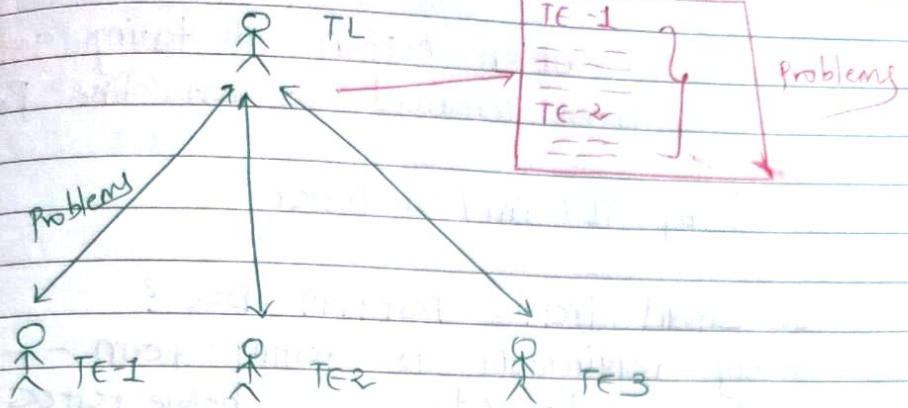
* Test Execution Report :-

SNo	Feature	Total Tc	Tc Expected	Tc Pass	Tc Fail
1.	loans	1500	250	230	20
2.	insurance				

* Software Testing life cycle (STLC) →
it is procedure followed by Testing team to perform Testing on Application.



Retrospect Meeting



* Difference between SDLC and STLC

SDLC	STLC
① It is procedure followed by dev. app.	① It is procedure followed to test app.
② In SDLC developers, BA, AR testers are involved	② In STLC only testing team is involved.
③ STLC is part of SDLC	③ SDLC is not part of STLC
④ SDLC is used to optimize cost, quality and time	④ STLC ~~implem~~ is used to deliver high quality Product.
⑤ SDLC is implemented by using model like VfV, Agile etc	⑤ STLC implements using testing type like Functional integration etc.
⑥ to handle SDLC Process we used tools like ALM.	⑥ to handle STLC Process we used tools like QC, testlink JIRA etc.

2021/10/21 19:25

17-7-20

Test plan

Def → its a document which explain all testing activities that need to be perform performed to test the Product

* it is prepared by TL / TM / sr. Tester.

1) When we should Prepare Test Plan Doc ?

→ After sharing requirement to testing team --- tyter will be busy in understanding req. before testers start writting TC, Testlead should prepared Test Plan document

2) Why Test Plan is required ?

→ ① It act as a proof document for customer i.e. that testing followed which approach to test Product.

② this document will explain Procedure that need to be Followed to test app for new tester.

③ if tester face any difficulty while testing app. -- by reffer to this doc tester can easily identify solution.

* Standard Test Plan should contain 15 sections

1) objectives → This section explains main aim of Preparing this test plan doc.

2) Scope → This section explain feature / components that need to be focused / tested and not to be tested

2.1) Test cycle duration is very less

2.2) component that need to be tested

a) focus on critical features

b) focus on valid / +ve testing

c) focus on feature, which

3)

4)

5)

6)

7)

2021/10/21 19:25

2.1.2) Components that not to be tested
a) Dont focus on major & minor features
b) Dont focus on invalid, -ve testing
c) Dont focus on Feature which are stable / having less no. of bugs.

3) test Approch → This section explains approch followed by testing team to test the App.
3.1) Understand req for assigned features
3.2) Prepare Test scenario
3.3) review test scenarios
3.4) Prepare Test Cases.

4) schedule → This section explain test activities that need to be performed within estimated time.

5) Assumption → This section explain expectation of test lead
5.1) Every Team Member will stay until Project Completion
5.2) Testing should complete within estimated time

6) Risk → This section explains possibility of difficulty that might happen while testing.
6.1) one of the team member quit the job
6.2) Testing Product is delayed.

7) Mitigation / Backup → this section explains solution to overcomes the difficulty
7.1 → hire new tester & train them on application
7.2 → increse Team size / testing duration.

PTO

2021/10/21 19:25

8) Test Environment → its a setup which contains hardware & software configuration where testing team need to perform testing.

8.1) Server side configuration.

8.1.1) H/w configuration

a) Type of server

b) Hard disc size

c) Processer.

8.1.2) s/w configuration

a) os

b) server s/w

c) Graphy card

8.2) local m/c configuration

8.2.1) H/w configuration

a) Hard disc

b) processes

8.2.2) s/w configuration

a) os

9) Defect Tracking → This section explain how to testing team need to log & Track the bugs

9.1) Procedure to log bug

9.1.1) Defect report Templete

9.1.2) Severity

a) Blocker

b) critical

c) major

d) minor

9.1.3) Priority

a) immidiate

b) High

c) medium

d) low

2021/10/21 19:25

9.1.4) Reproducability
9.2) Bug status
9.2.1) open
9.2.2) New
9.2.3) Assign
9.2.4) fixed

10) Roles and Responcibility →
Role → Position of the team member in testing team
Responcibility → duties / task
10.1) Test Lead
a) Assign Req. to testing team
b) Prepare Test Plan
c) Provide sufficient time to tester to understand req.
d) Prove Test Scenarios etc doc prepared by testing team
e) interact with dev team
f) Conduct meeting with testing team
10.2) Sr tester -
a) understand req
b) write test cases
c) review Test Cases
d) execute Test Cases
10.3) Jr tester -
a) understand req
b) execute test cases

11) Deliverables → this section explain documents that need to be delivered to customer at the time of relasing product.
11.1) Test Plan
11.2) Test Scenarios
11.3) Test Cases

11.4) Traceability matrix
11.5) Relase Note

12) Templete - This section explains templetes that need to be followed by testing team to prepare document
12.1) Test plan
12.2) Test scenario
12.3) Test scenario review Doc
12.4) Test Cases
12.5) Traceablity matrix.

13) Entry & Exit Criteria
1) Entry criteria → It explains condition that need to be satisfied to start the activity
2) Exit criteria → to stop activity condition that need to be satisfied

13.1) Black Box testing
13.1.1) Entry criteria
a) UBT should be completed
b) Test cases need to stored in Test Case repository
c) Test plan Doc should be redy
d) Test Environment should be setup
e) Testers need to be available.

13.1.2) Exit Criteria
a) App should be bug free
b) Customer req should be completely satisfied
c) All Test cases should be executed & status should be pass.

14) Test Stop criteria → (when testing Team need to stop testing SLW)

14.1) if app quality is too high i.e whatever customer expected working fine.

14.2) if app quality is too low.

Ex → if testing team identify more no of bug but dev team not fixing bug then app quality becomes low

14.3) if app testing estimated cost is exceeding

14.4) if app testing exceeds estimated time / duration

15) Test methodologies → this explains types of testing that need to perform on APP.

15.1) Functionality testing type

a) Function testing

b) integration

c) system

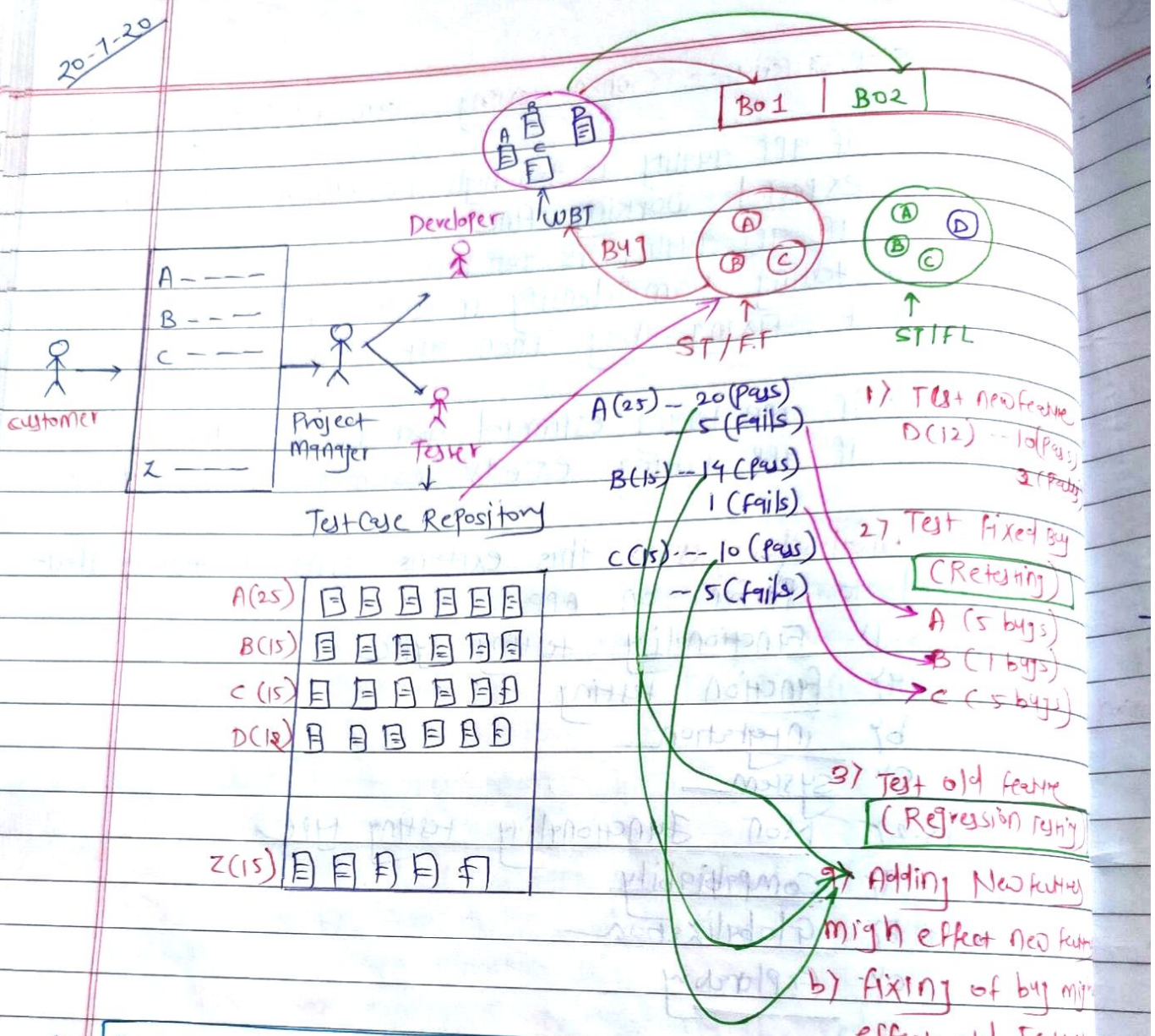
15.2) Non Functionality testing type

a) Compatibility

b) Globalization

c) Exploratory

20-1-20



* **Regression Testing & Retesting**

1) **Retesting** -:
   1. checking ~~fixed~~ Fixed bug to ensure bug is really fixed or not.
   2. Verifying modified Feature is effective or not
   3. Reexecuting previous build Failed test case to ensure bug Fixed or not in new build

2) **Regression Testing** -

① Testing old feature in new build

② Testing unchanged feature in ensure because of fixing a bug its related features functionality effected or not.

③ Testing old feature ~~other feature functionality~~ to ensure because of adding new feature other features functionality effected or not.

④ Reexecuting previous build pass test cases in new build to ensure old feature effected or not.

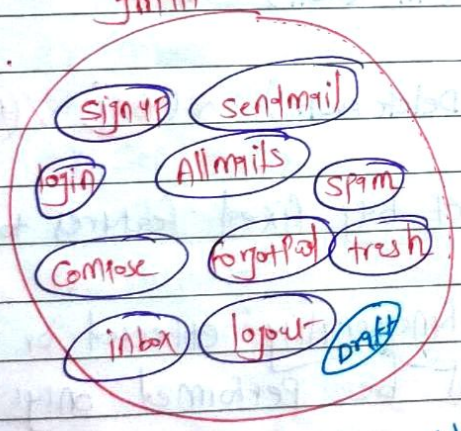
Q) How will you decide on which feature we need to perform regression Testing ?

→ Based on relations,

a) How new added feature related with old feature

b) How fixed bug related with old features

ex.



Analyze the build

i) what changes dev made in this current build.

a) New feature added

b) which features bugs are fixed

Regression ---- relation

New Features — Draft

a) Compose

b) Inbox

c) All mails

d) sendmails

e) Spam

f) Trash

Bug fixed — signup

a) signup

b) login

c) logout

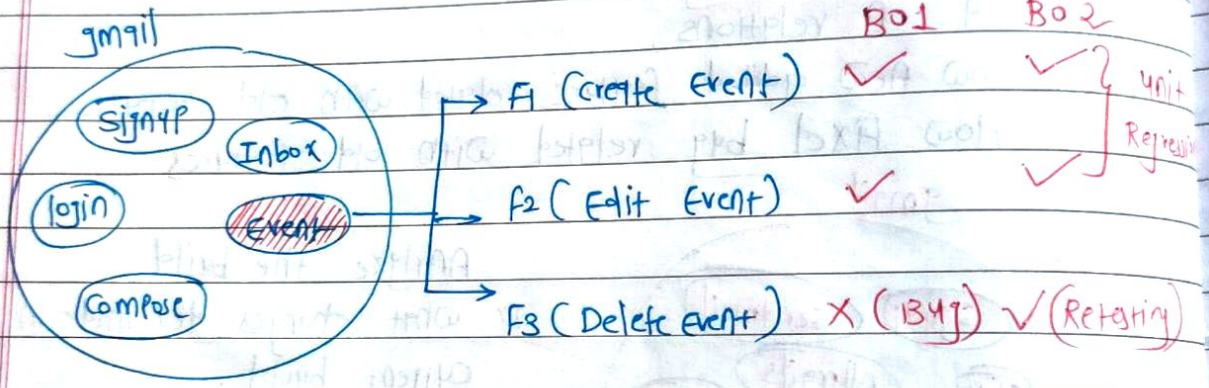
d) forgotpwd

Q.2) Q) What type of testing will be Performed in regression?
→ Any type of testing on old feature will be considered as regression

Q.1) Q) When a build come for testing -- do we need to perform regression testing on all old features?
→ No

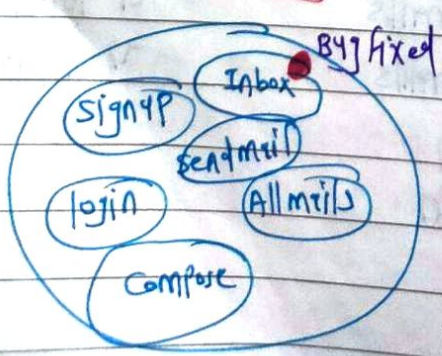
* TYPE of REGRESSION TESTING

1) Unit Regression Testing →



① Testing other functionalities of bug fixed features to ensure because of fixing a bug.
in one functionality other functionality effected or not.
② Unit Regression Testing will be performed only on independent features

2) ~~Regression~~ Regional Testing →



2021/10/21 19:25

① Testing Related features to ensure because of fixing a bug in one feature other related features effected or not.

3) full regression Testing -
① Testing all feature functionality

* When we do FRT →
→ Before release product to customer for 2-3 build FRT will be Performed

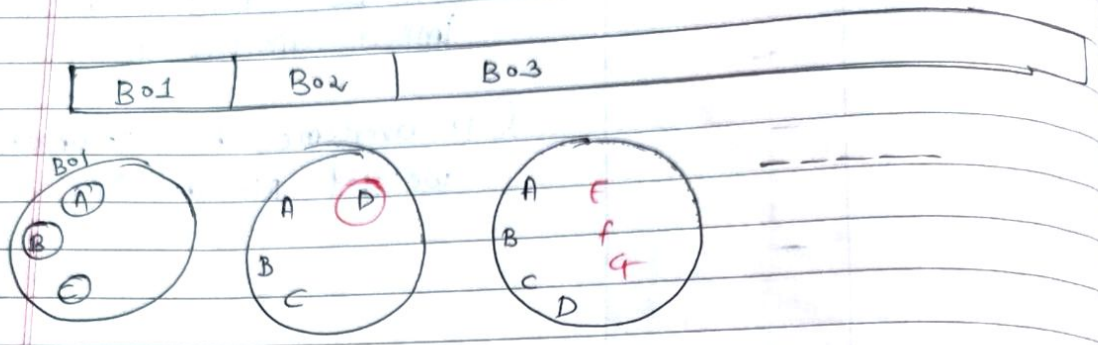
21-7-20
How will you identify features that need to be performed with regression testing
→ Approch -1 → Based on Application knowledge
Ex - gmail Application →
Bug is fixed in composed features
Features that need to be performed with reg testing
a) Inbox
b) Draft
c) Send mails
D) all mails
e) important
f) starred.

limitation → 1) chances are tester might miss one or two related feature regression testing
2) New Testers might face difficulty in performing regression testing on old features.

Approch - 2 → Prepared impact analysis matrix before performing regression testing.
[IAM] → its a document which explains which bug identified feature / newly added feature shows impact on other features

In this meeting every team member will explain how update build will effect old feature & which feature need to performed with regression testing

* Limitation of Regression testing



No of feature → 03	New features - 01	New features - 03
	old feature - 03	old feature - 04
	total feature → 04	total feature - 07

① Tester need to spend more efforts to perform reg. testing
② Project duration & cost is increased.

* To overcome reg testing limitation we used Automation testing.
* Automation testing is used to do reg testing (old features)

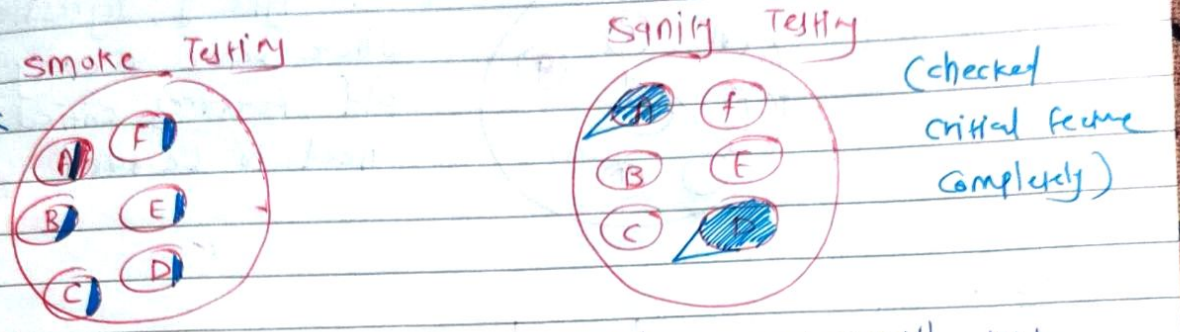
* Difference between regression & retesting

Regression	Retesting
① Testing old features	① Testing fixed bug
② In reg. we check unchanged features	② In retesting we check changed features

* Sanity Testing

Def-01 → Sanity testing is similar to smock testing

Sanity == smoke testing



its a build verification testing where tester will test critical features completly before accepting build.

* Diff between smoke & sanity

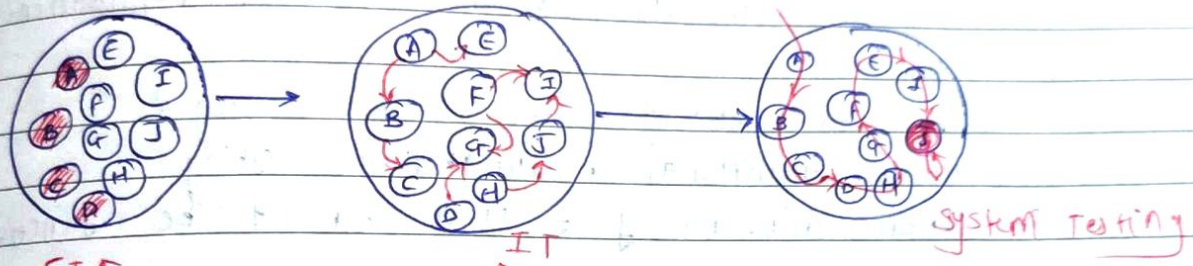
smoke	sanity
1) its build verification testing	1) its build verification testing
2) in smoke all features functio-nality will be tested	2) in sanity only critical feature will be tested completely
3) only +ve testing perform	3) Both +ve & -ve testing perform
4) smoke is wide and shallow	4) sanity is narrow and deep.
5) smoke is scripted	5) Sanity is unscripted

2021/10/21 19:25

System Testing → its end to end testing Performed when testing environment should be similar to Production environment.

* who required s/w → customer
* why customer req s/w → to handle customer business

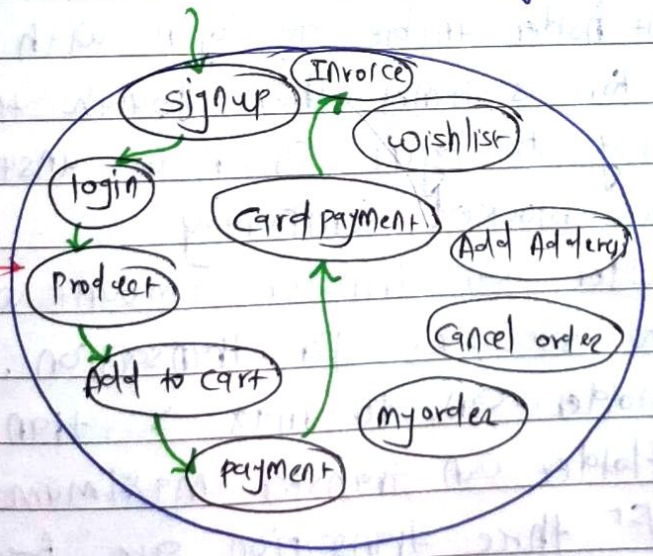
~~End to End Testing~~ →



1) Each Feature functionality tested or not

Check relation betn Feature

Ex Amazon App (to understand sys. testing)



Req → if user purchase using ICICI card will get 30% cashback / discount on Product price.
(we have to check is 30% cashback getting or not)

Product → RS → 1000
ICIC card → 700 (discount)
other card → Rs. 1000

(System testing)

* End to end testing → Navigating through all features and verifying end feature working or not.

* In system testing tester need to check business flows

* Why we should do system Testing ?

→ 1) To check App supports customer business or not
2) to avoid loss in customer business
3) to have confidence on developer developed Product
4) to make sure app support customer requirement or not

* When we should do system testing ?
1) After Functional, integration testing
2) when min. no. of feature should be functionally stable
3) when testing environment is similar to production environment.

Ex) Banking App →
Bss Req - Transaction.
1) If Account holder unable to login with valid UN & PWD consequently for 3 times then fourth time if account try to login which is unsuccess full then Account will be blocked temporary
2) Account holder can transfer amount within a range of 500 Rs to 50000 Rs per transaction.
3) Account holder can do max 5 tran / day.
4) Account Holder can transfer Maximum Rs 200000 /day
5) In a day - 1st three transaction are free and remaining two tran. are chargable
4th → Rs 50
5th → Rs 75

2021/10/21 19:25

21-7-20

Testing enviroment

Environment → setup which will be maintained by the organization to handled software / Application

* it will contain server softward & hardware configuration
* configuration are depends on no of user accept the software

Type of environment

1) Developing environment (Dev) → its a setup which will contain server used by developer to develop

2) Testing Environment (TE) → its a setup which will contain server used by tester to test the app

3) UAT environment (customer / end user) → Acceptance testing

4) Production environment (customer / enduser) → its a setup which will contain server used by end user to use app.

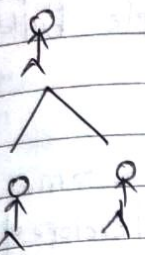
* why test environment should be similar to production environment ?

→ ① chances are there features which are working in testing environment will not work in Production environment. because of configuration changes so we need to maintain testing & production environment configuration same.

② customer will provide environment either at initial stages of testing or in between product testing

classmate
Date
Page

Why testing environment should be similar to production envi

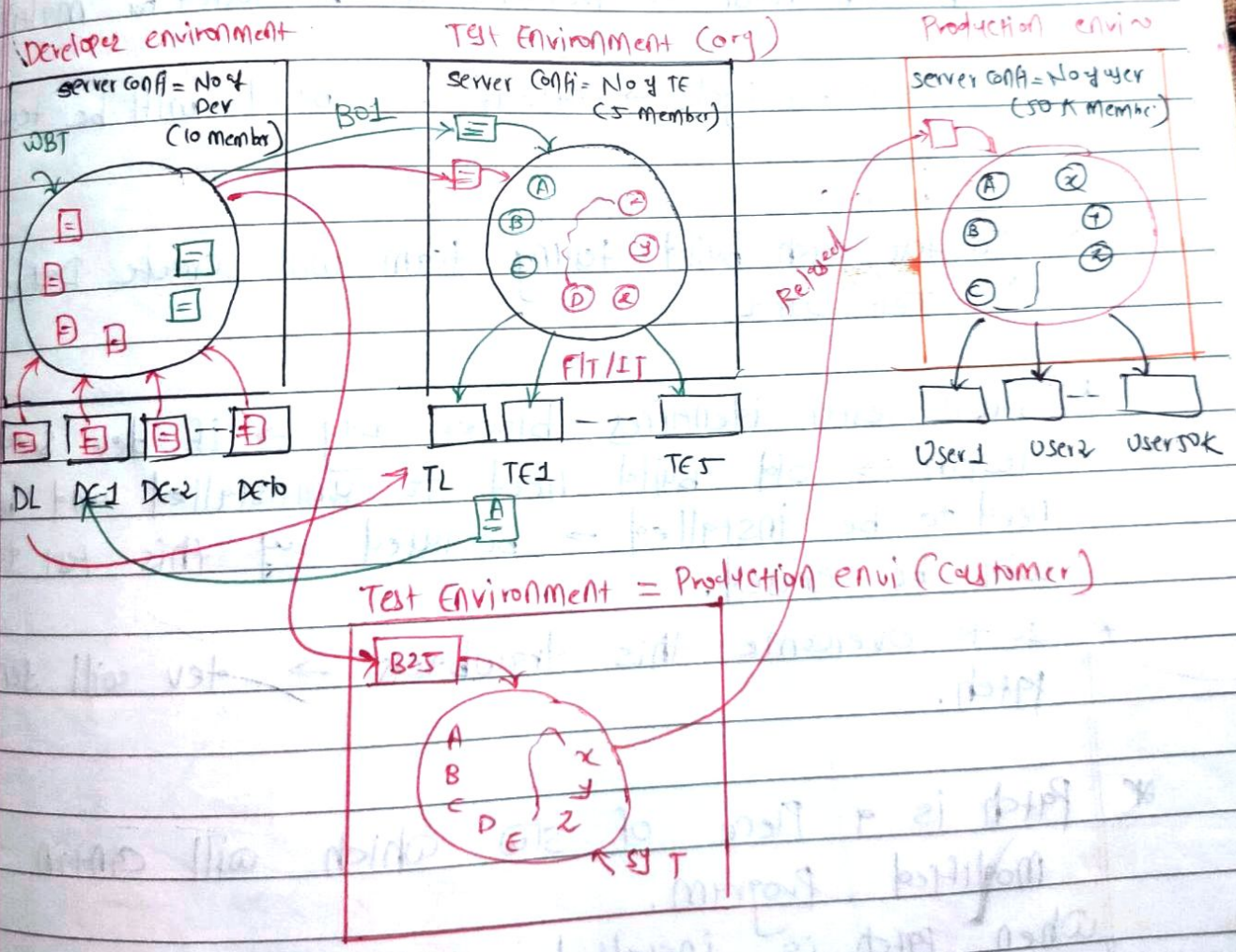


Customer Partners

Req

size 10 member

size 5 member



Acceptance Testing

Testing performed by customer to check developed product quality is known as acceptance testing

1) Why we should do A.T

→ 1) under business pressure project team may launch product with bug, so to identify those bugs customer will do A.T.
2) To avoid loss in business
3) To check app supports business or not.
4) To improve quality of app
5) To check if any new feature need to be added to handle customer business

2) ~~Why~~ When we should do A.T ?

→ one project team release product to the customer, before launching product to production environment

3) How customer will perform AT ?

To do the AT customer will used
a) end user
b) Hire new Tester
c) organization Test team

2021/10/21 19:25

careers@tracxn.com

Paul.Kiruba Karan@zalaris.com

hrd@jsmtechnologies.com

Usability testing →

* Test whether application is user friendly or not
* its a non functionality testing type

What will be tested in Usability →

1. We will App look & feel → theme, background color, Fontsize, logo, Font.
2. We can understand app easily or not ⇒ user should never face problem in using app.
3. App navigations should be quit easier
4. App layout → size of app
5. We need to verify App content visible to user or not.

Ex]

Facebook vs instagram →

1) pic uploaded in App → u need to like that pic
   a) FB → like button is visible
   b) insta → double click on pic

Type of Usability →

1) GUI Testing → in this type we will check app content is graphically aligned properly or not

2) Accessability Testing → in this type we will check app can be useful for physically challanged user or not.