21/7/21

Software Development Life Cycle (SDLC) ⇒ Release

client → BA (Business Analyst)

→ (Requirements for client → Register, Invoice, Online Payment, Products, Order, Tracking etc)

PDC: Requirement Collection (BA)

↓

Feasibility Study (HR Team, Finance Team, Project Manager / Architech)

↓

Design (High Level Desi → Module level → Arch
Low Level Desi → Components level → PM)

↓

ready (defects) Coding (Devs → C, C++, C#, Java - - - →)

↓ (Code-changes)

Testing (Test Engineers) → UI

stable

↓

Installation (Inst^n Team)

↓

Maintenance (LDEV, ITE)

→ whole thing is SDLC ⇒ Release

→ Every app^n → various modules / features
↳ various components / objects

→ S/w / App^n / Product / project / Tool / Build

Scanned with CamScanner

→ SDLC stands for Software Development Life Cycle.

→ It is a step-by-step procedure to develope an appⁿ.

→ It consists of various phases such as requirement collⁿ, feasibility study, Design, Coding, Testing, Installⁿ, Maintenance.

Requirement Collection

→ In this phase the Business Analyst collects the business need of the costumers in the form of doc.

Feasibility Study

→ In this phase a team of high level people sit together and decide whether the project is doable or not, on the basis of some factors i.e → HR team decides the availability of resources.

→ Finance team decides the project is projitable or not.

→ Project Manager and Architech decides the project is technically doable or not.

Scanned with CamScanner

Design

→ Design means blueprint of the appⁿ.

→ Design are of 2 types:

⇒ High Level Design (HLD)

It's also known as module level design. and is done by the Architech

⇒ Low Level Design

It's also known as component level design and is done by the managers.

Coding

→ In this phase the developer starts writing the code by choosing one programming lang.

→ This process of writing the code by devs continue until the appⁿ is ready.

Testing

→ In this phase the test engineers starts checking the ready appⁿ.

→ While checking the appⁿ the test engineers may find some defects / bugs.

→ All these bugs are handed over to the devs.

→ The devs does the necessary code changes and give back to the TEs in the form of new appⁿ.

Scanned with CamScanner

→ This process of finding the bugs by TE and getting fixed by the devs continue until the stable appⁿ is acheived.

Installation

→ In this phase the final stable appⁿ is moved from the company environment to the costumer's environment.

→ It is done by a seperate team called Installation Team.

Mainlenance

→ Once the appⁿ is deployed in the costumer environment, costumer start using the appⁿ.

→ while using the appⁿ the costumers may find some issues.

→ Inorder to overcome such issues and get it resolved immediately we go for mainlenance where → 1 TE and 1 Dev is deployed into the costumer's environment for a certain period of time.

21/7/21

Models o

→ Water

→ Proto

→ Spira

→ v & v

→ Hybr

Agile

Waterfal

Req

Fea

D

co

T

D

N

Rel

→ In

→ We g

Scanned with CamScanner

21/7/21

Models of SDLC

→ Waterfall Model
→ Prototype Model
→ Spiral Model
→ V+V Model
→ Hybrid Model

Agile Methodology

Waterfall Model

Req collⁿ
↓
Fea stu
↓
Des
↓
Cod
↓
Tes
↓
Ins
↓
Main

Rel-1

in b/w changes

Req collⁿ
↓
FS
↓
Des
↓
Cod
↓
Tes
↓
Ins
↓
Main

Next Release (Rel-2) - - - - Rel-n

customer changes:-
→ Market competition
→ Business Need Tes
→ Technology changes

→ In b/w changes not allowed while appⁿ des

→ We go for WM,

→ WM is the base model of SDLC, and all other models are derived from it.

→ In this model, once requirements are collected we freeze it and thereafter execution happens in a sequential order, i.e., the o/p of 1 phase is the i/p for the next phase.

→ In b/w changes given by the costumer are implemented in the next release.

→ No backtracking is allowed.

Disadvantages

→ In b/w changes of requirement are not allowed.

→ No parallel deliverables. (Two phase working simul)

→ Review process doesnot exist.

→ Downward flow of bugs.

→ Dev used to test the appⁿ.

Advantages.

→ It is the best suitable model for life-critical & machine-critical appⁿ.

Protolyp

Pro

Pro

De

→ We

to the

→ In th

appⁿ c

before

the sp

Advanta

→ Costum

→ Protolyp

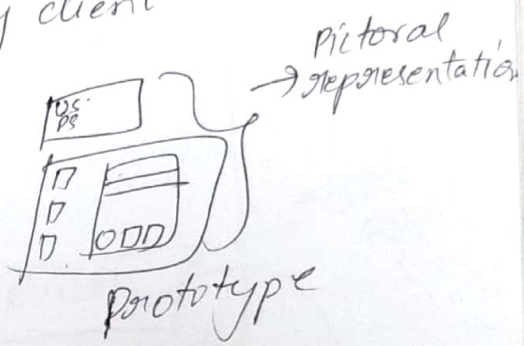
similar

→ In b

Scanned with CamScanner

Prototype Model

Req collⁿ
↓
Fea Stu
↓
Prototype Development → By UI/UX Eng
Prototype Testing → By BA
Review & Approval → By client
↓
Design
↓
Cod
↓
Ins
↓
Main



→ we go for PM, whenever the costumer is new to the software.

→ In this model we develope a replica of the appⁿ and get it approved by the client before we move into actual development of the s/w.

Advantages

→ Costumer satisfaction exists / No cos rejection.

→ Prototype once developed can be reused for multiple similar costumers.

→ In b/w req changes are allowed.

Scanned with CamScanner

Disadvantage

→ Time taking model.

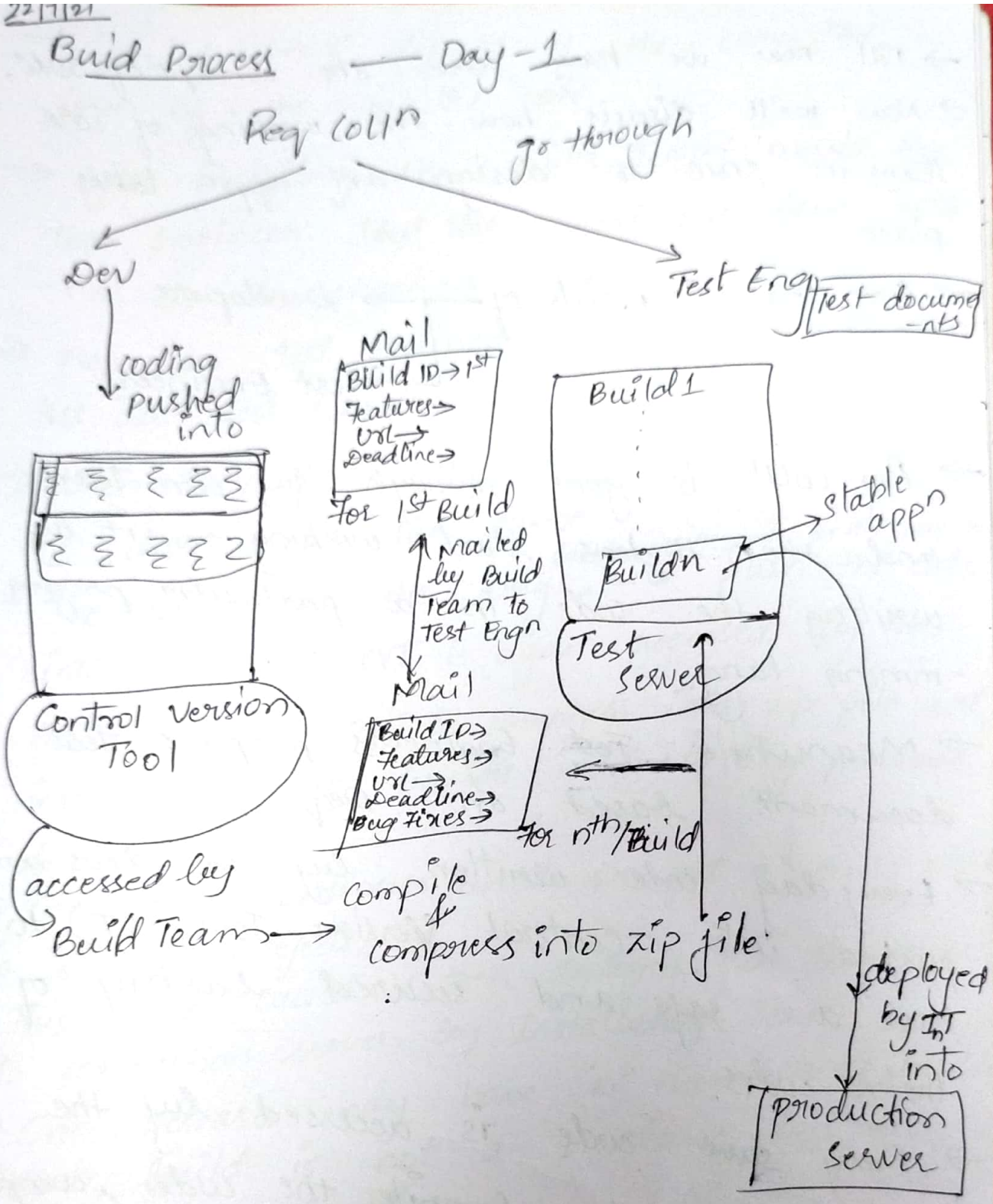
→ No parallel deliverables.

→ Review process doesn't exist.

Scanned with CamScanner

22/7/21

Build Process — Day-1



Build :-

→ It is a piece of code converted in the form of appⁿ. Each build has some features, hence a build can be a part of or whole appⁿ.

Scanned with CamScanner

→ Till now we have seen the concept of SDLC.

→ Now we'll discuss how the working of core team in SDLC to design an appⁿ takes place.

→ Core team consists of → Developers
→ Test Engineers

→ Req Collⁿ is gone through by core team and then the devs start working on it by writing the code in a particular programming lang.

→ Meanwhile Test Engineers prepare test document based on Req Collⁿ.

→ Every day code written by the devs are pushed into Control Version Tool (CVT) to have a safe and secured backup of their codes.

→ Every save code is accessed by the Build Team who compile the code, compress it into a zip file and installs the appⁿ version of the written code in Test Server.

→ Build
form of

→ After
Test En
-lled

→ Now
test doc

→ when
the d

→ The
into
Build
instal
build

→ This
is sta

→ This
to pr

→ One

→ Each
releas

Scanned with CamScanner

→ Build itself defines as the converted form of the written code.

→ After installing the Build Team mails the Test Engineer that the build has been installed to be checked.

→ Now the test Engineer with the help of their test doc test / check the appⁿ.

→ When they find any defects they inform the devs about the same.

→ The devs fix the bugs, besides writing code for other features and push the code into CVT, this CVT is again accessed by the Build Team, compile & compress into zip file and install the next builds uninstalling previous builds.

→ This process continues until we get a build which is stable (no defects).

→ This stable build is then deployed from Test server to production server by Installation Team.

→ One release may have 'n' number of builds

→ Each build is not a release, but each release is a build.

Scanned with CamScanner

Control Version Tool

→ Its a secure tool to store the codes written by the devs.

→ Each dev has access to it with their own credentials.

Build Team

→ It is a dedicated team to convert the code in the form of zip file and instal it to the Test server as well as intimate it to the Test Engineers.

Test Server

just like a cpu.

→ It is the machine where the build is installed.

Note:-

① In some company there is no separate build team. In such case the job of the build team is done by the devs themselves.

② In some companies the zip file is mailed to the test lead. The test lead install it in the test server as well as inform it to the test engineers, via mail.

23/7/21

Bu

10 Bugs — 10 days → 6 + 10B

B001	B002

5 Features

3NF + 4BF

2NF + 5BF

3NF + 5BF

2NF + 5BF

15BF

16BF

⋮

10BF

5BF

3BF

1BF

Test Cycle

→ Test C

particula

→ Sometim

may e

the fu

In o

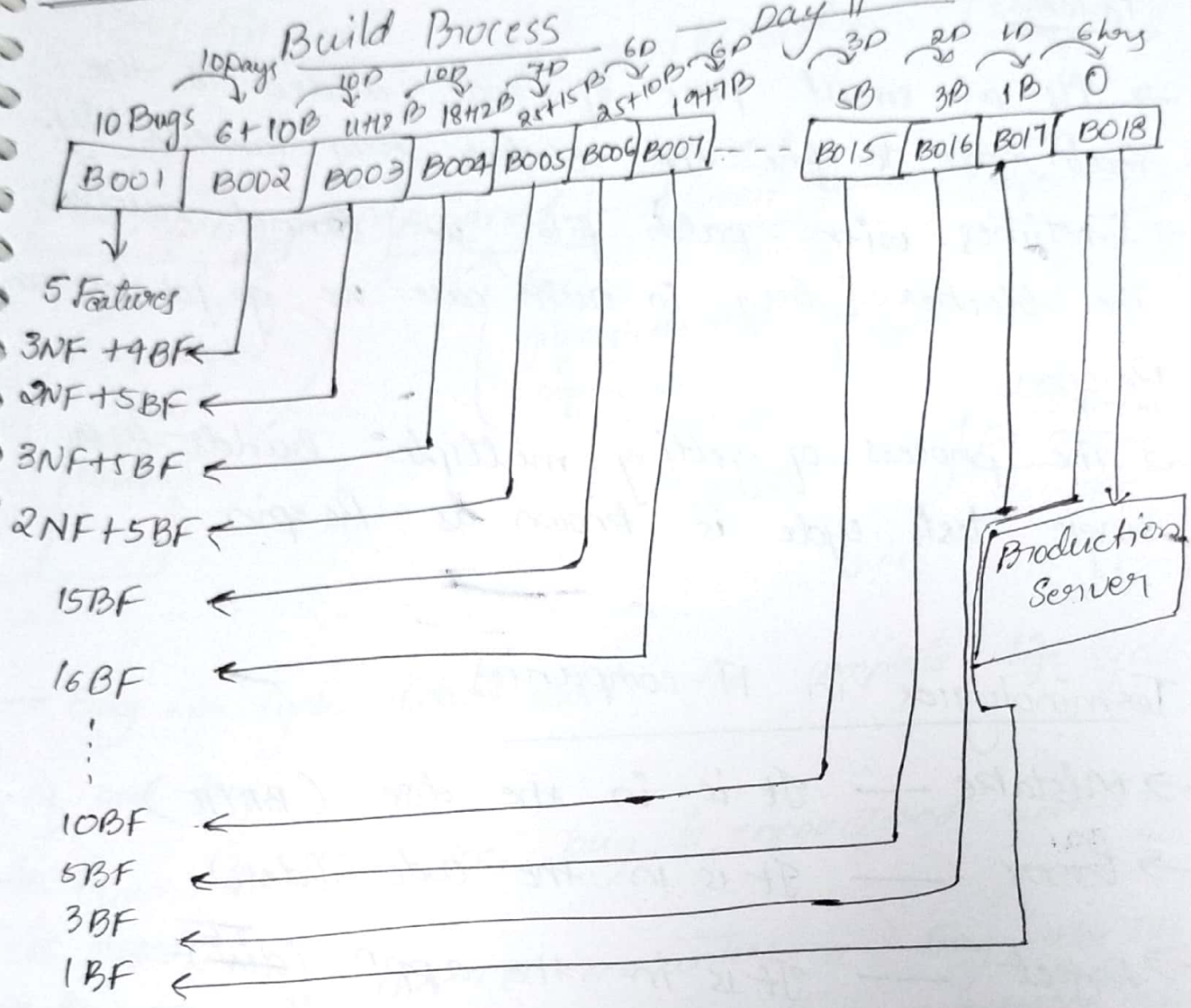
fix the

Patch

Scanned with CamScanner

23/7/21

Build Process — Day 11



Test Cycle

→ Test Cycle is the time duration to test one particular build.

→ Sometime while testing the build the test eng may encounter some blocker bugs which stops the further test execution process.

In order to overcome such situation and fix the blocker bug immediately we go for Patch or Re-spin

Patch

→ Its a small piece of code added to the main code to fix the blocker bug immediately.

→ Sometimes with patch file we cannot resolve the blocker bug in such case we go for Re-spin.

Re-spin

→ The process of getting multiple builds in a single test cycle is known as Re-spin.

Terminologies in IT companies

→ Mistake — It is in the doc (BA/TE)

→ Error — It is in the code (dev).

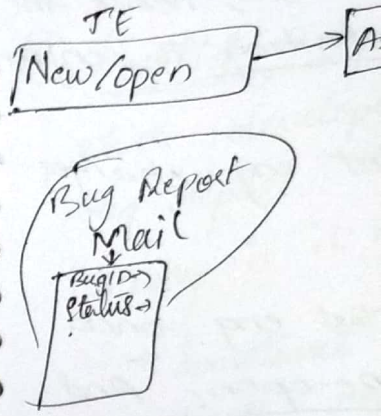
→ Defect — It is in the appn (~~dev~~ TE)

→ Bug — It is informal name of defect (TE)

→ Issue — Defect found by the client.

→ Failure — It is for the project (company) (multiple defects with no solution) (team)

Bug Life Cycle



→ Bug life cycle of one bug.

→ From the tim is closed.

→ As soon as create a bug

→ Initially, the

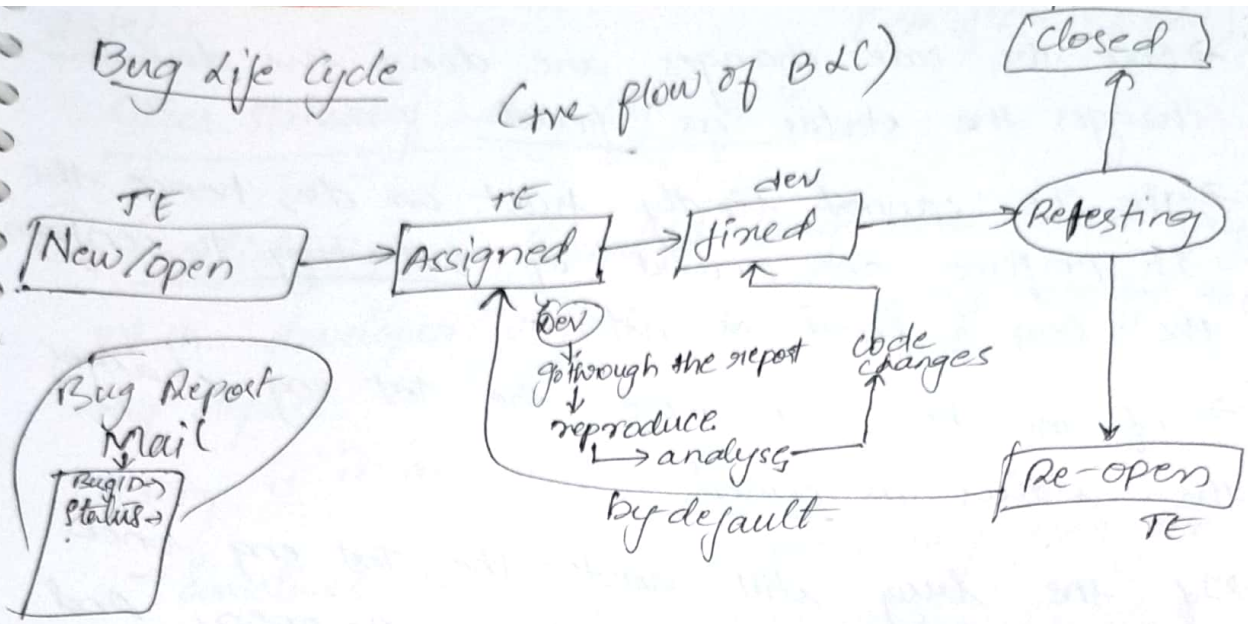
→ The bug rep by changing

→ The concerned read the bug by following analyse and

Scanned with CamScanner

Bug Life Cycle

(the flow of B&C)



→ Bug life cycle talks about the complete life cycle of one bug.

→ From the time the bug is encountered untill it is closed.

→ As soon as the TE encounters a bug, the TE create a bug report with all the necessary info.

→ Initially, the status of the bug is new/open.

→ The bug report is sent to the concern dev by changing the status as assigned. (via mail)

→ The concerned dev go through the bug report i.e., read the bug report line by line reproduce the bug by following the steps written in the bug report, analyse and finally does the code changes.

Scanned with CamScanner

→ Once the code changes are done, the dev changes the status as fixed.

→ The TE cannot blindly trust on dev, hence the TE perfome one round of retesting to confirm the bug is fixed or not.

→ If the bug is no more the test eng changes the status as closed.

→ If the bug still exists the test eng once again changes the status as Re-open. And Re-opened bugs are by default re-assigned to the same dev.

→ This process continue until the bug is fixed and closed.

24/7/21

Other

① Inval

which

bug

* wh

→

* t

→

d

k

*

→

t

⇒ Lif

New

Bu

B
st

24/7/21 (-ve flow of BLC)

Other statuses of Bug Life Cycle

① **Invalid Bug** :- Invalid bugs are those bugs which developer reject after going through the bug report.

* why is it an Invalid Bug Report?

→ Sometimes due to misunderstanding over the client's requirement.

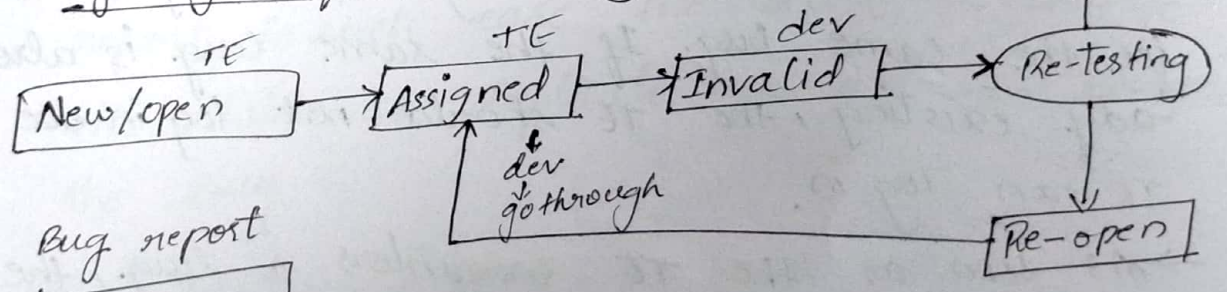
* How to avoid Invalid Bug?

→ In order to avoid Invalid Bug both dev and T.E should have very good product knowledge.

* Can we re-open IB?

→ IB can be re-opened if the dev has misunderstood the requirement and changes the status as Invalid.

⇒ Life Cycle Of Invalid Bug Cycle



Bug report

Bug ID →
status →
observ →

② **Duplicate Bug** :- when the same bug is logged by TEs for more than one time, for the 1st time the devs will accept but from 2nd time onwards the dev will reject it by changing the status as 'Duplicate Bug'.

* Why Duplicate Bug Occurs?

→ a) when the same module is assigned to multiple TEs.

b) Two TEs working on the dependent module.

c) Bug present in common module.

NOTE :- Duplicate Bugs are bad name for the TEs.

* How to avoid the Duplicate Bug?

→ All dependent modules should be assigned to the same TE (which is practically impossible).

→ As soon as the TE encounters a bug, the TE should go to the 'Bug Repository' search for the same bug. If the same bug is already existing, the TE should not log in else TE can log in.

→ As soon as the TE encounters a bug, the TE should go to 'Bug Repository' search for

the same
present in
by keepi

* Can we

→ Someti
but no
TE can

Eg:- att

If TE
dev fix
again [i.
do look
same].

→ If t
in the
again
definet
report
the
the

Eg:-

Scanned with CamScanner

the same bug. If the same bug is not present in the repository, the TE should begin by keeping the other TEs in the loop.

* Can we re-open the DB?

→ Sometimes the two bugs looks the same (alike) but not exactly the same. In such case the TE can re-open the duplicate Bug.

Eg:- attaching PDF, attaching xls

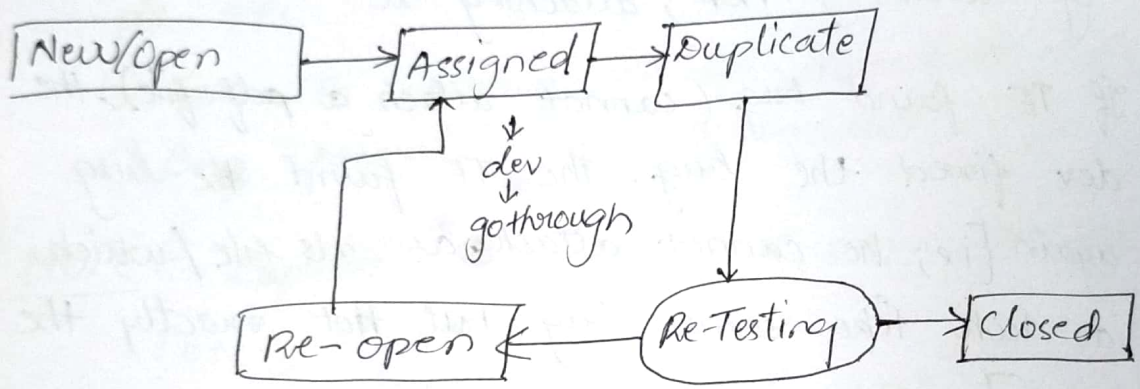
If TE found bug (cannot attach a pdf file), the dev fixed the bug. The TE found the bug again [i.e, he cannot attach as xls file / which do look like same bug but not exactly the same].

→ If the bug ~~looking~~ looked previously is already in the status of closed, in such case the TE again encountering the same bug ~~which~~ will definetely report to the developer in a new report, in such case the developer change the status as duplicate the TE can re-open the bug.

Eg:- B005 - - - B009 - - - B014

The bug is encountered in build 5 and closed in build 9. Again the same bug is encountered in build 14. The TE can re-open a bug to dev in a new report, in such case the dev change the status as dupli-cate the TE can re-open the bugs.

⇒ Life Cycle of DB



26/07/21

③ Deferred Bug :- Deferred Bugs are those bugs which are accepted by the devs but the devs postpone it for the next release.

* why DB occurs?

→ DB are due to the lack of time/time constraint.

* How to
→ Practic
bug may
the last
soon as
to the
enough t

* Can we
→ we ca
major

NOTE :-
① All DBs
are not

* Life cycle

TE
New/Open

Bug Report

Scanned with CamScanner

* How to avoid RB?

→ Practically, we can't avoid deferred bug since bug may appear due to the code changes in the last few builds but as the TE as soon as encounters a bug must report it to the dev, so that the dev will have enough time to fix it.

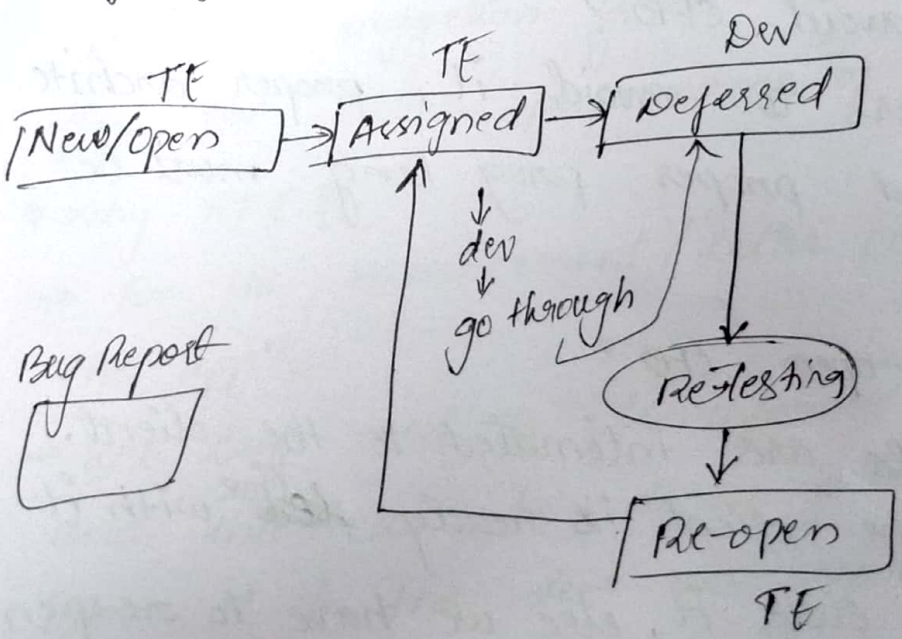
* Can we re-open RB?

→ we can re-open a RB if the bug is a major bug.

NOTE:-

① All RBs are minor bugs, but all minor bugs are not RBs.

* Life cycle of RB:-



④ **Can't fixed:-** Can't fixed bugs are those bugs which, when are accepted by the dev tried to be fixed would effect the whole appⁿ and the appⁿ might mis behave, hence the dev changes that bug's status as can't fixed.

* Why CFB occur?

→ CFB are due to the poor architecture or limitation of prog lang.

→ Sometimes, these bugs are in the archite-cture of the appⁿ & if such bugs are fixed the entire appⁿ may mis-behave.

* How to avoid CFB?

→ In order to avoid it, proper archite-cture and proper prog lang must be choosen.

* Can we re-open CFB?

→ All CFBs are intimated to the client. And if the client is ready ~~alive~~ live with it we can close it, else we have to re-open.

and d
irrespectiv

* Life Cyc

TE

New/open

Bug Re

BID
Status
steps to rep
observatio

⑤ Requ

It is
for

* Why

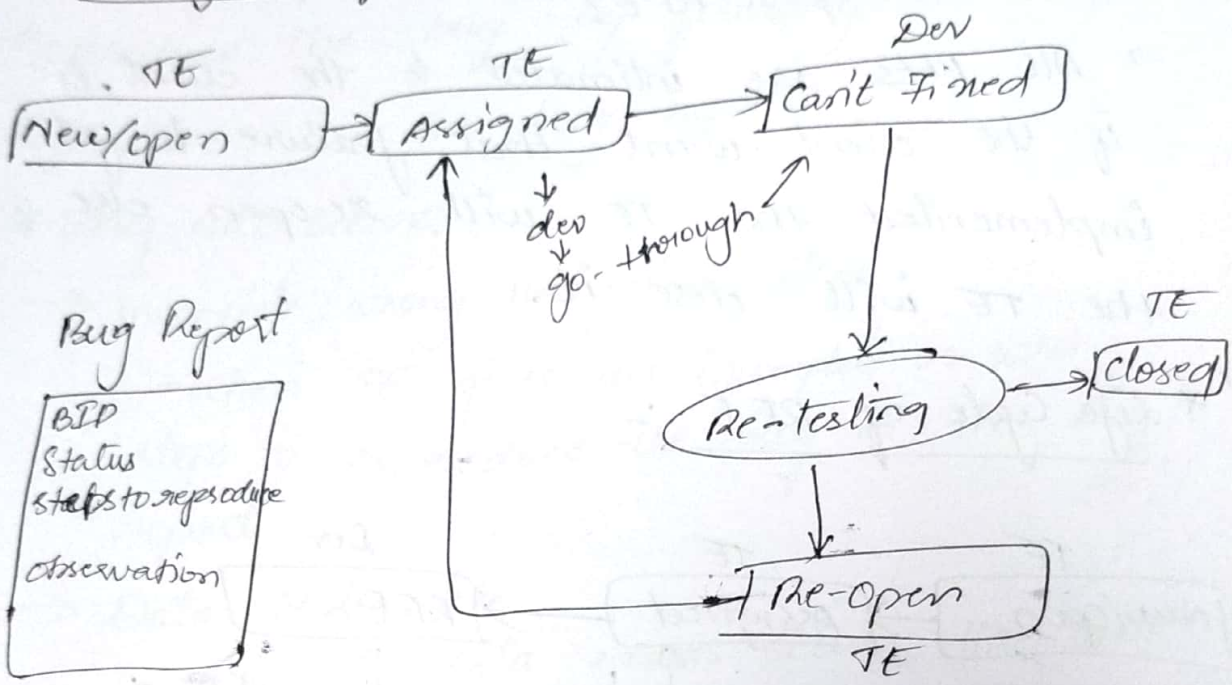
→ For
the

→ RFE
and
inform

Scanned with CamScanner

and devs have to do the code changes irrespective of the consequences.

* Life Cycle of CFB :-



⑤ Request for Enhancement (R.F.E) Bug :-

It is a suggestion given by the Test Eng for the better performance of the appⁿ.

* Why RFE ?

→ For the enhancement / better performance of the appⁿ.

→ RFE bugs need to be informed to the devs, and this can be done once the TE inform it to the deve via report.

* How to avoid RFE?

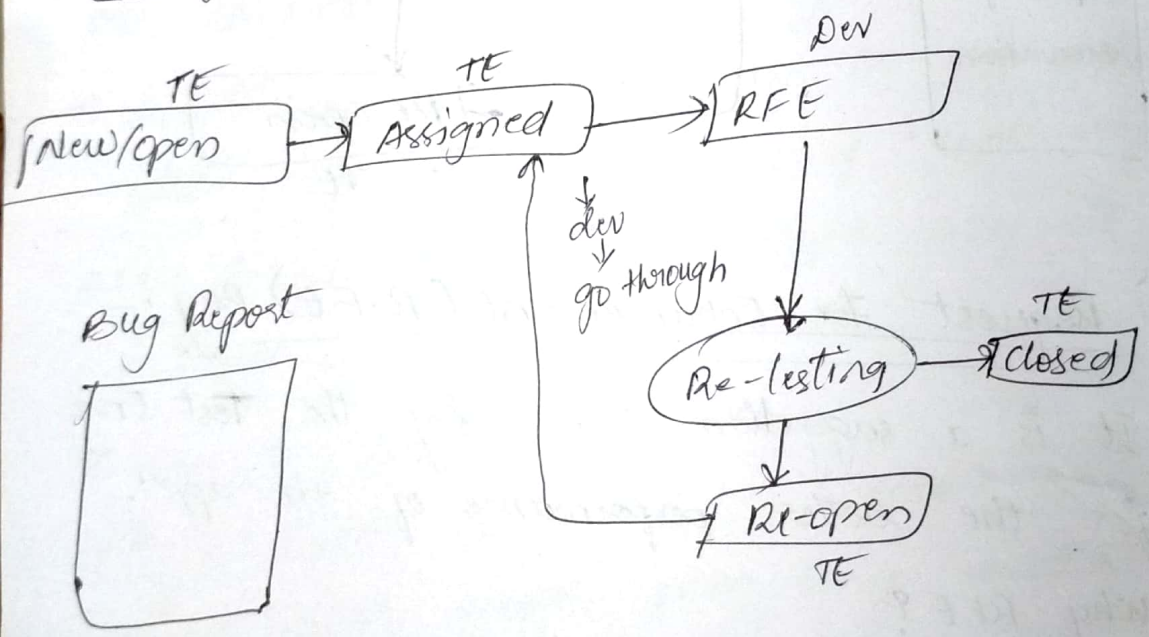
→ We have to provoke RFE.

→ We should not avoid RFEs.

* Can we re-open RFE?

→ All RFEs are intimated to the client, & if the client want that feature to get implemented the TE will re-open else the TE will close it.

* Life Cycle of RFE :-



NOTE :-

① Practically it's not the mistake of dev, but the dev have to do the code changes to fix it, hence it is included in bug life cycle.

27/07/21

⑥ Not-Rep

are accept

the bug

reproduce

reproduce

* why NR

→ Incomp

- wh

steps

repo

→ Data

-

an

is

→ Serve

-

Se

b

→ Plat

-

-

21/01/21

(6) **Not-Reproducable** :- NRBs are those bugs which are accepted by the dev after going through the bug report but the dev is unable to reproduce the bug by following the steps to reproduce in the bug report.

* Why NRB occurs?

→ Incomplete / wrong navigation steps
- when TE gives an incomplete or wrong steps to reproduce the bug in the bug report.

→ Data Mismatch
- When i/p data errors occurs, like take an example of FB messenger when user id is correct by mismatch in rec id occurs.

→ Server Mismatch
- when TE encounters a bug in Test server but dev tries reproding the bug in dev server or so.

→ Platform Mismatch
- when TE encounter the bug in UNIX OS, but dev did write code and was appli-cable with no errors in windows OS.
- when encountered bug was in chrome but was fine w.r.t microsoft edge / firefox.

Scanned with CamScanner

→ Build Mismatch

- when a bug was encountered in build 4 but due to some code changes it got automatically fixed at build 10.

→ Inconsistency

- when a bug encountered appears during few execution times but doesnot appear most of the time.

* How to avoid NRBs?

→ In order to avoid NRBs, the TE should prepare a proper bug report with all the necessary information.

→ Inorder to avoid inconsistent bug which is practically impossible but as a TE we should take a screenshot of the bug as soon as we encounter it.

* Can we re-open NRBs?

→ Only in build mismatch case the NRB can be closed, if not it has to be reopened.

* Life Cycl

TE

New/Open

Bug Repo

Process:-

→ Bug

-tory

Final Di

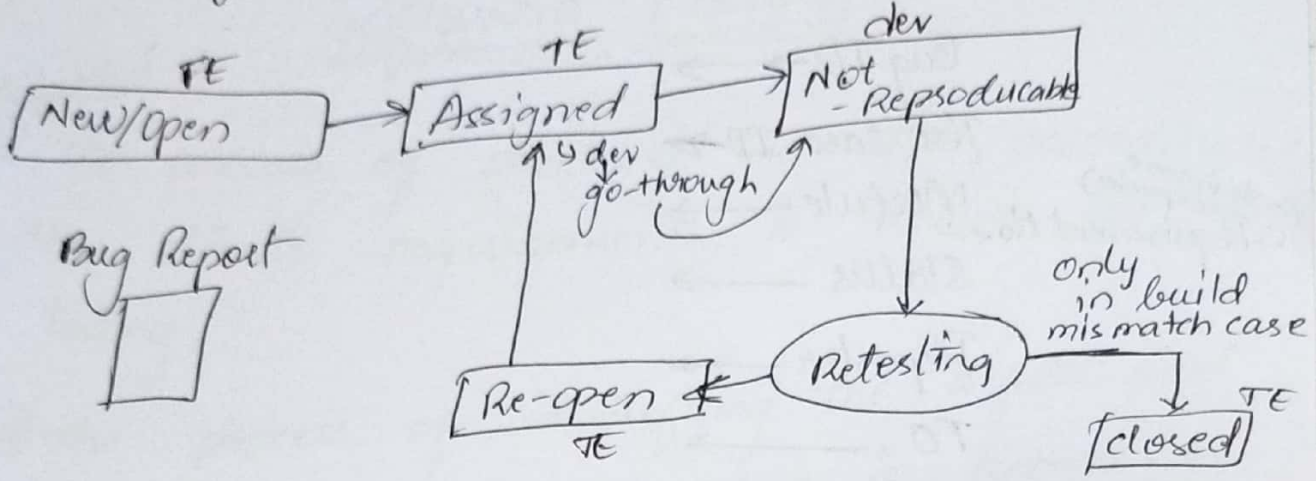
TE

New/Ope

Bug Re

Scanned with CamScanner

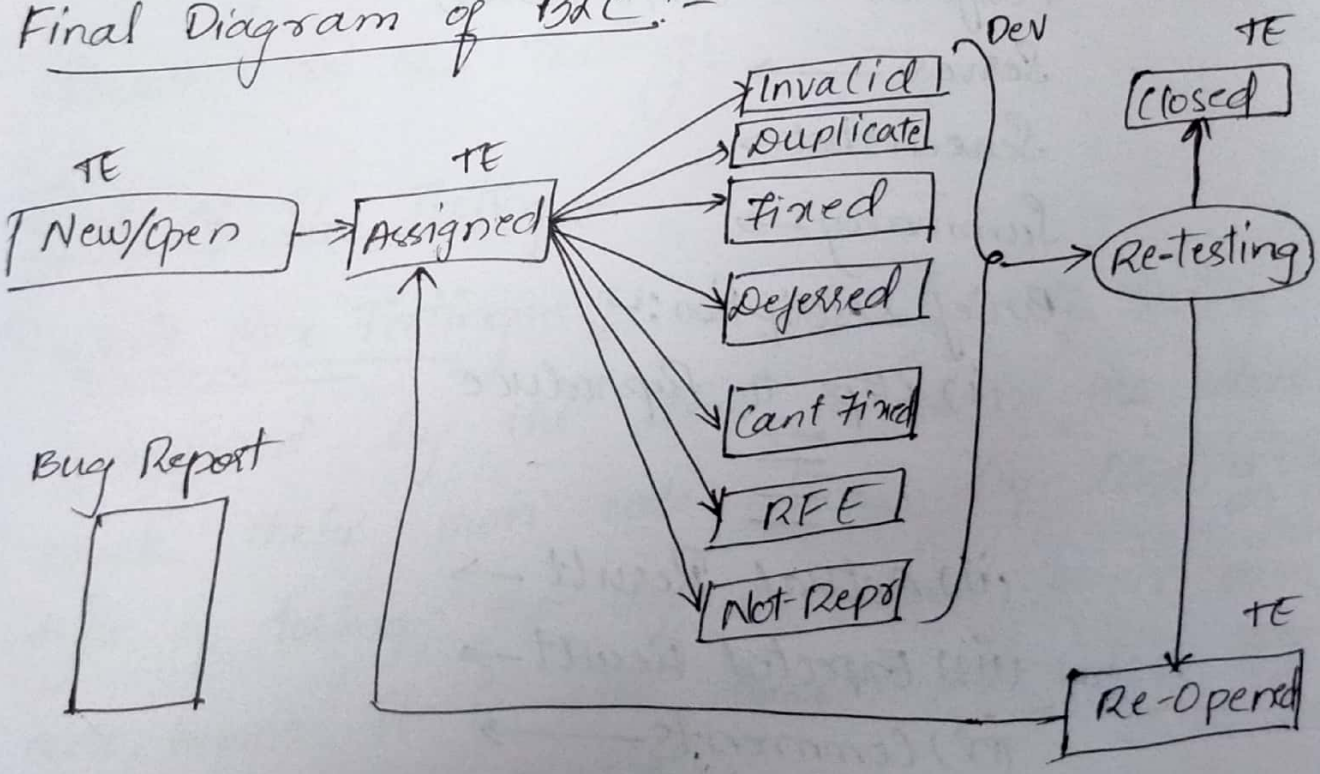
* Life Cycle of NRBs :-



Process :-

→ Bug encountered → Take SS → check bug reposi-tory
- → if bug exists (no need of BR)
- → if bug not there / closed (BR is done)

Final Diagram of BLC :-



Scanned with CamScanner

Bug Report Template :-

Bug ID →

Test Case ID →

Module →

(only in some companies) ↳ Requirement No ←

Status →

Reporter →

TO →

CC →

Date →

Severity →

Priority →

Test Data →

Build ID →

Platform → (OS/browser)

Server →

Screenshot →

Summary →

Brief Description :-

(i) Steps to Reproduce

(ii) Actual Result →

(iii) Expected Result →

(iv) Comments →

Scanned with CamScanner

28/07/21

Software Testing

definitions

① The process of checking the appⁿ, according to the client's requirements is known as s/w testing.

② The process of identifying the defects and get it fixed by the devs is known as s/w testing.

③ The process of using the appⁿ as a user until the appⁿ is stable is known as s/w testing.

④ The process of verification & validation is known as s/w testing.

Types of S/w Testing:-

① **White Box Testing:-** It is a type of testing performed by the devs in which the devs check their own code line by line. In this type of testing the dev can see their own code, hence it is also known as Glass Box Testing/Transparent Testing.

Scanned with CamScanner

NOTE :- White Box Testing is not responsible to make the appⁿ stable. It is done by the devs to reduce the count of errors in the code. A small error in the code can project as a critical bug for the Test Engineer.

② Black Box Testing :- It is a type of testing performed by the TEs, in which the TE check the appⁿ to find the defects in the appⁿ. In this type of testing the code is not visible to the TE. Hence it is also known as Closed Box Testing.

NOTE :- Black Box Testing is responsible to make the appⁿ stable.

③ Gray Box Testing :- In this type of testing, the same set of people will check the code as well as the appⁿ. It is the combination of white Box Testing and Black Box Testing.

NOTE :- It is the bad approach since checking the codes as well as appⁿ by the same set of people may causes lack of accuracy.

Scanned with CamScanner

White Box Testing Process

Requirement Doc	Business Logic	Unit Test Program
1. Login	1. Code for Login	1. UTP for Login
2. Statements	2. Code for statement	2. UTP for statements
3. Balance Enq	3. Code for Balance Inq	3. UTP for B.I
4. Amt Transfer	4. Code for A.T	4. UTP for A.T
⋮	⋮	⋮
n. Logout	n. Code for Logout	n. UTP for Logout

→ UTP written in few tools such as:
- TestNG → open source tool
- Jmeter
- Nmeter

→ Manually checking line by line would take 4-5 hrs but through

→ WBT is always done with the help of tools since manually checking the code line by line may causes lack of accuracy as well as ~~it will~~ will consume a lot of time.

Some commonly used WBT tools are:

- testNG
- Jmeter → Junit & Nunit
- Nmeter etc...

→ Unit Test Program :- It is a small peice of code written by the dev which inreturn check for the main business logic.

Advantages of UTP:

- UTPs can be reused accross multiple releases.
- It saves the time by ~~giving~~ faster checking the code.

29/07/21

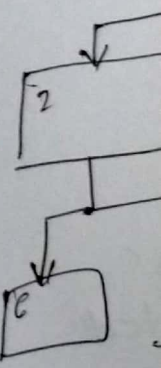
→ WBT is also known as Unit Testing since in this type of testing the ~~Test Engineer~~ Dev use UTP to test the code line by line.

29/07/21

→ Types of

1. Path
2. Loop
3. Cond
4. Me
5. Re

1. Path T



→ In

for the

ways

block

branc

furth

use e

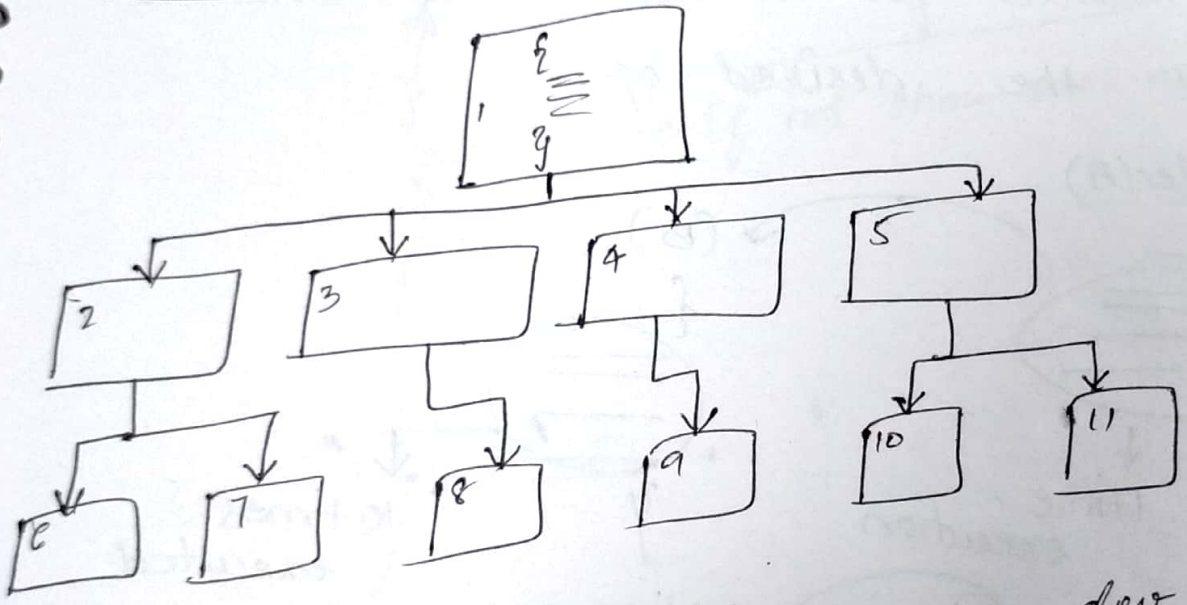
Scanned with CamScanner

29/07/21

→ Types of WBT:-

1. Path Testing
2. Loop Testing
3. Condition Testing
4. Memory Testing
5. Response Time Testing

1. Path Testing



→ In this type of WBT, we as a dev check for the execution of the code in various ways to get the desired output, i.e, one block of code may be linked with multiple branch code. Those branch code may have further sub-branch and in this type of testing we ensure each code & branch code are getting

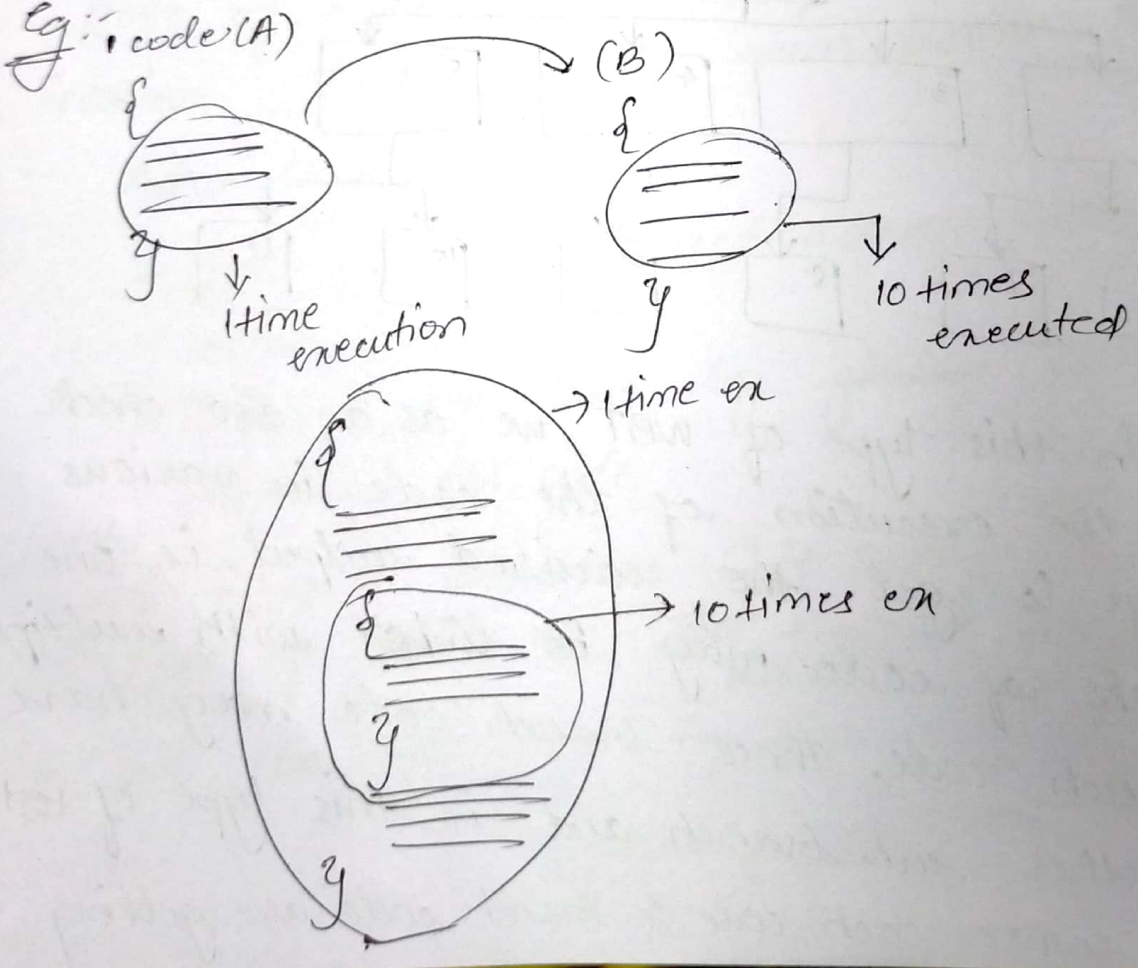
executed whenever required.

Eg:- 1, 3, 8
1, 2, 7
1, 5, 11 etc...

2. Loop Testing

→ In this type of WBT the dev check for the execution of the code, for 'n' number of times to retrieve the desired o/p i.e., in a code a particular section of that code may get executed for 'n' number of times to retrieve the desired o/p.

Eg:



3. Conditio

→ In this
executio
conditi
each l
- ted.

Eg:-

4. Memo

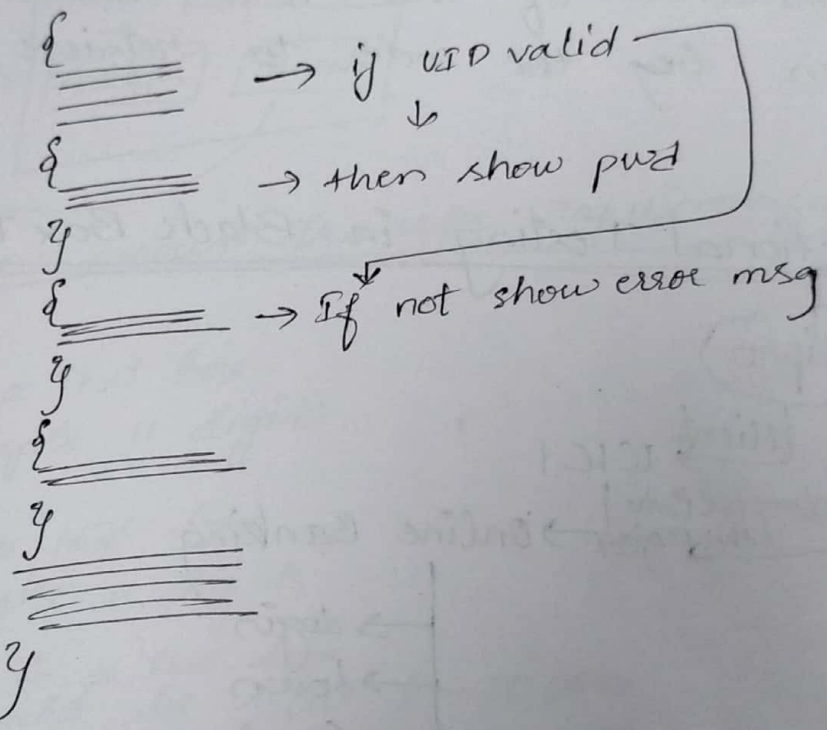
→ In t
the
o/p, i.e
code
and

Scanned with CamScanner

3. Condition Testing

→ In this type of WBT the dev ensure the execution of the code according to the condition, i.e., everytime we execute ~~our~~ the code each line of that code maynot get execu-ted.

Eg:-

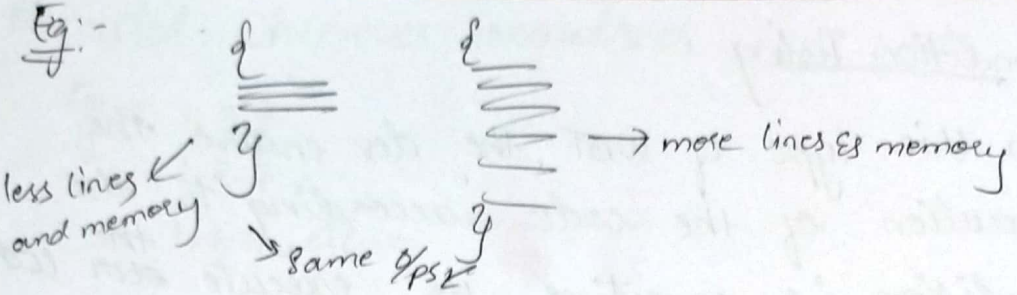


4. Memory Testing

→ In this type of WBT the dev checks for the consumption of memory to get the desired o/p, i.e., as a dev we should ensure that the code has followed all the coding standards and has consumed least memory.

Scanned with CamScanner

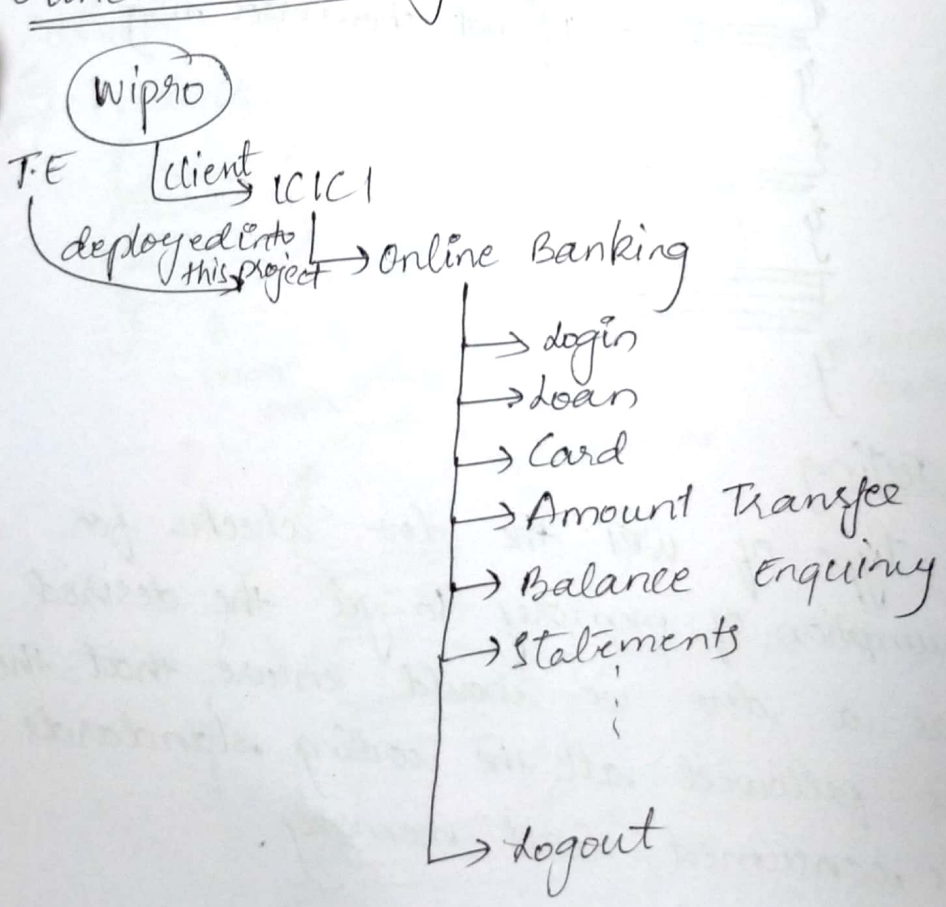
Eg:-



5. Response Time Testing

→ In this type of testing we dev check for the performance of the code that is the time taken by the code to retrieve the desired o/p.

Functional Testing in Black Box Testing:-



Imagining

Test the

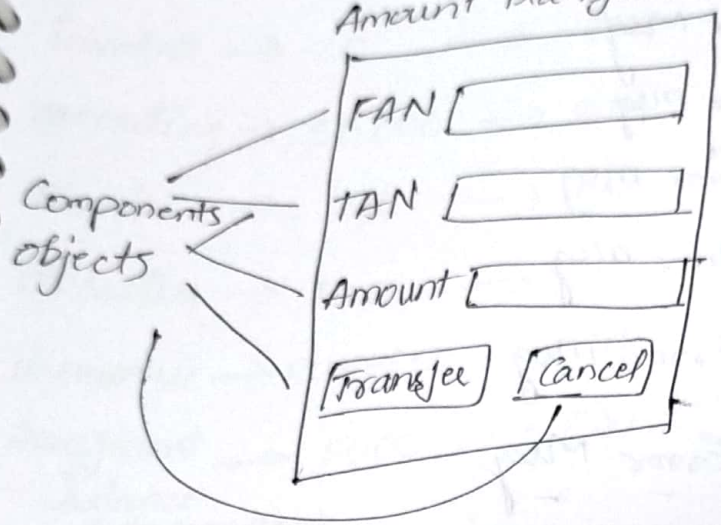
Components
objects

4th Modul

4.1 FAN
4.1.1 FAN
4.1.2 It a
4.2 TAN
4.2.1 TAN
4.2.2 It a
4.3 Amou
4.3.1 Amou
4.3.2 It
4.4 Tran
4.4.1 Tr
4.4.2 It
4.5 Can
4.5.1 Can
4.5.2 It

Scanned with CamScanner

Imagining myself as a T.t who is given to Test the Amount Transfer Module.



4th Module: Amount Transfer requirements

4.1 FAN
4.1.1 FAN is a text box
4.1.2 It accepts 11 digits
4.2 TAN
4.2.1 TAN is a text box
4.2.2 It accepts 11 digits
4.3 Amount is a text box
4.3.1 Amount should be b/w 1 – 50000
4.3.2 It
4.4 Transfer
4.4.1 Transfer is a button
4.4.2 It should be enabled
4.5 Cancel
4.5.1 Cancel is a button
4.5.2 It should be enabled

Scanned with CamScanner

I/ps for PAN (various combns)

Valid → 1111111111 → Accepted

Invalid → 111111111 → Error Msg

Character → sahr → Error Msg

Special Characters → @*/_ → Error Msg

α-numeric → ab111$ → Error Msg

more than 11-digits → 1111111111 → Error Msg

Blank → N/A → Error Msg

A/c doesnt exist → 1124111981 → Error Msg

I/ps for TAN (various combns)

Valid → 22222222222 → Accepted

Invalid → 22222 → EM

Character → Arund → EM

Special character → !$_/1 → EM

α-numeric → cd111@ → EM

more than 11-digits → 222222222222 → EM

Blank → N/A → EM

A/c doesnot exist → 4116237450 8 → EM

Invalid TAN → 1111111111 → EM

Scanned with CamScanner

I/P for Amount

valid → 100 → accepted

invalid → 0 → EM

exceeding → 55000 → EM

Blank → N/A → EM

characters → FIVE → EM

α-numeric → RUPEES50 → EM

insufficient balance (only 2000 there) → 5000 → EM

Daily Limit → 200 → EM

30/07/21

→ Checking each and every module as well as components of a module individually or 1 by 1 is known as Functional Testing.

→ In order to perform FT we have to follow some rules:

Rule 1 :- Always start testing with valid i/p, and then continue with all other comb^n of invalid i/ps.

Rule 2 :- while checking one component, the i/ps for all other components should be always valid. Inorder to locate where exactly is the defect.

Scanned with CamScanner

Rule 3 :- Donot check for similar type of i/ps, since similar type of i/ps may causes over testing which is an endless process & time consuming.

Rule 4 :- while performing the testing process we create some data hence delete should be the last action, else we have to create the data again and again.

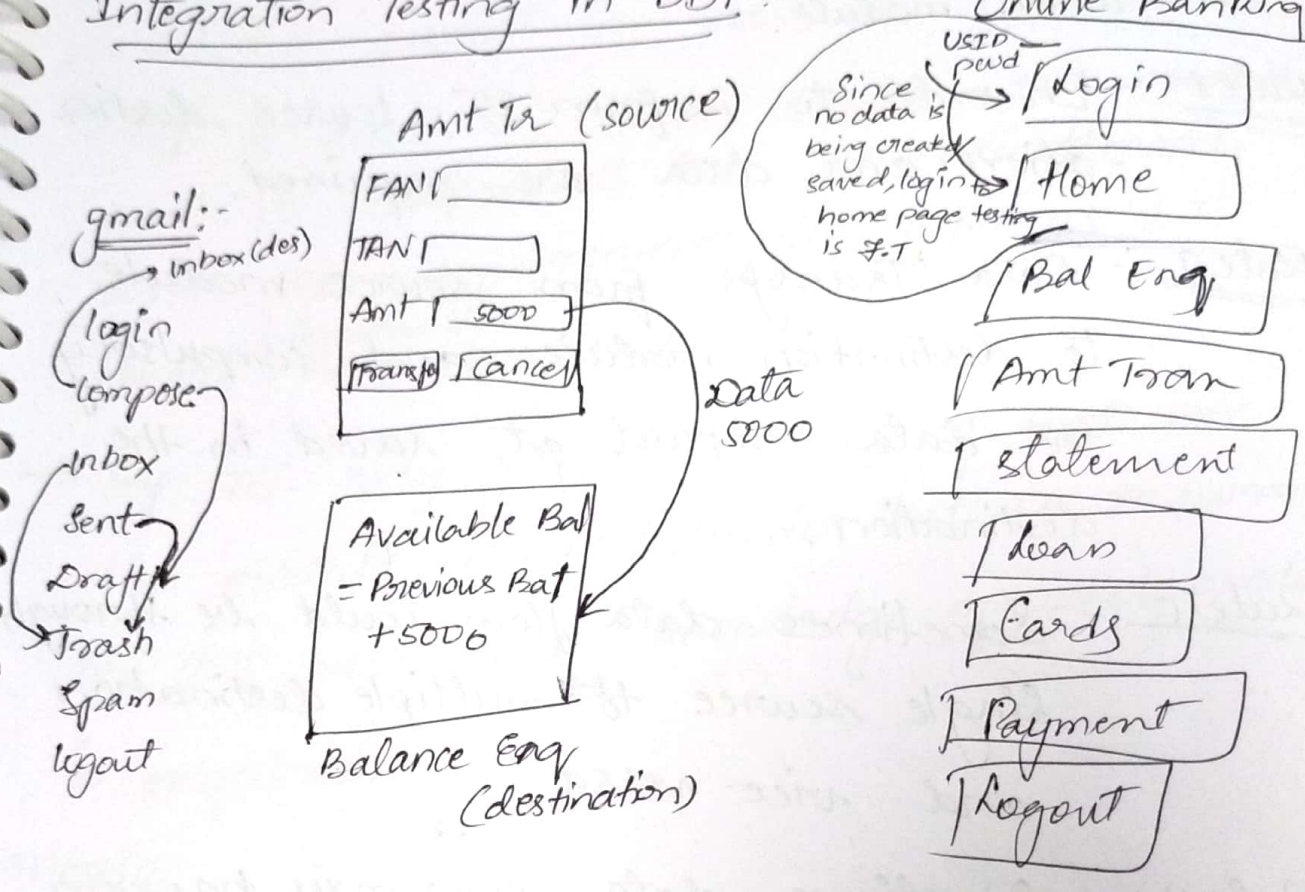
NOTE : ① The toughest part during the functional testing is to decide when to stop testing, & this decision is taken by the T.E once they feel the max test coverage is acheived.

② once each and every module along with its components are checked with various combⁿ of i/ps, we can say functional testing is over for the complete appⁿ.

Scanned with CamScanner

30/07/21

Integration Testing in BBT :-



Rule 1 :- As soon as we get an appn, we go for F.T.

Rule-2 :- F.T is done in every individual module.

Rule 3 :- Once F.T is over we go for I.T.

Rule 4 :- I.T is done b/w dependent modules.

Rule 5 :- IT is not done for some modules.

Rule 6 :- In order to perform I.T, very good product knowledge is required.

Scanned with CamScanner

Rule 7 :- I.T is done b/w source module to destin-ation module.

Rule 8 :- In order to perform IT, source, destin-ation and data are required.

Rule 9 :- Data Transfer from source module to destination module and compulsory the data should get saved in the destination.

Rule 10 :- Sometimes data flow could be through single source to multiple destination and vice-versa.

Rule 11 :- Sometimes data flow may happen within the module, hence I.T is done within the module too.

Interview

Difference b/w F.T and I.T :-

→ F.T is done for every individual module, I.T is done ~~for~~ b/w dependent module, hence I.T maynot be done for some modules.

→ F.T doesnot require source / destination, I.T is actually done b/w source / destination.

Scanned with CamScanner

→ Data created is not saved in F.I but in I.T it is saved in the destination module.

→ F.I is done to check whether each module and its components are according to match with the requirement doc, whereas I.T checks ~~how~~ interaction b/w two modules.

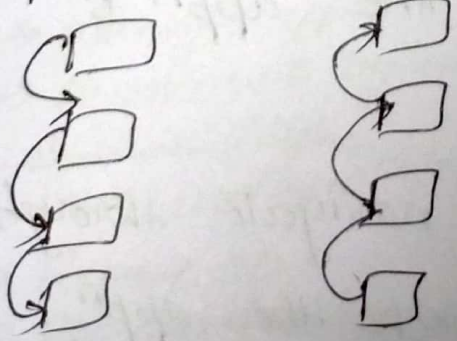
→ For F.I very good knowledge of ~~that mo~~ the module assigned / under test is required. For I.T very good knowledge of whole product is required.

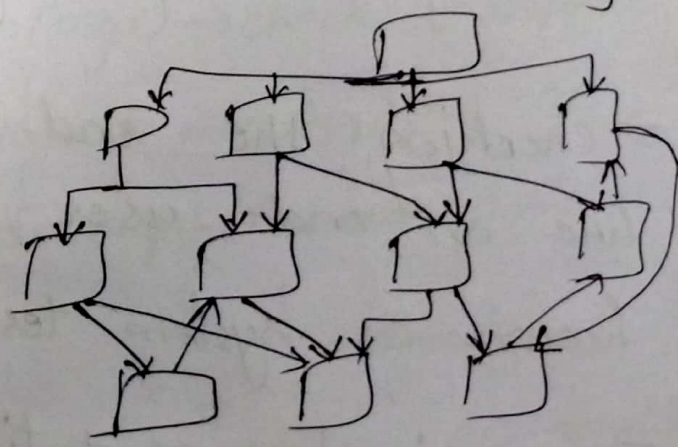
31/07/21

Types Of IT :-

I.T (Integration Testing)

- Incremental Intergration Testing (I.I.T)
  - Top-to-Bottom
  - Bottom-to-Up
- Non-Incremental Integration Testing (N.I.I.T)





P.I.T :- In this type of I.T, data flow in a sequential order from one module to the next module. It is of two types:

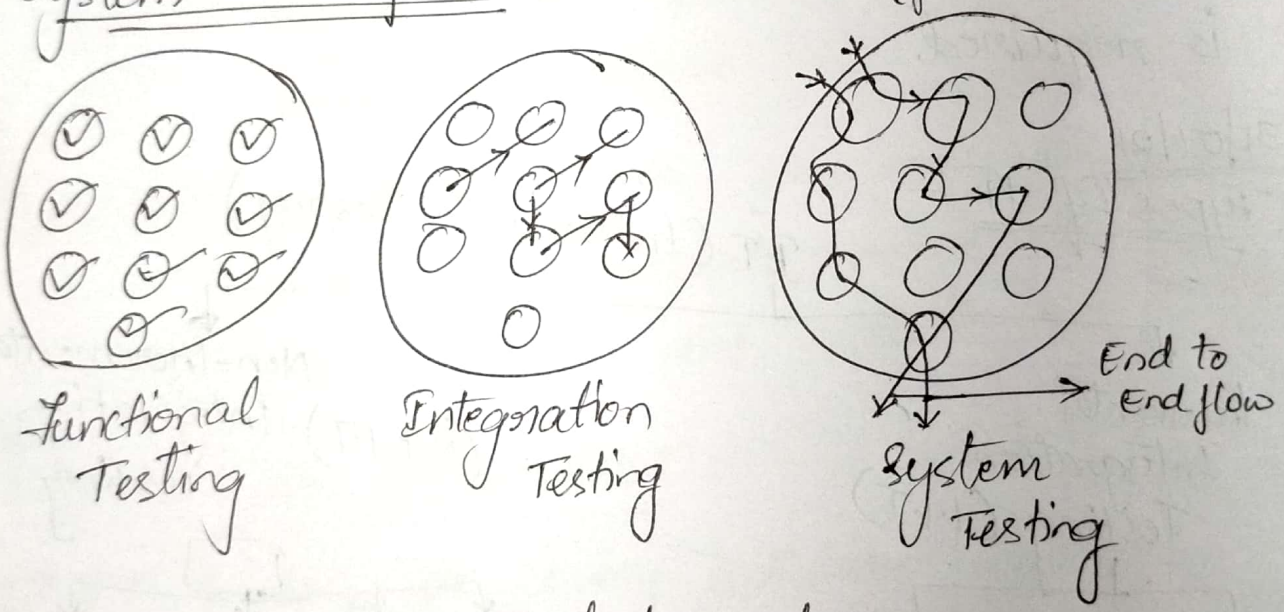
- Top to Bottom
- Bottom to Up

N.I.T.T :- In this type of I.T, data flow random-only b/w any two modules without following any sequence or pattern.

31/07/21

System Testing in BBT :-

Eg:- Gmail compose send logout
Gmail inbox logout



→ checking the end to end flow of the appⁿ like an end user, using the appⁿ is known as system testing.

→ In this type of testing we navigate through all necessary modules to check the appⁿ,

as a whole system.

NOTE :-

(i) All dependent modules should be assigned to the same T.E.

(ii) One appⁿ may have 'n' number of modules, each module may have 'n' number of sub-modules. And each sub-module may have multiple components.

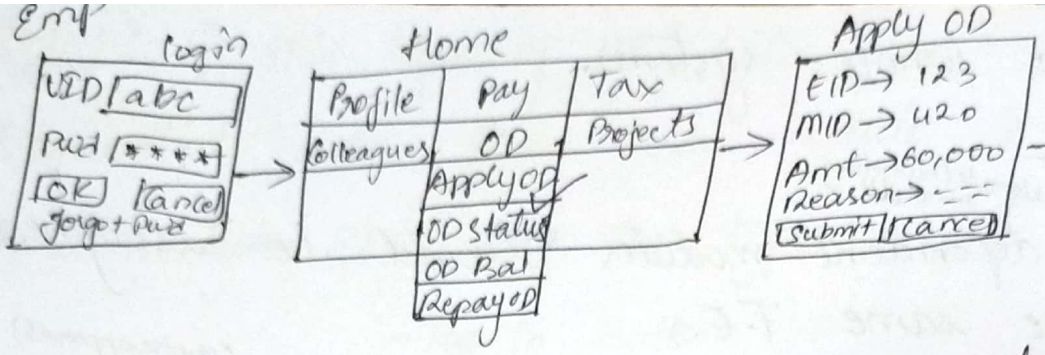
(iii) Every TE perform functional Testing, Integration Testing and system Testing on their assigned module.

System Testing on Over Draft Module :-

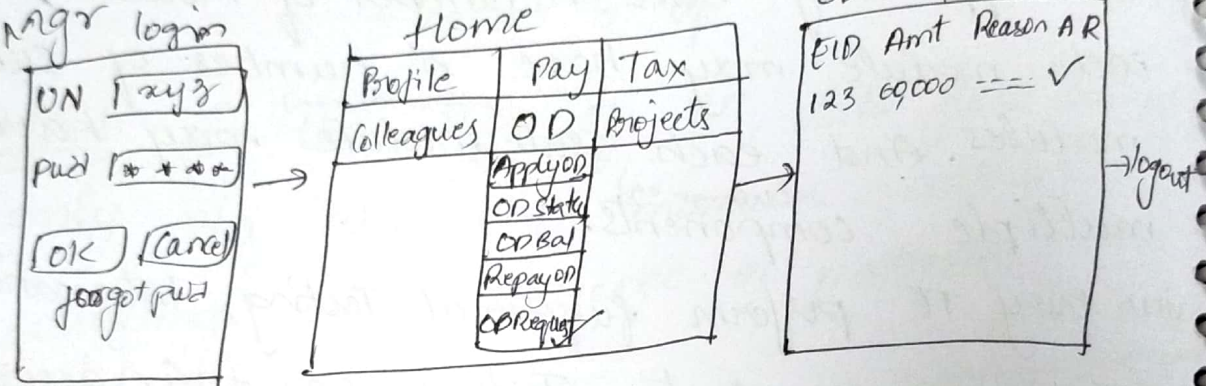
ERP (domain) → Over Draft (Module)

Requirement	Scenarios
→ 2 months salary in advance.	1. Login (emp) → Apply OD → logout
→ Apply OD. (by emp)	2. Login (mgr) → check OD requests ↓ Approve OD → logout
→ OD request is sent to Manager.	3. Login (emp) → check OD status ↓ logout
→ Manager approves OD.	4. login (emp) → Repay OD → logout
→ OD received by emp ~~after 10 days.~~	
→ Amt credited	
→ Repay OD	
→ Processing charge 2.5%	
→ 3 months within no interest	

1. Emp

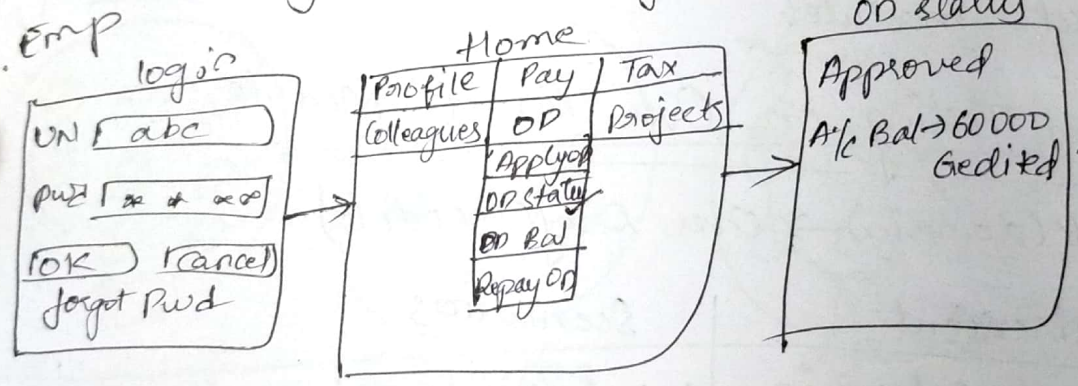


2. Mgr

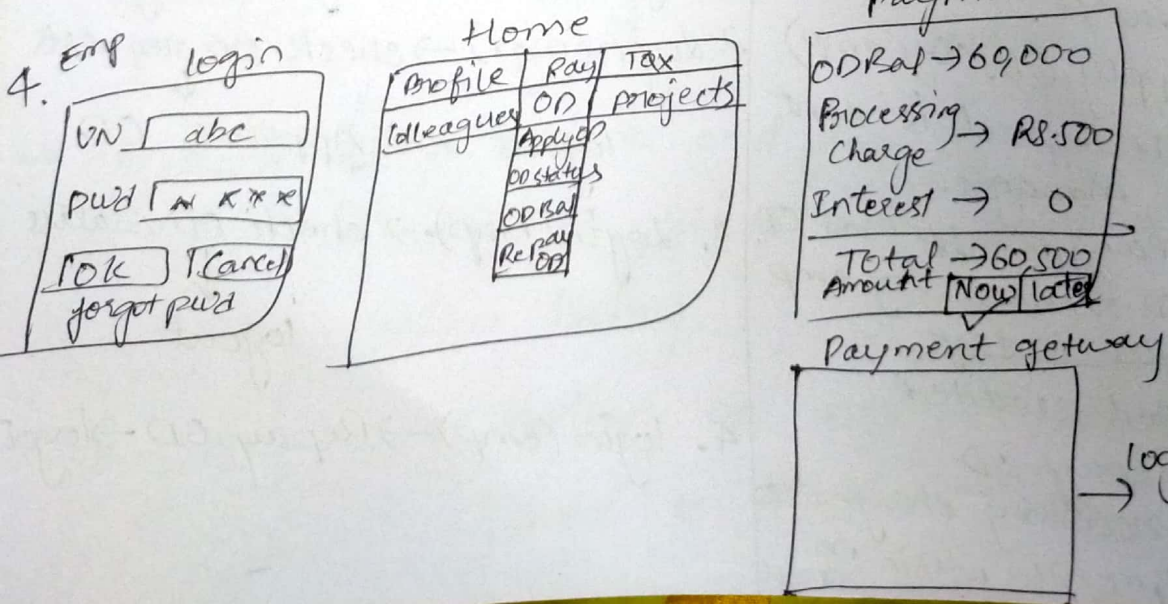


→ After 10 days

3. Emp



4. Emp



NOTE :-
→ Some
we m
order
we ta
→ System
machin
that
machin

Scanned with CamScanner

NOTE :-

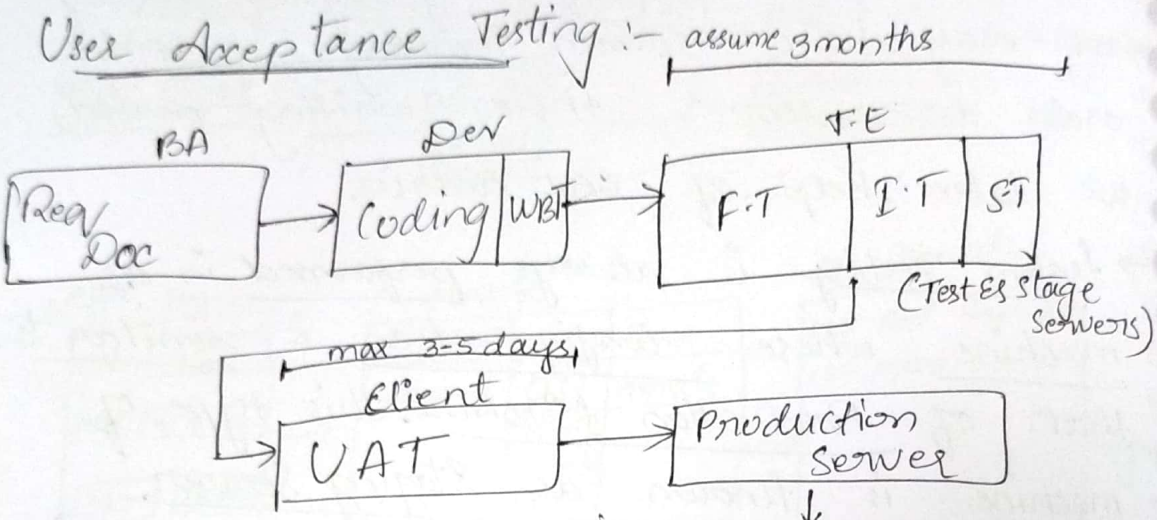
→ Sometimes while performing System Testing we may have some waiting period, in order to overcome those waiting period, we take help of SQL Queries.

→ System Testing is always performed in a machine whose configuration is similar to that of Production Machine. This type of machine is known as Staging Server.

Scanned with CamScanner

31/07/21

User Acceptance Testing :-



1. Based on Business knowledge
2. Most commonly used scenarios
3. Real-Time Scenarious

Difference b/w ST & UAT :-

→ In ST the end to end flow for all possible scenarios are checked by TEs. whereas in UAT based on real-time most commonly used scenarios the data flow is checked by the client himself.

→ In ST T.Es have whole product knowledge for UAT client has the knowledge of features which is required for him to know.

UAT :- It
the clien
→ In UAT
on all
check f
for thei
→ General
rios a
still th
self sc

2/08/21

Cases Of

1. It is d
- In
domain
of them
the a
should

Scanned with CamScanner

UAT:- It is a type of testing done by the client before accepting the product.

→ In UAT the client donot concentrate on all the features but they mainly check for the regularily used scenarios for their business.

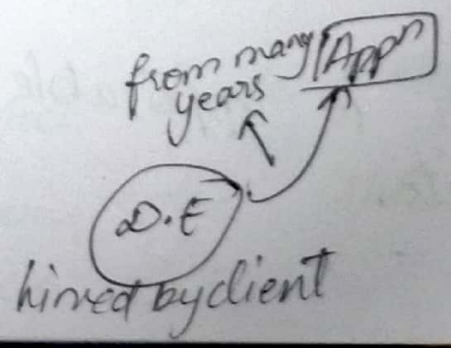
→ Generally, even though the business scenarios are checked by the TEs in the company, still the costumer do it for their self satisfaction.

2/08/21

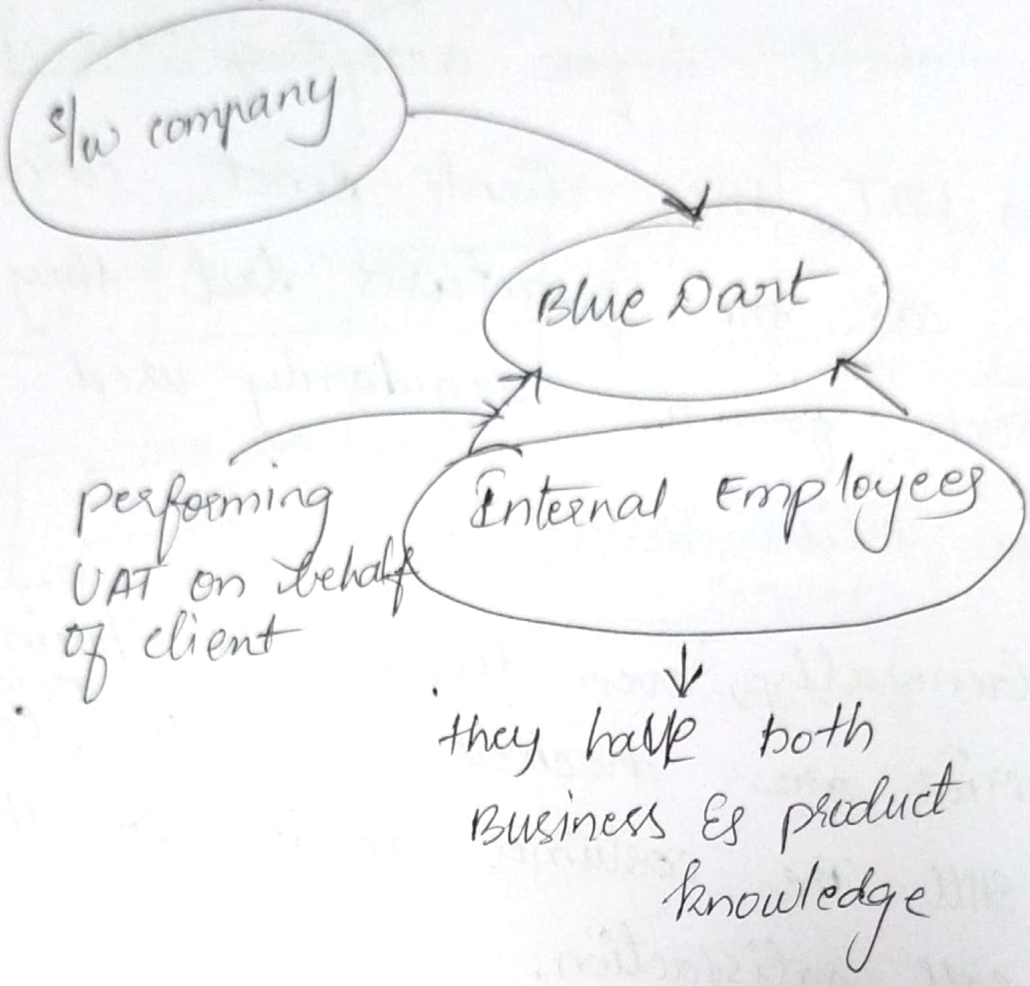
Cases Of UAT :-

① It is done by the client (Domain Expert)

- In some company the client hires the domain expert to perform UAT on behalf of them. The domain expert on the basis of the domain knowledge, checks the appn which should meet the client's business need.



② It is done by the client (Internal Employe

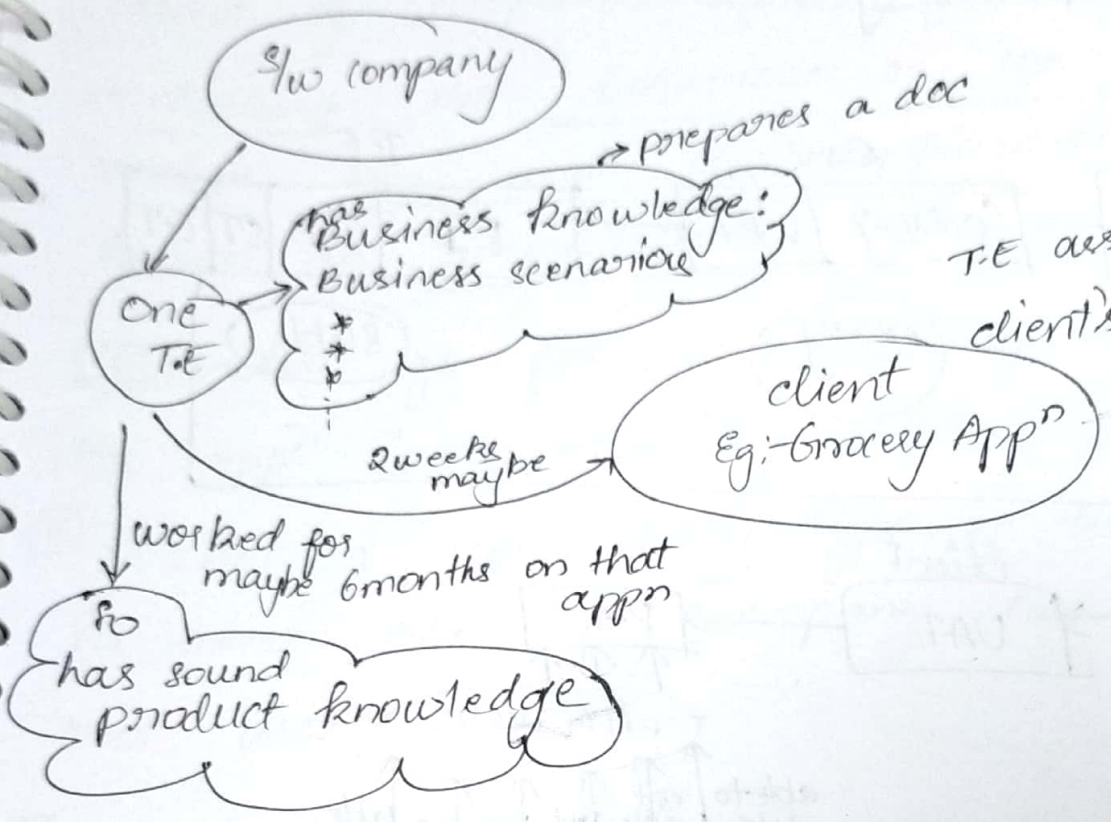


— Sometimes the client asks their Internal Employees to perform UAT on behalf of them, i.e., the internal employees has the complete business knowledge along with which the client also have the produ-ct knowledge. With both the knowledge the internal employees perform one round of UAT.

<u>NOTE</u>:- this approach is applicable only for the old existing products.

Scanned with CamScanner

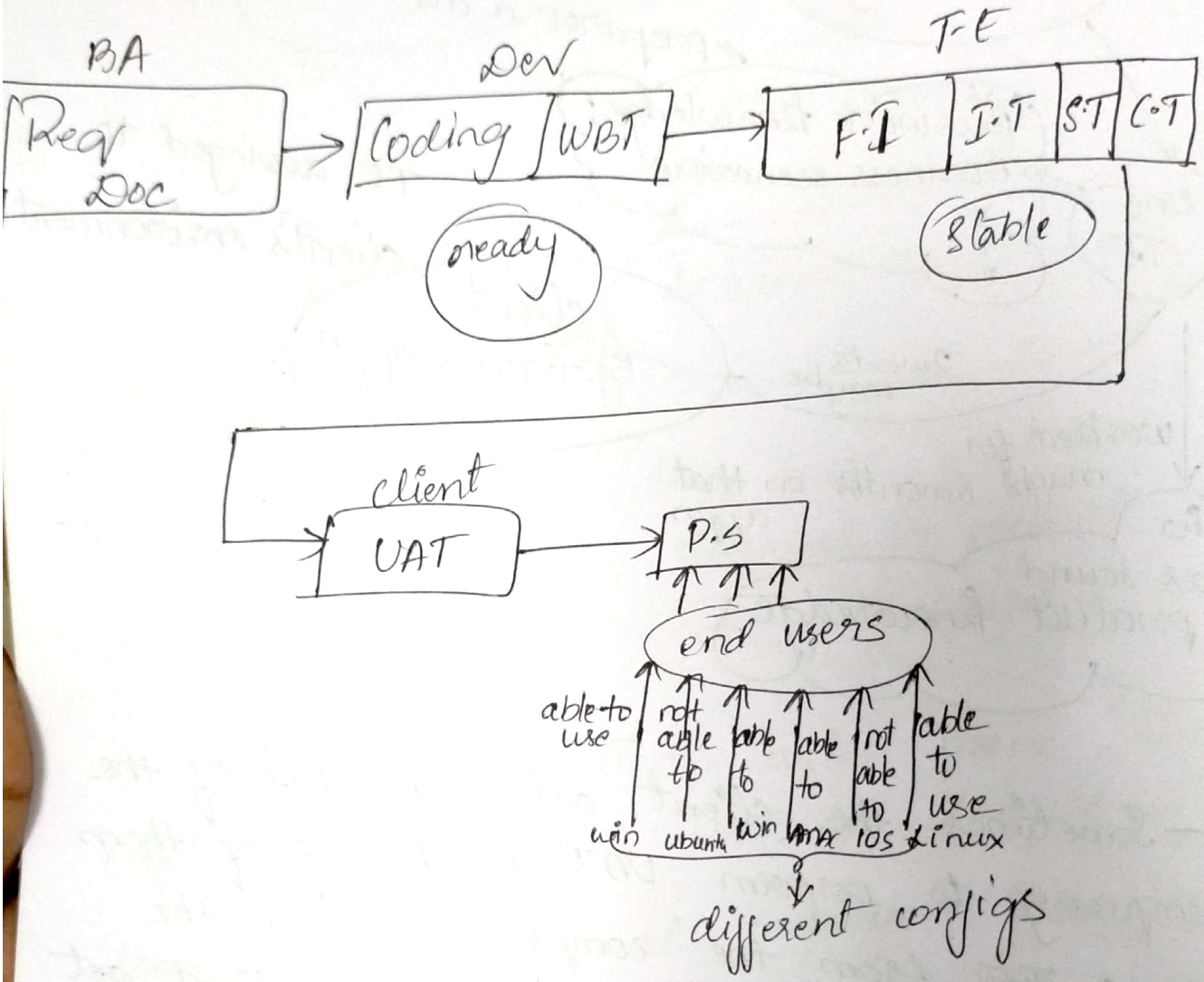
③ It is done by the client (T.Es of the Company)



- Sometimes the client asks the T.Es of the company to perform UAT on behalf of them i.e., a T.E from the company goes to the client's environment, sit with the client get their business knowledge, create document based on the business scenarios and at the same time, the TE also has the product knowledge since they work for the project inside the company. With both the knowledge the T.E perform one round of UAT.

2/08/21

Compatability Testing in BBT:-



— Once the appⁿ is functionally stable it is moved into the Production Server, where the end users start accessing the appⁿ and we donot have any control over users machine. In such case the end users start accessing the appⁿ with various system configurations. As a result some of the users may not be able to access the appⁿ. Inorder to overcome such issues, inside

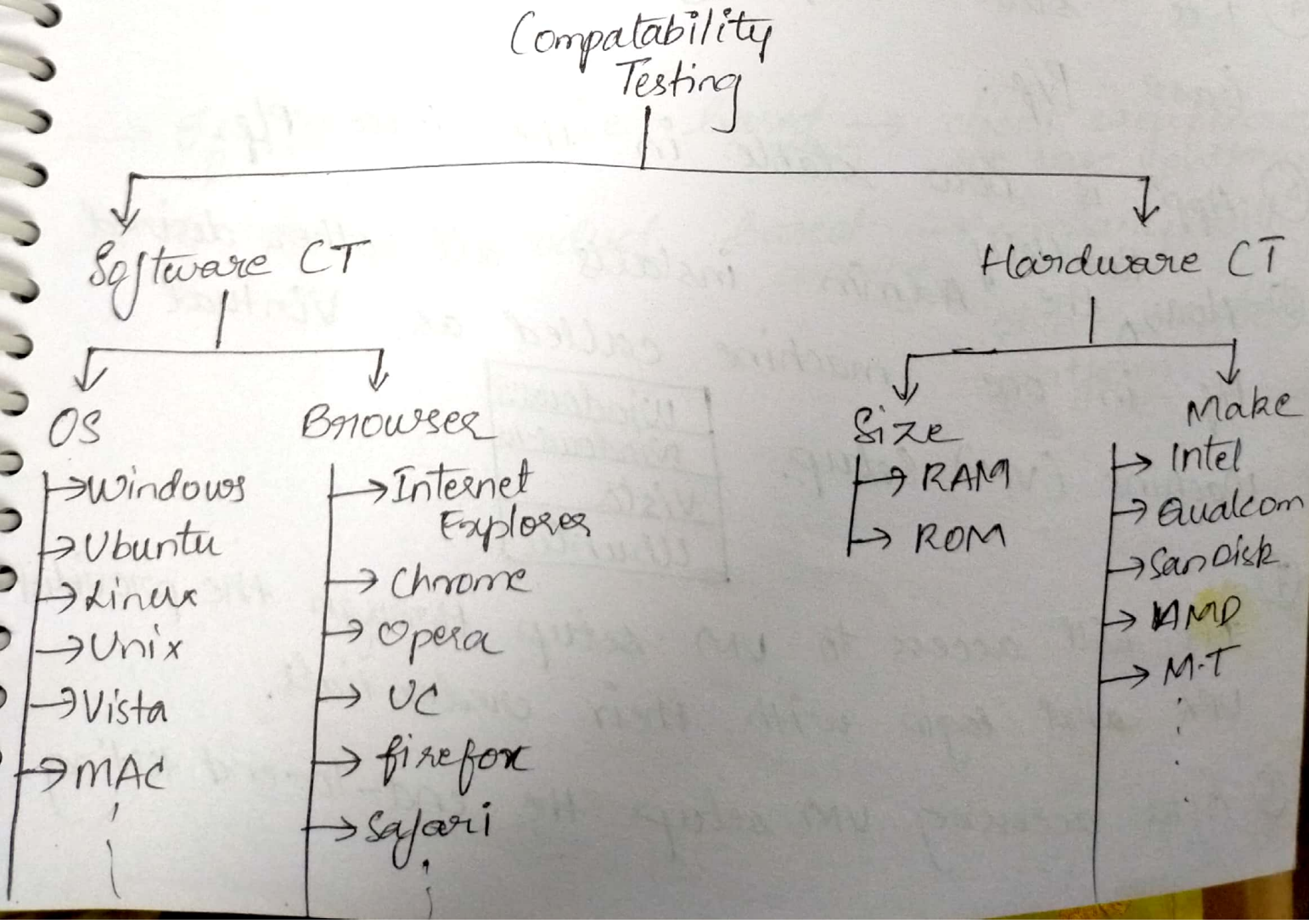
the company the test engs perform one round of compatability testing.

- checking the behaviour of the appⁿ in various h/w and s/w configurations is known as compatability testing.

NOTE:- ① We go for compatability testing once the appⁿ is functionally stable.

② It's not a compulsory type of testing since in some appⁿ the company has a control over user's machine.

Types of Compatability Testing :-



Scanned with CamScanner

* OS Compatability Testing Process:

① List out desired Platforms → provided by the client.

Windows-7
Windows-8
Windows-10
Vista
Ubuntu

② Identify the Base Platform (by market analysis) [assume windows-7 as base platform].

③ Admins Install the base platform in every Test Engineer's machine.

④ T.Es start performing FT, IT, ST on the base P/f.

⑤ App^n is now stable in the base P/f.

⑥ Now, Parallely the Admin installs all other desired P/f in one machine called as Virtual Machine (VM) setup.

Windows-8
Windows-10
Vista
Ubuntu

T.Es will access to VM setup through the provided URL and login with their credentials.

After accessing VM setup the end-to-end testing

Scanned with CamScanner

of the appⁿ is performed in all the desired P/fs.

⑨ If compatability Issues such as:

→ Overlapping Object

→ Missing Object

→ Alignment Issues

→ Look & feel occurs then the TE reports them to the dev and the dev gets it fixed.

⑩ This process of CT continues until the appⁿ is working fine in all the desired P/fs.

→ If its a service based company → client says/chooses the base platform

→ If its a product based company → company will choose the base platform.

Scanned with CamScanner

Cross

* Browser Compatability Testing Process:

→ In order to perform browser C.T, we donot need VM setup since we can install multiple ~~OS~~ browsers in a single OS. In order to access it we can parallely use it by shifting the windows, But, sometimes costumer want their appⁿ to be stable in different versions of the same browser, in such case we need VM server.

Scanned with CamScanner

3/08/21

Test Plan:-

→ Test Plan is a dynamic (changable) document which derives the entire testing process.

→ It is prepared at initial stage, once we get the requirement from the client.

→ It is a document prepared by the test leads and test managers.

NOTE:-

1. For every release of the project, we may have separate test plan, plan may varies from release to release for the same project.

Various Attributes of Test Plans :-

The various Attributes of Test Plans

1. Objective
2. Scope
3. Approach
4. Test Methodology
5. Test Environment
6. Schedules
7. Templates

Scanned with CamScanner

8. Tools
9. Roles & Responsibilities
10. Effort Estimation
11. Entry & Exit Criteria
12. Test Deliverables
13. Assumptions
14. Risk
15. Mitigation

---

1. Objective:
- It talks about the purpose or aim of writing the test plan.
- For example, this test plan is written to test the functionalities of the appⁿ. (e.g.: (whatsapp release no 18))

2. Scope:
- Scope talks about the features which need or need not to be tested in the current release of the appⁿ.
- Scope is of two types:

(i) In Scope - which need to be tested in the current release

(ii) Out Scope - which need not to be tested (which is already done in previous release) or check few

chat ✓ contacts Profile setting Rel1

Status: calls Rel2

calls in chats page Rel3 → for Rel 3 ⇒ In → ✓ Out → no ✓

3. Appr
- s
or
E

4. Test
- S
which
curre
-
Sm
Co
el

5. Test
-
nec
we
os

6. Sch
-
in
i.e.,
to
-

Scanned with CamScanner

3. Approach:

- Approach talks about the sequence or flow of the appⁿ.

Eg:- login → home → compose → sent Items → logout

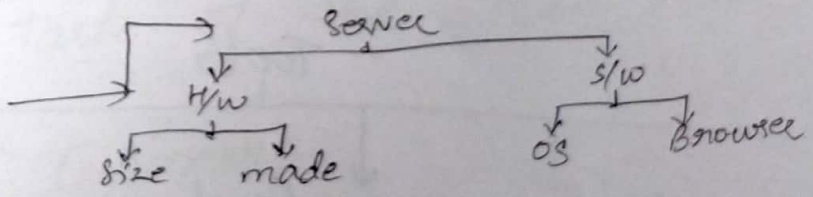
4. Test Methodology:

- It talks about the type of testing which needs to be performed in the current release of the appⁿ.

- The various types of testing such as: Smoke testing, functional testing, IT, SI, UAT, Compatability Testing, Performance Testing, usability Testing, Adhoc (Monkey) Testing.

5. Test Environment:

- It talks about the surrounding or necessary machine configurations on which we perform the test execution.



6. Schedules:

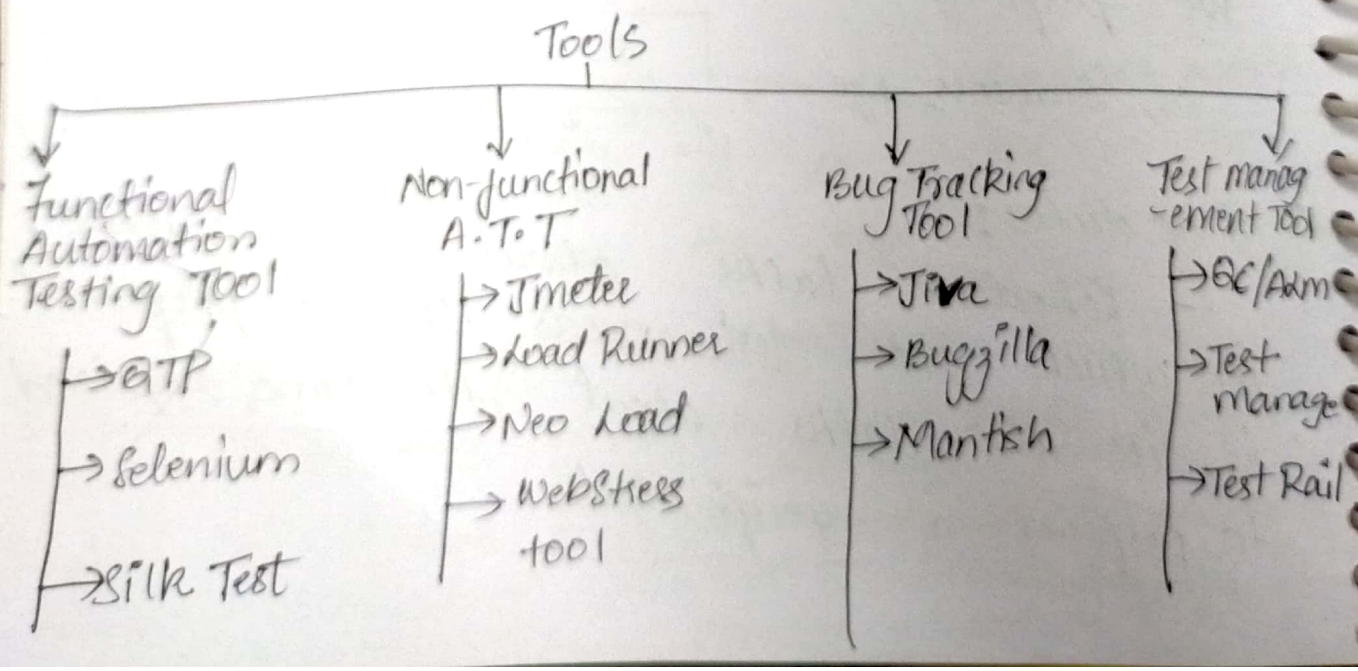
- Schedule talks about the sequence in which the test needs to be performed. i.e., it also talks about the time required to perform a specific task.

Scanned with CamScanner

7. Templates:

- These are predefined formats of preparing the test document.
- The various types of templates are:
  - (i) Test Case Template
  - (ii) Test Case Review Template
  - (iii) Requirement Tracability Matrix
    Cross Reference Matrix
    RTM / CRM Template
  - (iv) Test Execution Template
  - (v) Bug Report Template
- Every company follow their own template to maintain the uniformity b/w the test engineers.

8. Tools:



04/08/21

9. Roles

a) R&

→

→

→

→

b) R

→

→

→

c) R

→

→

Scanned with CamScanner

04/08/21

9. Roles & Responsibilities:

a) R & R of Test Manager:-

→ Review/prepare test plan.

→ Conduct meetings with the clients.

→ Conduct monthly status meeting

→ Review the day-to-day activities and plan for upskilling the team members.

b) R & R of Test Lead:-

→ Participate/prepare the test plan.

→ Assign the task to the team members.

→ Track day-to-day activities of the team members.

→ Conduct daily stand-up meeting.

→ Review the task of the team member.

c) R & R of Test Engineer:-

→ Understand the requirements.

→ Participate in various types of meetings.

→ Prepare the test document.

→ Perform test execution process.

→ Log & track the bugs.

Scanned with CamScanner

10. Effort Estimation:

→ It talks about the efforts which needs to be applied by each team member to acheive the goal.

11. Entry & Exit Criteria:

→ It talks about the necessary cond^ns which need to be fulfilled inorder to ~~acheive the goal~~. start or stop the testing process.

⇒ Entry Criteria :-

1. Build should be ready
2. Necessary Test Doc should be ready
3. Task should be assigned to the team member
4. Resources necessary must be ready.

⇒ Exit Criteria :-

1. All test cases must be executed.
2. Most of the test cases must be passed.
3. No Blocker, Critical or MajorBugs must be present. whereas few minor bugs may exist.

12. Test
→ The
need
along
1. Test
2. RT
3. Tes
4. So
5. Tr

13.

14.

Scanned with CamScanner

ich needs
er to

ns
er to
sting

12. Test Deliverables:

→ These are the necessary docs which need to be handed over to the costumer along with the appⁿ.

1. Test Doc [scenarios & cases]
2. RTM Doc
3. Test Execution Report
4. Source Code
5. Release Note –
   (i) List of new features.
   (ii) List of existing features.
   (iii) List of existing bugs.
   (iv) List of fixed bugs from previous release.
   (v) User Manual.
   (vi) Installation Guide.

13. Assumptions:

→ Assumptions are the challenges which may occur while the process of testing.

14. Risk:

→ Risk are the consequences of the assumed problem.

Scanned with CamScanner

15. Mitigation:

→ Mitigation is the back up plan to overcome the assumed risk.

05/08/21

Usability Testing in BBT:-

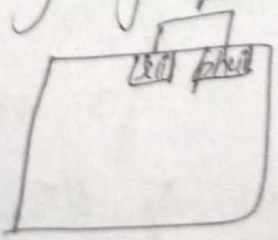
→ Checking the user-friendliness of the appⁿ is known as U.T.

→ The various factors of U.T are:

1. Easy to Access
   - All important features must be highlighted.

→ Among these words ~~there's~~ signin/signout are mingled. (not highlighted).

2. Easy to Understand
   - It should be in simple english language.



→ not understandable but perform simple fns.

Scanned with CamScanner

3. Look & Feel
— font size, colour, style, background

Pink Bg with yellow text

like 'cmd'.

4. Quick Response

fea-25	fea-50
FB	LB
Response within seconds	Response in minutes

So people would prefer FB instead of LB.

NOTE:-

1. Usability Testing is not a compulsory type of testing since in some appns UI doesnot play a vital role. For example, plug-ins in browsers, snipping tool, command prompt etc.,

2. We go for UT once the appn is functionally stable.

Usability Checklist: (total 40 are there)

→ Page must have logo.
→ Home button must be enabled.
→ Pages must have link.

Scanned with CamScanner

→ login & logout features must be present.

How to perform U.T:

Initial stage:-

1. Identify the need of U.T
2. Derive the checklist (by T.E)
3. Hand it over to the client for review & approval.
4. Implement the changes.

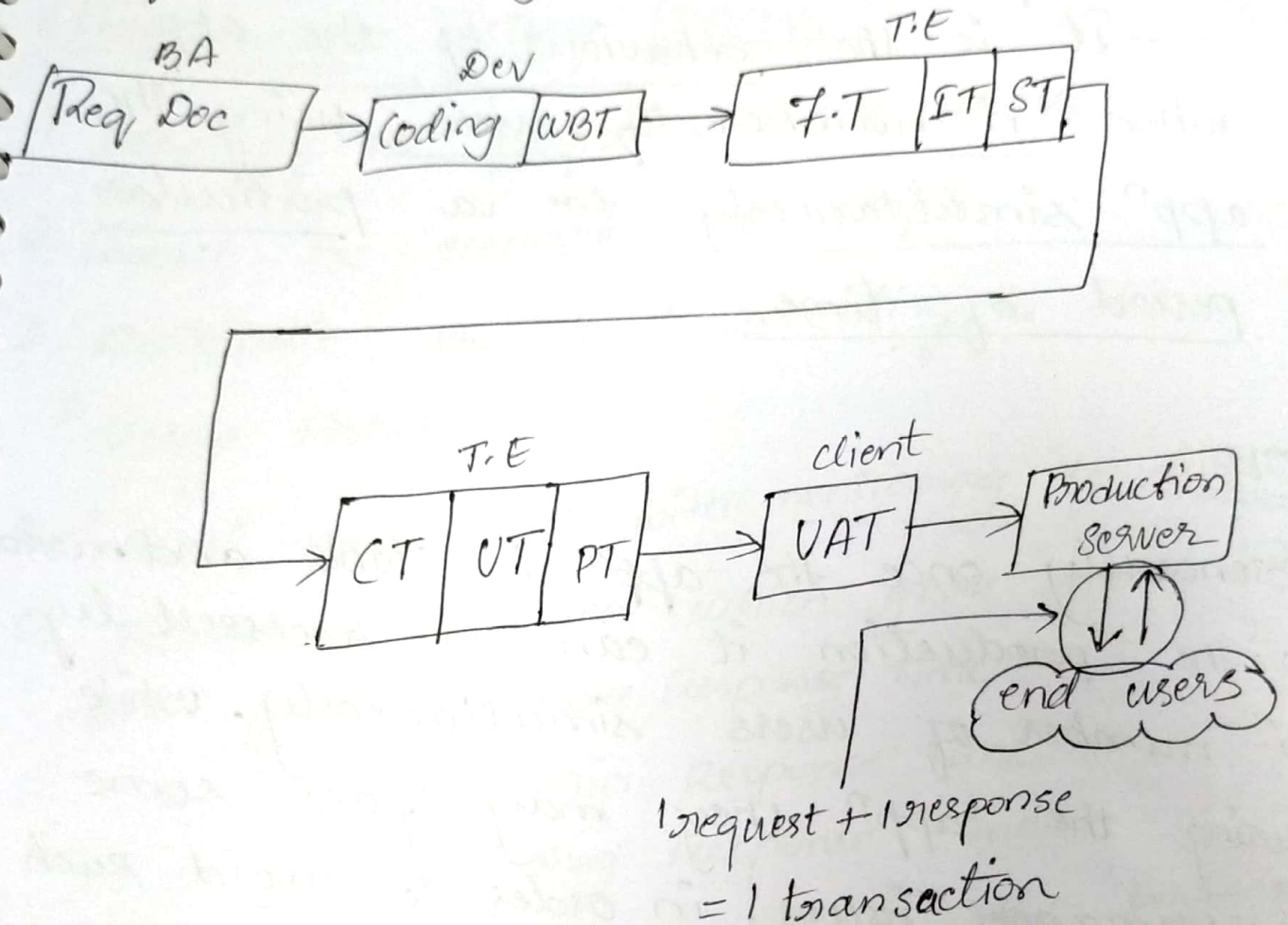
→ once the appn is stable

5. Execute the checklist on the appn
6. Usability checklist report.

Usability checklist Report:

Pages	1	2	3	4	3	4	2
Login	P	N/A	N/A	N/A	N/A	N/A	P
Home	N/A	N/A	P	P	P	P	P
Link	P	P	P	N/A	P	P	P
⋮							
Logout	N/A	P	P	P	P	P	P

Performance Testing In BBT:



→ Checking the behaviour of the appⁿ by applying load is known as performance testing.

→ Various factors of performance testing are:-

1. Load
   - No. of users using the appⁿ simultaneously/at a time.

2. Response Time
   - It is the time taken by the server to give response for a particular request.

Scanned with CamScanner

3. Stability.

- It is the behaviour of the appⁿ when 'n' number of users using the appⁿ simultaneosly for a particular period of time.

NOTE :-

1. Generally once the appⁿ is stable and moved to the production it can be accessed by 'n' number of users simuttaneosly. While using the appⁿ they may face some performance issues, in order to avoid such issues we go for one round of performance testing.

2. P.T is always done with the help of tools since manually performing P.T is a costlier approach as well as accuracy can't be maintained while sending the request and getting the response. Some commonly used P.T tools are Jmeter, Load Runner, Neo Load, Webstress Tool, Nload.

Not compulsory
eg- biometric (one load is there)

06/08/21

Performa

1. Identif
2. Convert
3. Distrib
" Usage
4. "RUN"
5. Resu
-tant

6. Analy
team)
7. If th
identif
⇒ Bottle
h/w /
8. After
9. After
10. This
goal

06/08/21

Performance Testing Process :-

1. Identify the performance scenarios.
2. Convert the scenarios into scripts.
3. Distribute the load according to the "Usage Pattern".
4. "RUN" the scripts, with mentioning the duration.
5. Result is obtained which gives three impor-tant factors: (i) Max Response Time
   (ii) Min Response Time
   (iii) Avg Response Time
6. Analyse the result (done by the performance team) (Avg RT is considered).
7. If the Avg RT is more than the desired RT, identify the "Bottle Neck".

   ⇒ Bottle Neck are the performance issues in h/w / s/w configs / network / scripts.
8. After identifying "Bottle Neck" go for "Tuning".
9. After "Tuning" re-run the scripts.
10. This process continues until we acheive the goal.

Scanned with CamScanner

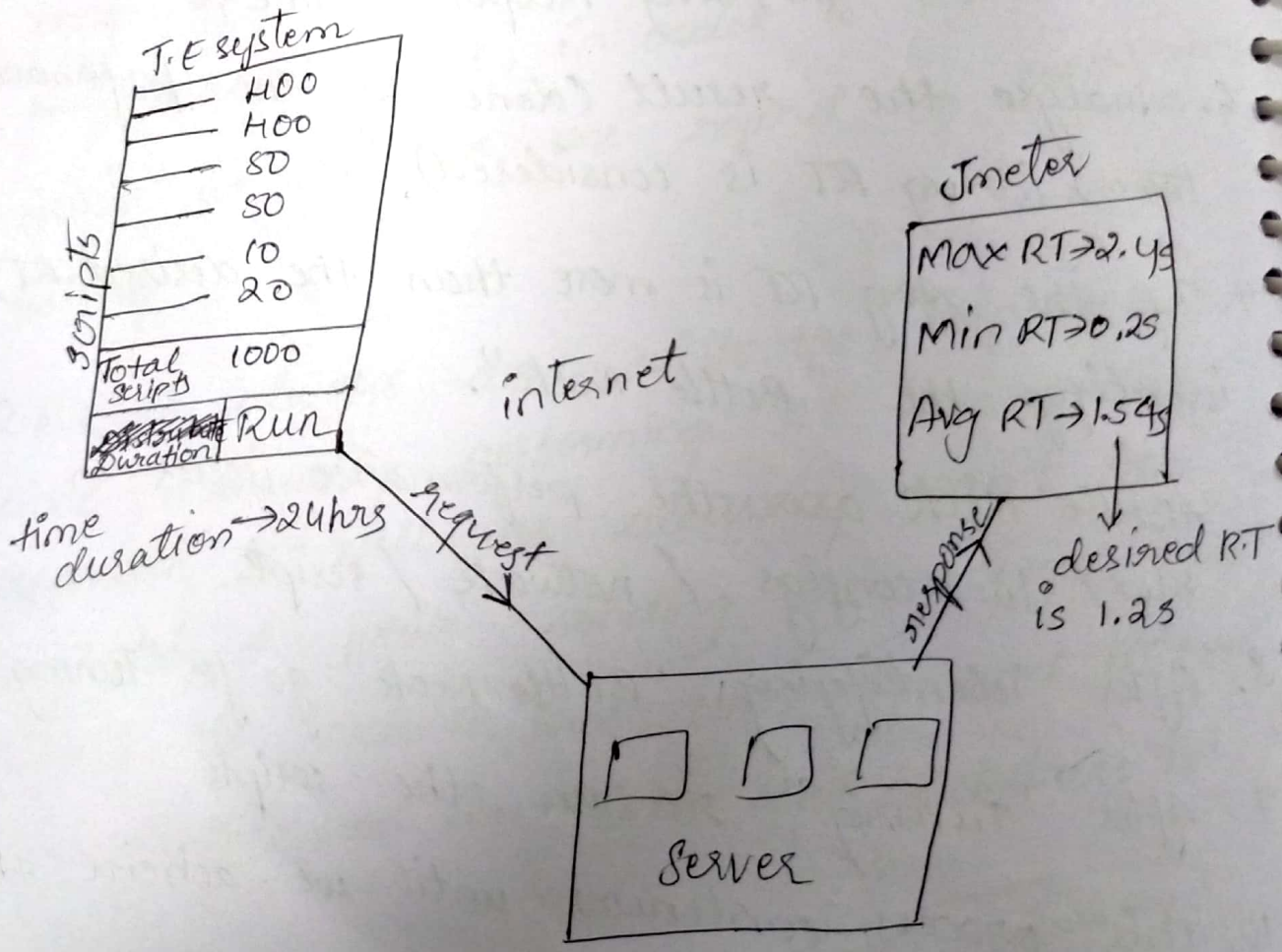
Example:- Gmail

performance scenarios

↓

load distribution (for 1000 users)

400	1. login → home → compose → Sent Items → logout
400	2. login → home → Inbox → logout
50	3. login → home → draft → logout
50	4. login → chat → logout
10	5. login → help → logout
20	6. login → settings → logout



Types

1. Load
2. Str
3. Sta
4. Sc

1. So
2. Vo

Slow part

1. Load

→ C
lo
→

2. Sl

Scanned with CamScanner

Types of Performance Testing:-

→ logout

1. Load Testing
2. Stress Testing
3. Stability Testing
4. Scalability Testing

---

1. Soak Testing
2. Volume Testing

} → not until interviewer asks.

Slw part

1. Load Testing:

→ Checking the behavior of the appⁿ by applying load less than or equal to the desired load.

→ For example:-

A client wants the appⁿ to be checked for 1000 users. So we give a load which is $\angle = 1000$.

2. Stress Testing:

→ Checking the behaviour of the appⁿ, by applying load more than the desired load.

→ For example:-

Costumer wants his/her appⁿ to be tested for a load of 1000 users. We apply a load of > 1000 users.

Scanned with CamScanner

3. Stability Testing:

→ checking the behaviour of the appⁿ by applying load continuosly for a long period of time is known as stability testing.

→ For example :-
costumer wants the appⁿ to be tested for 1000 users. 1000 users using the appⁿ at a time for say 48hrs. Meanwhile even after 48hrs or within the appⁿ shouldn't crash.

4. Scalability Testing:

→ Checking the behaviour of the appⁿ by ↑ing or ↓ing the load in a particular scale is known as Sc.T.

→ Sc.T are of two types:

Upward Scalability		Downward Scalability	
1000	1.2S	1000	1.2S
1100	1.2S	900	1.2S
1200	1.4S	800	1.0S
1300	1.4S	700	0.99S
1400	1.49S	600	0.99S
1500	1.5S	500	0.90S
1600	1.5S		
1700	1.6S		

→ To
R.T is

H/W part
1. Soak
→ C
uns
Test
→
2. Vo
→

→ To tell costumer at what load the desired R.T is acheived.

H/W part

1. Soak Testing:-

→ Checking the behaviour of the appⁿ in an unsupportive environment is known as Soak Testing.

→ For example:-

Consider the h/w config to be of 4GB ram in which 1000 users load could give result in 1.2s. The same appⁿ is checked in 2GB ram and if it gives 1.2s R.T thats more preffered one.

2. Volume Testing:-

→ Checking the behaviour of the appⁿ with huge amt of data transaction is known as V.T.

→ For example:-

Uploading a large file or downloading the same.

Scanned with CamScanner

06/08/21

Severity & Priority :-

→ Severity talks about the impact of the bug on the appⁿ, that is how much the bug is effecting the appⁿ is known as its severity.

→ Severity is decided by the T.Es but it can be changed by the devs since wrong severity may effect the performance of the dev.

→ The various levels of severity are:

1. Blocker / show stopper
2. Critical
3. Major
4. Minor

1. Blocker :

- Blocker bugs are those bugs, due to which the complete test execution process stops.

- Eg:- If the compose option is clicked but a blank box appears we would not be able to check other modules such as inbox sent items and all.

2. Critical

- C

are

appⁿ

- E

g

3. Major

- M

in

fun

-

4. Mino

-

mu

-

Scanned with CamScanner

2. Critical Bugs:

- Critical bugs are those bugs which are present in the main functionality appⁿ.

- Example:-
  The mail in the sent items are not getting opened.

3. Major Bugs:

- Major bugs are those bugs which are found in the supporting functionality of the main functionality.

- Eg:-
  A particular type of attachment is not working.

4. Minor Bugs:

- All UI related bugs are known as minor bugs.

- Eg:-
  Font size, color, background, . . .

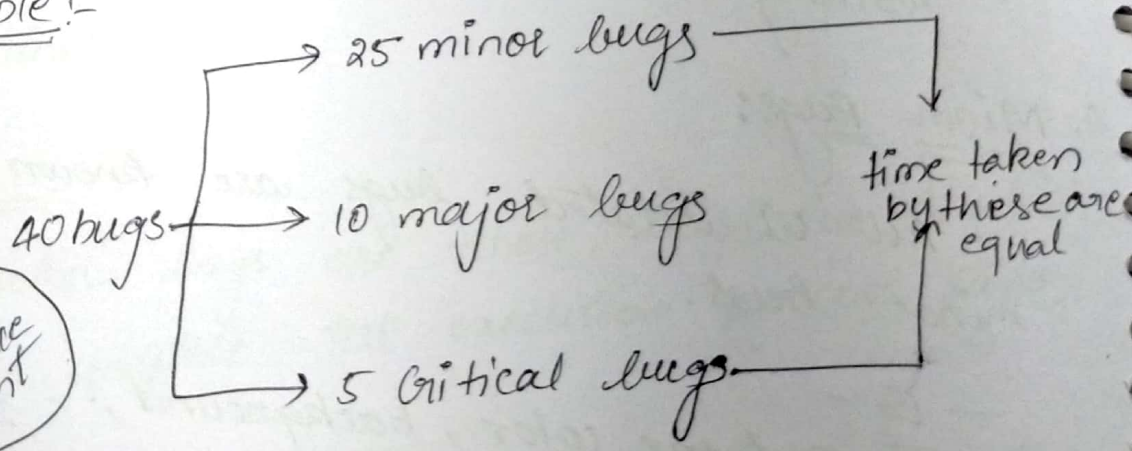
Scanned with CamScanner

Priority :-

→ Priority talks about the sequence in which the bug need to be fixed i.e., when, which bug need to be fixed is known as its priority.

→ Priority is decided by the TEs and it can't be changed by the devs since if dev get a chance to change the priority, they'll fix all minor bugs fix and if major bugs are fixed in the later stage, the TE will not have enough time to check the impact area.

→ Example :-

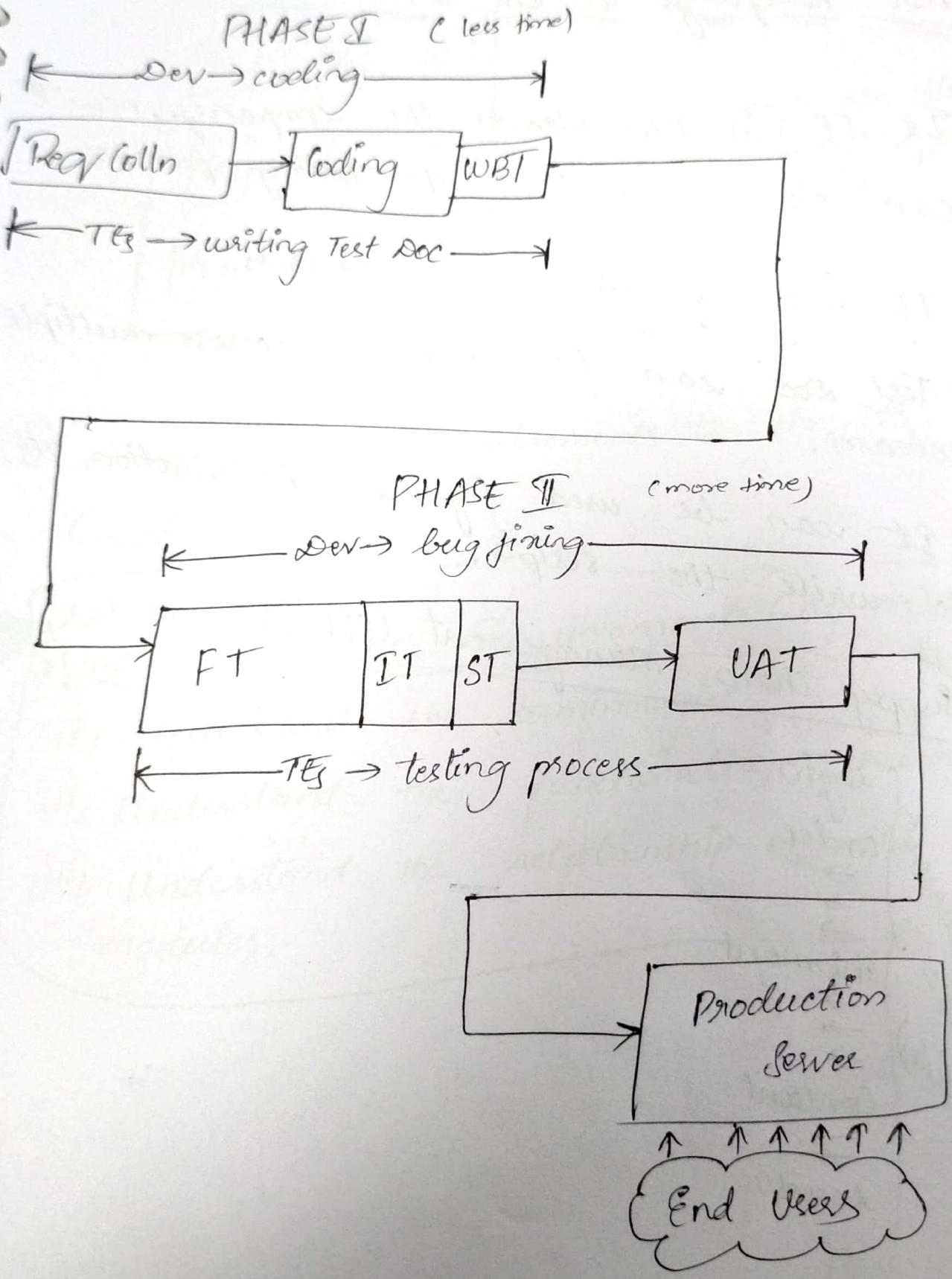


dev always tries to reduce the bugs count

→ So dev tries fixing the 25 minor bugs
→ 5 CB are left later this gives rise to more 25/30 major bugs.

04/08/21

Test Doument



Scanned with CamScanner

* Advantages of writing the test Doc and not keeping it in our brain:-

1. If TE in b/w leaves the company, new TE can easily continue the testing process.

2. TE may forget some scenarios.

3. Test Doc can be re-used accross multiple releases.

4. It can be used by the Automation TEs to write the scripts.

Example

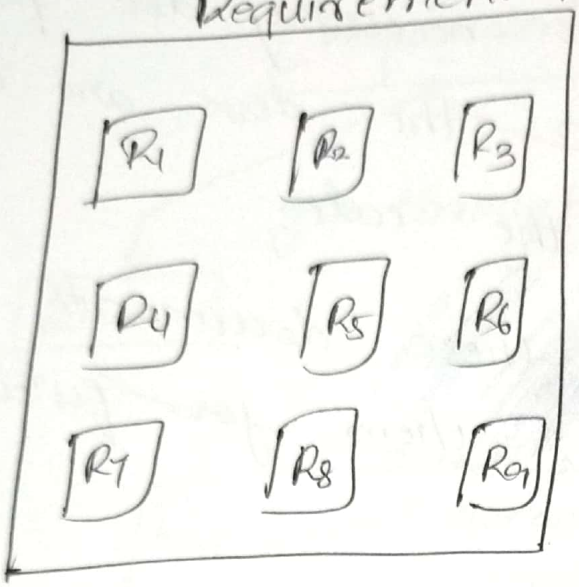
Supply chain management (appⁿ):-

Requirement doc (modules)

- login
- order
  - →
  - →
  - →
- shipment
  - →
  - →
  - →
- Contant
  - →
  - →
- Lead
- ⋮
- Logout

(1)

Requirement Repository



→ All the reqs are stored in the requirement reposi-tory by BA himself.

(2) Assigning the requirements to the TEs (called task).

what does it mean?

→ Understand the Requirements (by TEs):-

(i) Understand the terminologies.

(ii) Understand the functionality of sub-req.

(iii) Understand the relationship b/w the modules.

Scanned with CamScanner

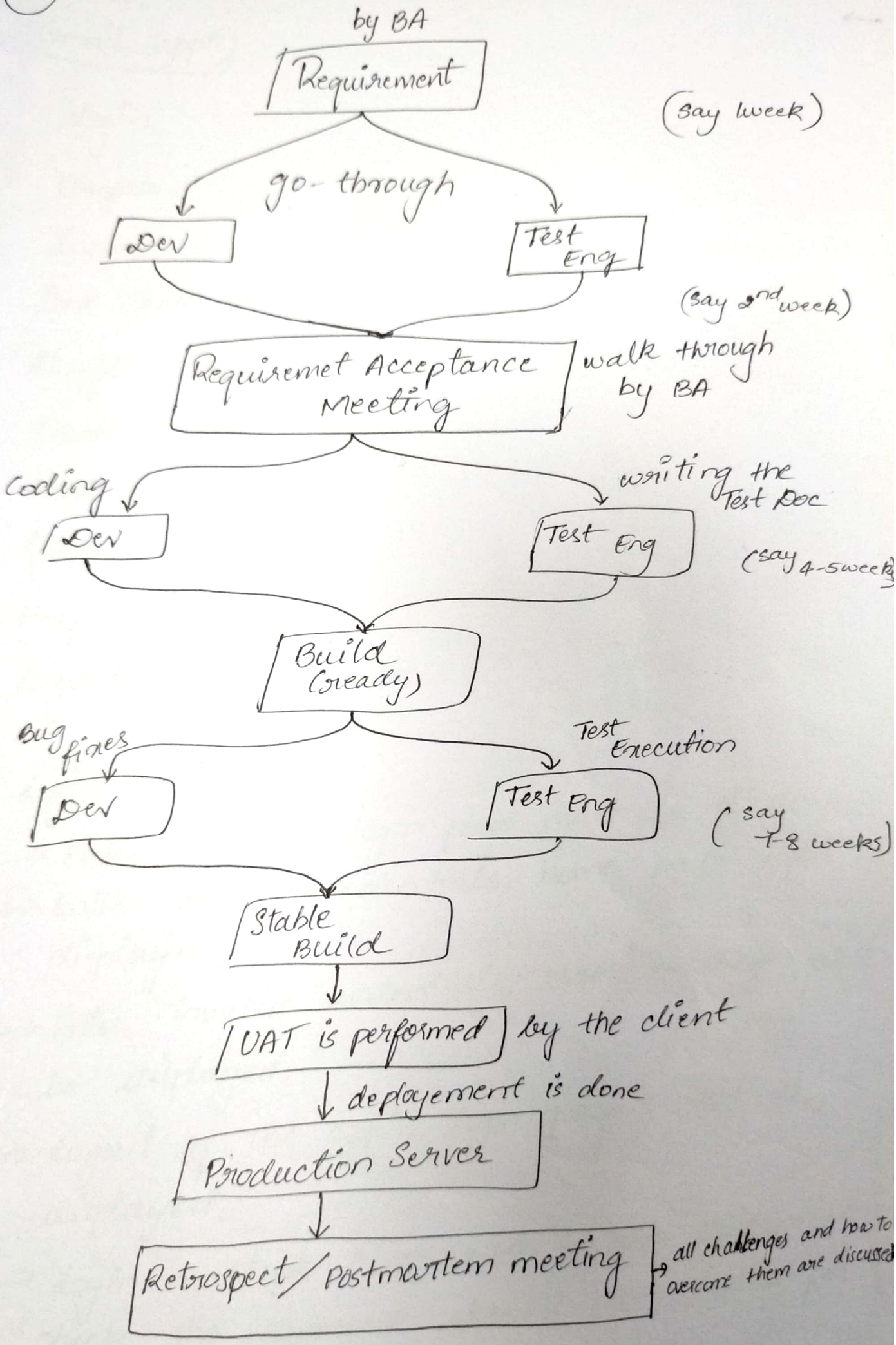
Test Document:-

→ Test document is the necessary doc prepared by the test eng when the devs are completely busy in writing the code.

→ Generally we write these documents so that we can refer them for future test execution process.

→ Once the test doc is ready thereafter the complete test execution depends on the test document itself and we donot refer the requirements to perform the test execution.

Scanned with CamScanner



Scanned with CamScanner

09/08/21

Gmail (appⁿ)

login
Compose
Inbox
Sent Items
Draft
Trash
chat
Spam
Help
Logout

Login module:

→ Enter the url, login page must be displayed.
→ Enter valid credentials, home page must be displayed.
→ Enter invalid credentials, error message must be displayed.
→ Logout of the appⁿ, login page must be displayed.
→ login multiple times with invalid credentials, check the account should be blocked for sometime.

Scanned with CamScanner

Compose module:

→ ~~Comp~~ Click on compose module, compose page mbd.

→ Compose a mail ~~and send~~, the mail must be received in inbox.(receiver's)

→ Compose a mail ~~and~~, the sent mail in sent items must be displayed. (copy of that mail))

→ Compose a mail ~~and~~ discard, the mail must be present in the draft.

→ Compose a mail to myself, the mail must be received to my self ~~inbox~~.

Inbox module:

→ click on inbox, inbox page must be display-ed.

→ Mail received in receiver's inbox, the copy of that mail must be present in sent items.

→ Spam Mail received in inbox, the mail must go to the spam.

→ Mail sent to myself, the mail in my inbox must be displayed.

→ Save the inbox mail, the mail must be present in the draft.

→ Delete ~~the~~ inbox mail, the mail should go to trash.

Scanned with CamScanner

Sent Items module:

→ click on sent items, sent items page must be displayed.

→ Compose a mail, the copy of the mail must be present in SI.

→ Forward a mail from SI, the receiver's inbox must have the forwarded mail.

→ Save ~~the~~ a mail in SI, the mail goes to draft.

→ Delete a mail from SI, the mail must go to trash.

Draft module:

→ click on draft, draft page must be displayed.

→ Compose a mail and discard, the mail must go to draft.

→ Send a mail from draft, the mail must be received in receiver's inbox.

→ Send a mail from draft, the mail copy must be present in SI.

→ Delete a draft mail, the mail must go to trash.

→ Discard draft mail, the mail must go to draft with few modifications done.

Scanned with CamScanner

Trash module:

→ Click on trash, the trash page must be displayed.

→ Delete a mail from inbox, the mail must be displayed in trash.

→ Delete a mail from SI, the mail must be displayed in trash.

→ Delete a mail from draft, the mail must be displayed in trash.

→ Delete a trash mail, the mail must be permanently deleted.

Chat module:

→ click on chat, chat (window) mail must be displayed.

→ Click on ~~a receiver's~~ chat, all active receivers must be displayed.

→ Click on a receiver's chat, all previous chats must be displayed.

→ Send a message, the message must be received in receiver's chat box.

→ Open chat window, calling option must be present.

→ Delete a chat, the chat must be permanently deleted.

Scanned with CamScanner

Spam module:

→ Click on spam, spam window must be displayed.

→ Send a spam (mail) message, the mail should go into spam instead of inbox.

→ Go to inbox and mark a mail as spam, the mail should be present in the spam now.

→ Include some hypelinks in mail, that mail should go to receiver's spam.

→ Delete a spam mail, it should go to trash.

Help module:

→ Go to help, help window must be displayed.

→ Go for all pages, help must be enabled.

→ Go to help. ~~and~~ ~~follow~~ select an issue and follow the steps provided, the result must be exactly as that provided in help.

→ Close help window, the previous page open befor help must be displayed.

Scanned with CamScanner

Logout module:

→ Click logout, login page must be displayed.

→ Login appⁿ, logout must be present.

→ Go for all pages, logout page must be enabled.

→ Login with multiple accounts, by single step logout all accounts must be logged out.

→ Login with multiple devices, by logout the account must be logged out of all devices

→ Click logout, the logout must be enabled.

Scanned with CamScanner

09/08/21

Test Scenarious:-

→ Test scenarios are the high level doc which talks about all possible situations or multiple ways of testing the appⁿ.

→ Generally we write the test scenario so that we can understand the complete flow of the appⁿ.

→ In order to write test scenarios, we follow so rules:

1. Always start writing the scenarios by picking module by module so that we donot skip any module as well as a proper sequence / flow will be maintained ~~by~~ while writing the test scenarios.

2. While writing the test scenarios, we mainly concentrate on flow of the modules. But, sometimes we may talk about the components level.

3. While writing the test scenarios, we may create the test data. Hence delete should be the last option. Else we may waste a lot of time in re-creating the data.

Scanned with CamScanner

4. Test scenarios should be written in simple english language i.e in one or max two lines so that clarity exists while reading the scenarios and its easy to understand.

5. Test scenarios should be always written in <u>Do and Check format</u>, i.e., we perform so action and check for the result.

<u>Scenarios for Objects:</u>

Pen, Paper, Water Bottle, Table, Chair, Laptop, File, AC, Projector.

[Consider all the types of testing, FT, IT, ST, (UT, CT, PT), Adhoc]
in seq — not req, does check — noneed of seq. — at last

<u>For Pen:</u>

→ Remove the pen cap, the cap must be removed.
→ Write on a paper, ~~the~~ it must be written.
→ Close the pen cap, it must be fixed properly.
→ Remove the ink tube and insert again, the pen must be fixed properly.
→ Remove the empty tube ~~and~~, refil it, it must work properly.
→ Remove the pen cap ~~and~~ write on a paper and close the cap.

→ Open pe
and wri
work.
→ Open pen
be there
→ Hold
pen sh
→ hold
grip,
fluent
→ Write
→ Write
→ Writ
→ ~~Writ~~
pen
→ Drop

Scanned with CamScanner

→ Open pen cap and keep it open for a while and write, for instance the pen should not work.

→ Open pen cap and hold it, proper grip must be there.

→ Hold the pen with ease, thumb grip, the pen should not fall.

→ hold the pen at a certain height from grip, ~~the~~ and write, it must not write fluently.

→ Write on paper, it should fluently write.

→ Write on hand, it should not write fluently.

→ Write on board, it must not write.

→ ~~Write~~ Take pen write for 200 pages, pen ink must finish.

→ Drop a pen from certain height

Scanned with CamScanner

Water Bottle

→ Open the cap, check that the cap's open.

→ Pour water into the bottle and see it holds it.

→ Close the cap, and check cap got fixed tightly.

→ Open the cap pour the water see it holds it close the cap. Then after a while again open the cap ~~and~~ drink the water and close the cap.

→ Hold the water bottle in hand and see its comfortable to hold.

→ Drink water from bottle and see its comfortable to drink.

→ Fill bottle with different types of fluids and see it holds it.

→ ~~Keep~~ Fill different temperature fluids and see it holds it.

→ Keep the water bottle on a chair to check it stands properly.

→ Re-use the water bottle for long time to check it's safe to use.

Scanned with CamScanner

→ ~~Re~~ Smell the water and no smell should come out of it.

→ Taste the water the taste should not change.

→ Drop the water bottle from small height and see it's not broken.

Negative Scenarios on Water Bottle:-

→ ~~Fill the~~ Close the cap of the bottle with water and put the bottle upside down and see the water should not come out.

→ Close the bottle with another bottle's cap and check it should not close.

→ Try to open the cap in clockwise direction and see it should not open.

→ Try to close the cap in anti-clockwise direction and see it should get closed.

→ Tightly fix the cap and check it should not move.

→ Put the bottle in the boiling water, it should not melt.

→ Fill the water into closed cap and see it should not get filled.

→ Fill the bottle against its capacity and see it should not hold it.

Paper:

→ Take a paper and check it's size, the paper must be of few inches length enough to write some 400 words.

→ Write on the paper with a pencil, the written part must be clearly visible.

→ Write on the paper with a pencil and erase the written part, it ~~should not~~ must be neatly erased.

→ Write on the paper with a pen, it must be clear to read.

→ Write on the paper with a sketch pen, it must leave marks back of that paper.

→ Fold the paper until it's impossible to fold and unfold it to it's original state, it must be comfortable to write/use even then.

→ Spill few drops of water onto the paper, the paper must turn wet and tear.

→ Spill few drops of oil on the paper, the paper must have oil stains on it.

→ Write on the paper spill water, the ~~pen~~ ink/written part must be stained.

→ Apply powder on the oil stained paper, it must remove the oil texture upto max extent.

→ Check
of a
→ Write
still be
→ Write
again
be c
→ Check
writing
the p
→ Pun
→ Stap
→ Hol
be
→ Pas
zero
→ Ch
clea

Negati
→ Tea
it
→ Sm
of

Scanned with CamScanner

→ Check the company of the paper, it must be of a good well known company.

→ Write on the paper fold it and unfold, it must still be clearly visible to read.

→ Write on the paper ~~and~~ scrunch it and again flaten it, whatever is written must be clearly visible.

→ Check the quality of the paper with various writing objects, it must leave marks behind the paper for markers & sketches.

→ Punch the paper, it must punch easily.

→ Staple the paper, it must staple easily.

→ Hold the paper of certain thickness, it must be easily held.

→ Paste pictures on it, it must get pasted with zero glue leakage behind the paper.

→ Check for the paper colour, it must be clearly visible for whichever colour ink written

<u>Negative scenarios for paper:</u>

→ Tear the paper into pieces and stick again, it must not like the original paper.

→ Smell the paper, it should not give any kind of bad odour.

Scanned with CamScanner

→ Dip the paper in the water and write, it must not support. ~~the~~

→ Write on the paper with a waterproof pen and dip in the water, it must not smudge.

→ Pass on the paper into a shredding machine, the paper should not come out as it is.

→ Put on some weight on a paper placed on a top surface, the paper should not fly away.

→ Scrunch a recyclable paper, it must not spring back.

Table:

→ Place a table on a flat surface, it must stand properly

→ Place some loaded objects on the table, it must hold the load.

→ Apply some acetone on the table, the paint/colour of the table fades.

→ Check all its legs are to the floor, they all must be equal.

→ The table has foldable legs, it must fold properly.

→ The table has wheels fixed on its legs, it must slide with ease.

→ 2 Slide
its legs
it prope

→ Place
check

→ Check
certain
be c

→ Chec
files

→ Chec
mul

→ The
mu

→ Th
wh

Negativ

→ For
shou

→ The
of

→ Pr
m

Scanned with CamScanner

→ 2 Slide the table towards a spacious side unfold its legs work on it for a while then fold it properly.

→ Place a hot object on the table and check if it holds.

→ Check if the head of the table is of a certain height from the ground which should be comfortable to the user.

→ Check the table has drawers to keep files and other immediatories.

→ Check the table can / should support multiple users on it.

→ The table must have a space for multiple users to comfortably work on it.

→ The table must be static / stationary when multiple users work on it.

Negative test scenarios for Table:

→ For certain small amount of load the table should not collapse.

→ The table should not stand still when all of its four legs are unequal in length.

→ Properly fold all it's legs and check it must not unfold.

Scanned with CamScanner

→ Close the table drawers and check it must not open.

→ Unfold the table legs and see it should not fold.

→ Spill the water on the table and see it should not absorb it.

Test Scenarios for Chair:

→ Chair must stand properly/still, ~~as~~ all the legs of the chair are equal in length.

→ Adjust Height of the chair ~~is adjustable~~, it must be movable.

→ Adjust the height of chair upwards and sit, it must hold the user.

→ Slide the chair along a plane surface, it must move.

→ Sit on chair, adjust its height lean back and get up, it should be as it is before use.

→ Check the chair that it must give comfort when lent backwards.

→ The chair should have supporters to keep users arms on it.

→ Place the chair on a flat surface and see it should ~~not~~ ~~m~~ stand still.

→ Place
see it

→ lean
support

Negative tes

→ Heighten the

→ Sit on the
must not

Scanned with CamScanner

→ Place the chair on a slope surface and see it should slide down

→ Lean back and see that the back support of chair must strech back too.

Negative test scenarious for chair:

→ Heighten the chair seat and see it must not slide down.

→ Sit on the chair and see that the back support must not strech back.

Scanned with CamScanner

NOTE: Test scenarios are the high level document which does not consist of i/p parameters or navigation steps. Hence it is not used for performing test execution.

09/08/21

Test Cases

→ Test cases are the indetail doc which consists of i/p parameters as well as navigation steps.

→ Test cases are used for performing test execution process i.e., once the test cases are ready thereafter every TE follow the test cases to follow the test execution process.

→ Test Cases consists of all the navigation steps and i/ps as well as scenarios which need to be tested on the appⁿ.

→ In order to write test cases the TE needs the following things:

1. Requirements → to derive the i/ps.
2. Test Scenarios → to understand the complete flow of the appⁿ
3. Test Case template → So that every TE follow the same template to maintain the uniformity among the TEs.

Scanned with CamScanner

Test Case Template :-

Pre-condition:

→ It is the necessary condⁿ which need to be fulfilled in order to perform execution of that test case.

→ It is actually a data created by the test engineer.

→ Inputs cannot be the pre-condⁿ.

→ Actions cannot be pre-condition.

→ Some test cases does not have pre-condⁿs. (Eg:- google search module)

Test-Data :

→ Inputs created by TEs inorder to test that particular test case is known as test-data.

→ URL cannot be test-data.

Scanned with CamScanner

09/08/21 Test Case Template

HEADER

Test Case Name / ID	ReleaseNo_VersionNo_App_Mod
Test Case Type	FT / IT / ST
Requirement #	
Module	
Status	Pass / fail
Severity	critical / major / minor
Release	
Version	
Pre Condition	
Test Data	
Summary	

Scanned with CamScanner

BODY

Steps	Description	Inputs	Expected Results	Actual Results	Status	Comments

FOOTER

Author Name

Date

Reviewed By

Accepted By

Test Case of

Requirements
1.1
1.1.1
1.2
1.2.1
1.3
1.3.1
1.4
1.4.1

Scanned with CamScanner

Test Case of Login Module :-

(S/w req specification (SRS))

Requirements :	login	
1.1	Un	Text Box
1.1.1	Un	3 digits only
1.2	Pwd	TextBox
1.2.1	Pwd	2-4 alp
1.3	OK	Button
1.3.1	OK	Enabled
1.4	Cancel	Button
1.4.1	Cancel	Enabled

Scanned with CamScanner

Test Case Name	Cheeta_3.0_ICICI_login
Test Case Type	Functional Test Case
Requirement #	1
Module	login
Status	(After execution it is decided)
Severity	Critical
Release	Cheeta
Version #	3
Pre-condition	Valid login Registered / Created
Test-Data	Un-123, Pwd-abc
Summary	It is to check functionality of login

Step #	Description	Inputs	Expected Results	Actual Results	Status	Comments
1	Open "Browser"	www.icici.com	"LOGIN" page must			

Scanned with CamScanner

Step#	Description	Inputs	Expected Results	Actual Results	Status	Comments
1	Open "Browser" enter "URL"	www.icici.com	"LOGIN" page must be displayed			
2	Enter following values for "UN"					
	Valid	123	accept			
	Invalid	333	Err Msg "Invalid login" MBD			
	Blank	N/A	Err Msg "login cannot be blank" MBD			
	Alphanum	12a#	Err Msg "Invalid login" MBD			

Scanned with CamScanner

Step#	Description	Inputs	Expected Results	Actual Results	Status	Comments
	Enter following values for "PWD"					
	Valid	ab	accept			
	Valid	abc	accept			
	Valid	abcd	accept			
	invalid	a	Err Msg "Invalid login" MBD			
	invalid	abcde	Err Msg "Invalid login" MBD			
	Blank	N/A	Err Msg "Pwd cannot be blank" MBD			
3	Enter Valid "UN" "PWD" click on "OK"	123, abc	"HOME" pg MBD			

Step#	Description	Inputs	Expected Results	Actual Results	Status	Comments
4	Enter valid "UN" "PWD"	123, abc	All fields must be reset			

Scanned with CamScanner

Step#	Description	Inputs	Expected Results	Actual Results	Status	Comments
4	Enter valid "UN" "PWD" click on "CANCEL"	123, abc	All fields must be reset			

Date	09-08-2021
Author	Test Engineer1
Reviewed By	Test Engineer2
Approved By	Test Lead

Scanned with CamScanner

Test Case for Amount Transfer module :

Requirement	Login ~~Amt Transfer~~ Sender's Acc	
4.1	~~FAN~~ UN	TextBox
4.1.1	UN	3 digits only
1.2	Pwd	TextBox
1.2.1	Pwd	8-4 alp
1.3	OK	Button
1.3.1	OK	Enabled
1.4	Cancel	Button
1.4.1	cancel	Enabled

Requirement	Amt Transfer	
4.1	FAN	TextBox
4.1.1	FAN	11 digits only
4.2	TAN	TextBox
4.2.1	TAN	11 digits only
4.3	Amount	Text Box
4.3.1	Amount	Between 1-50000 digits only
4.4	Transfer	Button
4.4.1	Transfer	Enabled
4.5	cancel	Button
4.5.1	cancel	Enabled

Test Case N

Test Case

Requireme

Module

Status

Severity

Release

Version

Pre-con

Test-Date

Summary

Scanned with CamScanner

Test Case Name	Kalinga_5.0_ICICI_AmountTransfer
Test Case Type	Functional Test Casing
Requirement #	4
Module	Amount Transfer
Status	
Severity	
Release	15
Version	5
Pre-con	Two valid accounts must be registered/created and the sender's account must have valid amount for transfer
Test-Data	UN-123, Pwd-abc, FAN-1111111111, TAN-2222222222
Summary	It is to check the functionality of Amount Transfer module.

gits

Scanned with CamScanner

Step #	Description	Inputs	Expected Results	Actual Results	status	comments
1	Open "BROWSER" enter "URL"	www.icici.com	"LOGIC" page must be displayed			
2	Enter following value for "UN" valid	123	accept			
	Enter following value for "Pwd" valid	abc	accept			
3	Enter valid "UN" "PWD" click on OK	123, abc	"HOME" page must be displayed			
4	click on Open "AMOUNT TRANSFER"		"AMOUNT TRANSFER" (window) page must be displa-yed.			

Scanned with CamScanner

4	click on Open "AMOUNT TRANSFER"		"AMOUNT TRANSFER" (window) page must be displa-yed.
5	Enter the following values for "FAN"		
	valid	1111111111	accept
	invalid	11111111	Err Msg "Invalid ~~ip~~ ~~for FAN~~ Acc No" mbd
	Character	sahr	Err Msg "Acc No cannot have letters" mbd
	Special char	@*/-	Err msg "Invalid Acc No" mbd
	Alpha-numeric	ab111$	Err msg "Invalid Acc No" mbd
	more than 11 digits	111111111111	Err msg "Acc No cannot exceed 11 digits" mbd
	Blank	N/A	Err msg "Acc No cannot be blank" mbd
	Acc No which doesnot exist	12345762901	Err msg "Acc No doesnot exist" mbd

Scanned with CamScanner

Step #	Description	Inputs	Expected Results	Actual Result	Status	Comments
6	Enter the following values for "TAN"					
	valid	22222222222	accept			
	invalid	22222	Err Msg "Invalid Acc No" mbd			
	character	Arund	Err Msg "A/c No cannot have letters" mbd			
	Special character	!$_/1	Err Msg "Invalid Acc No" mbd			
	α-numberic	cd111@	Err Msg "Invalid A/c No" mbd			
	more than 11 digits	222222222222	Err Msg "Acc No cannot exceed more than 11 digits" mbd			
	Blank	N/A	Err Msg "A/c No cannot be blank" mbd			
	A/c doesnot exist	4231672459 13	Err Msg "A/c No doesnot exist" mbd			
	Same A/c No as PAN	1111111111	Err Msg "TAN cannot be same as PAN" mbd			
7	Enter the following values for "Amount"					
	valid	100	accept			

mbd

7	Enter the following values for "Amount"		
	valid	100	accept
	invalid	0	Err msg "Invalid No" mbd
	i/p no exceeding more that the limit	55000	Err msg "Amt No must not cross more than 50000" mbd
	Blank	N/A	Err msg "Amt cannot be blank" mbd
	Character	FIVE	Err msg "Invalid Amt No" mbd
	α-numeric	RUPEES50	Err msg "Invalid Amt" mbd.
	A/c having insufficient balance		Err msg "Insufficient bal" mbd.
	A/c has exceeded dail transfer limit		Err msg "Daily transfer limit exceed" mbd

Scanned with CamScanner

Step #	Description	Inputs	Expected Results	Actual Results	status	Comments
8	Enter valid i/ps for "FAN" "TAN" "AMOUNT" and click "TRANSFER"	1111111111, 2222222222, 100	"Amt of Rs.100 is debited from your account" mbd			
9	Enter i/p for following "FAN" "TAN" "AMOUNT" and click "CANCEL"	1111111111, 2222222222, 100	All fields must be reset			

Date	09-08-2021
Author	Test Engineer1
Reviewed	Test Engineer2
Approved by	Test Lead

10/08/21

Test Case Name ~~Cheeta 3.0~~ Kalinga 5.0 ICICI Amount Transfer

Test Case Type Integration Test casing

10/08/21

Test Case Name	~~Cheeta 3.0~~ Kalinga_5.0_ICICI_ AmountTransfer
Test Case Type	Integration Test casing
Requirement #	4
Module	Amount Transfer
Status	
Severity	
Release	15
Version	5
Pre-condn	
Test-Data	Un-123, Pwd-abc, Un-456, Pwd-def, FAN-1111111111, TAN-2222222222, Amount
Summary	It is to check that the amount transfered from sender's account is received by receiver's account.

Scanned with CamScanner

Step#	Description	Inputs	Expected Results
1	Open the "BROWSER" enter the "URL"	www.icici.com	Login Page must be displayed
2	(Sender) Enter valid "UN", "PWD" and click on "OK"	123, abc	Home Page mbd (sender)
3.	Click on "Amount Transfer"	N/A	Amount transfer Page mbd
4	Enter valid "FAN", "TAN", "Amount" and click on "Transfer"	1111111111, 2222222222, 50	'Amount Transfer Successfully' msg mbd
5	Click on logout	N/A	Login Pg mbd
6	(receiver) Enter valid "UN", "PWD", and click on "OK"	456, def	Home Pg mbd (receiver)
7	Click on 'Balance Enquiry'	N/A	Available Balance should be: previous Bal+50
8	Logout	N/A	Login Pg mustbd.

Date 10-08-2021

Author Test Engineer1

Reviewed by Test Engineer2

Scanned with CamScanner

7	Click on 'Balance Enquiry'	N/A	Available Balance should be: previous Bal+50
8	Logout	N/A	Login Pg must be.

Date 10-08-2021

Author Test Engineer1

Reviewed By Test Engineer2

Approved By Test Lead.

Scanned with CamScanner

Login (sender) → AT → Bal Enq → Statement → Logout
Login (receiver) → Bal Enq → Statement → Logout.

Test Case Name	Kalinga_5.0_ICICI_AmountTransfer
Test Case Type	System Test Casing
Requirement #	4
Module	Amount Transfer
Status	
Severity	
Release	15
Version	5.0
Pre-condn	
Test-Data	UN-123, Pwd-abc, UN-456, Pwd-def, FAN-!!!!!!!!!!!, TAN-!!!!!!!!!!, Amount-
Summary	

Description	Inputs	Expected Results
		must be

Scanned with CamScanner

Step#	Description	Inputs	Expected Results
1	Open "Browser" enter "url"	https://	Login Page must be displayed.
2	(sender) Enter valid "Un", "Pwd" and click "OK"	123, abc	Home pg mbd
3	Go to "Amount Transfer"	N/A	"Amount Transfer" pg mbd.
4	Enter "FAN", "TAN", "Amount" and click "Transfer"	1111111111, 2222222222, 100	"Amount transfer success-ful" msg mbd.
5	Go to "Balance Enquiry"	N/A	"Balance Enquiry" pg mbd. Available balance should be: (previous) current balance - ~~100~~
6.	Go to "Statements"	N/A	"Statements" pg mbd. It should show the record of latest/recent transaction. (transferred amt)
7.	Click "Logout"	N/A	Login pg mbd.
8.	Enter valid 'UN', Pwd and click "OK"	456, def	'Home' pg mbd.

Scanned with CamScanner

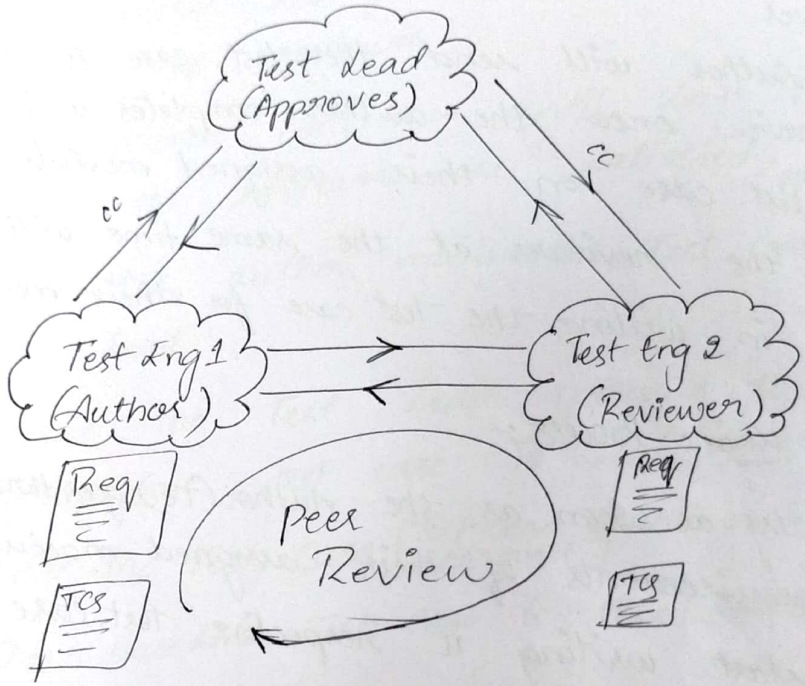
Step#	Description	Inpts	Expected Results
9.	Go to "Balance Enquiry"	N/A	Available Balance should be: previous bal + 100
10.	Go to "Statements"	N/A	"Statements" pg mbd. It must show recently credited amount history.
11.	Click "Logout"	N/A	"Login" Pg mbd.

Date	10-08-2021
Author	Test Engineer1
Reviewed by	Test Engineer2
Approved by	Test Lead

Scanned with CamScanner

10/08/21

Test Case Review Process



→ Once the test Cases are written and before performing test execution process, we must have done one round of Reviewal and Approval Process, so that:

a) The Author will be serious about writing the Test Case.

b) while writing the Test Case they will not miss any scenarios/inputs/navigation steps.

Scanned with CamScanner

NOTE :

1. Every Test Engineers are Author as well as Reviewer.
2. The Author will send the test case to reviewer for review once the author completes writing the test case on their assigned module. Because the reviewer at the same time will be busy in writing the test case for their module.

Test Case Review Process :-

→ Initially as soon as the Author (TE) goes through the requirements of their assigned module, they start writing it respective test case.

→ Once the author has completed writing the whole test case mails it to the reviewer.

→ The Reviewer (TE) reads the test case finds out the requirement number. Then the reviewer logs into the Requirement Repository and opens that particular requirement.

→ The reviewer carefully reads and understand the requirement and acquire all the required flow, inputs and expected result from the requirement

→ If the
mails t

→ The au
it to

→ This p
and

→ Then
Test d

→ Once
Case,
Base d

→ This
Test Ca

→ This
Review

Scanned with CamScanner

→ If the reviewer finds any mistakes he/she mails the same to the author. (writes the comment)

→ The author rectifies the mistake and mails it to the reviewer.

→ This process continues until both the author and the reviewer are satisfied. (peer review)

→ Then the author mails the Test Case to the Test Lead.

→ Once the Test Lead approves the Test Case, that test case is called to be Baseline Test Case. [Test Case which is reviewed and approved]

→ This Baseline Test Case is then saved in the Test Case Repository.

→ This whole process is called as Test Case Review Process.

Scanned with CamScanner

Test Case Review Template

Test Case ID	Step	Severity	Review Comment	Author Comment
Kalinga_5.0_ICICL AmountTransfer_ FunctionalTest	N/A	Critical	Pre-condition is missing	fixed
	5	Major	Not checked with character values	fixed
	5	Major	Not checked for 13 digits	Not fixed (since already checked with 12 digits)
	6	Critical	Not checked with "Same as FAN"	fixed

Ethics to perform

1. The reviewer in the test responsible to should do the required.
2. The reviewer test coverage scenarios i.e., all the imp steps.
3. ~~The~~ for mistakes and review
4. look for

Scanned with CamScanner

Ethics to perform Test Case Review:-

1. The reviewer is responsible to find mistakes in the test case but the reviewer is not responsible to give the solution. The author should do the changes in the test case if required.

2. The reviewer should check for the maximum test coverage, all steps as well as missing scenarios i.e., the reviewer should check for all the important inputs and navigation steps.

3. ~~The~~ For mistakes in the test case both author and reviewer are equally responsible.

4. Look for the content not to the author.

Scanned with CamScanner

11/08/21

Test Case Design Techniques

→ These are the approach / method or technique followed by every Test Engineer to derive the inputs for the component.

→ These techniques are applicable in the components level to derive the inputs.

→ With the help of these techniques we can make the test cases process oriented rather than person oriented.

→ Advantages:

- It helps us to achieve the Maximum Test coverage.
- It makes the test cases process oriented. which helps us to maintain the uniformity.

→ The different types of test case design techniques are:

(i) Error Guessing

(ii) Equivalent Partitioning

(iii) Boundary Value Analysis.

i/ps for Amoun
limit is from
i/ps/valid → 20
invalid → 0.
Blank → N
Sp-cha → @
α-numeric →
Exceeding limit →
daily limit
char
insufficient Balanc

(i) Error Gue

→ In this
based
-ement. It
it is p

→ We go
-onents
Test Co

→ Disada
- It
- Som

Scanned with CamScanner

i/Ps for Amount [ ] (component)
limit is from 1 to 5000

i/ps/ valid → 200
invalid → 0.
Blank → N/A
Sp-cha → @&-
α-numeric → RUPEES20
Exceeding limit → 5500
daily limit →
char → TWENTY
insufficient Balance →

→ Varies person to person
→ fails in MTC
→ person oriented

(i) Error Guessing Technique:

→ In this technique the TE guesses the value based on their understanding over the requir-ement. It has no rules to be followed hence it is person oriented.

→ We go for this technique for all the comp-onents so that we can acheive the Max Test Coverage.

→ Disadavantages:
- It is person oriented.
- Some important i/ps can be missed / skipped.

Scanned with CamScanner

(ii) Equivalent Partitioning: (helps to reduce the no of test cases, without compromising the test coverge.)

→ whenever the component is a range of values we go for one valid and two invalid i/ps.

→ whenever the component is a boolean we have to check for both true as well as false condition.

→ If the component is the set of values go for 1-valid and 2-invalid i/ps.

→ In order to check for drop-down we have to test for one-by-one all the i/ps present in the drop-down, i.e., set of values need to be tested individually or one-by-one for each and every values. (we divide set of test condns into a partition we should be considered as test condn itself).

(iii) Boundary Value Analysis: (better compared to equi-Part w.r.t quality)

→ Whenever the components are range of values we have to derive the i/ps by using BVA Technl.

→ In this technique we check the appn for both upper as-well-as lower limits, i.e., if the range is 'a - b' the i/ps are:

(Testing at the boundaries b/w partitions) a-1, a, a+1, b-1, b, b+1

Eg:- for i/ps

Examples:

1. UN [ ] 2 to 6 ch

2. Equival val inva

1. Error Gu

2. PWD 4 digit

1. Error

2. Equiva

3. Amount 10-

1. Error

2. BVA

Eg:- for Amount if its range is 1-5000

i/ps ⟹ 0, 1, 2, 4999, 5000, 5001.

Examples:

1. UN [ ]

2 to 6 ch → BVA → v, va, vaa, vaana, vaanaa, vaanaan

2. Equivalent Partitioning Technique:

valid → vaan

invalid → 1v2aa3n, 4527/Blank

1. Error Guessing: (valid) vaan, blank, (invalid) 123@v, @$, vaanaaa, a

2. PWD [0-9][0-9][0-9][0-9] 0001-9999 ↔ BVA → 80 EP | invalid | valid | Invalid

4 digits — doesnt exits

1. Error Guessing: (valid) 1435, 13, 14356, saara, blank, (invalid) 72@huli, ~~$##@~~

2. Equivalent Partitiong: (valid) 1435, 14356, (invalid) 72@h

3. Amount [ ]

10-200

1. Error Guessing: (valid) 200, 2000, 5, Rs200, blank, @20

2. BVA: 9, 10, 11, 199, 200, 201

4. Gender o female o male o trans

E.P

each and every ~~module~~ thing must be checked

5. DOB [1-31] [1-21] [1947-2021]

1. Error Guessing: 01-10-1999 (valid), 01-10, 10-1999, 01101999, one-Oct-One thousand nine hundered and ninety nine, blank

(calendar) (fixed)

2. Equivalent Par: 01-10-1999 (valid), blank, 01101999 (invalid)

(drop down)

3. BVA (range)

6. State [ ] (29)

1. Error Guessing:

2. EP: All the 29 components in the drop-down must be checked.

7. Forget Pwd link

- fixed
- Error Guessing

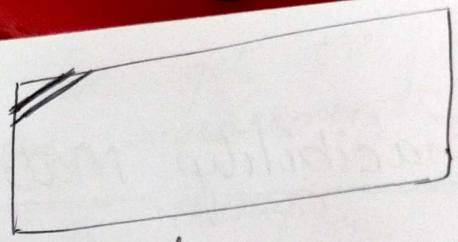
8. Skills ☐ MT ☐ Java ☐ SQL ☐ Selenium

E.P

9. Text A

BVA

9. Tent Area



BVA

Max 1000ch

Scanned with CamScanner

18/08/18

Requirement Tracibility Matrix (R.T.M) / Cross Reference Matrix (C.R.M)

→ As soon as stored in

→ The TE start prep reqs.

→ While w difficulties -ologies, f

→ The TE Test Case for both

→ But at the Reviewal

→ During will not case be review s to them

→ During the missin so many and ev

→ As soon as we get the requirements it is stored in the requirement repository.

→ The TE will go through the requirements and start preparing the Test Cases based on the reqs.

→ While writing the Test Cases may find some difficulties (difficulties i.e., of understanding Termin-ologies, functionalities or dependency).

→ The TE hence decides to skip/postpone that Test Case and continue writing test cases for other requirements.

→ But at the end the TEs forget to write the skipped Test Case and send it for Reviewal and Approval.

→ During reviewal process the reviewer TE will not able to track the missing test case because the job of the reviewer is to review only the test case which is mailed to them.

→ During approval, the TL too not able to track the missing test case because he has to approve so many test cases. Hence he cannot open each and every req and track it.

Scanned with CamScanner

So the TL will review only the basic and critical Test case and approve them.

→ The test cases once reviewed and approved are stored in the test case repository.

→ The test execution is done by picking the test cases from the test case repository and execute its step.

→ The test execution report is prepared based on the status of the test cases. during preparation of test execution report also the missing test case is not tracked.

→ Finally it (app^n) is moved to the production server where the end users start using the app^n.

→ While using the app^n the end user may find lot of issues related to that req.

→ This is the problem which is caused due to the fault of the test engineer. Hence it is a bad name for the company as well.

→ So before performing test execution process we prepare RTM document.

→ It is
-nts u
every T
modules.

→ All th
TE are
the RTM

→ The T
cases
test
it o

dimitatio

→ Some t
types
fail.

Scanned with CamScanner

→ It is a document which maps the requireme-nts with the written test cases. Hence every TE prepare RTM on their assigned modules.

→ All these individual RTMs prepared by the TE are sent to the TL. The TL consolidate the RTMs into single consolidated RTM.

→ The TL will cross check whether all the test cases are written or not. If any dummy test case is there the TL can easily find it out.

<u>Limitation of RTM</u> :-

→ Sometimes we need to write multiple TCs and types of test cases in such case the RTM may fail.

Scanned with CamScanner

RTM / CTM Template :-

Req No	Module	High Level Requirement	Low Level Requirement	TCID/ TCName
4	LOAN	4.1 Personal Loan	4.1.1 Loan for Business	TC_049
			4.1.2 Loan for car	TC_073
			4.1.3 Loan for massiage	TC_061
		4.2 Education Loan	4.2.1 Education in India	TC_055
			4.2.2 Education at Abroad	TC_033
		4.3 Home Loan	4.3.1 Loan for Land	TC_011
			4.3.2 Loan for Individu-al house	TC_016
			4.3.3 Loan for flat	TC_021

Test

→ The process test cases execution pick each written co

→ If the same tho "passed".

→ If all said to

→ If any case is

→ When deviates failed.

NOTE :- Sometime particular test case

Scanned with CamScanner

Test Execution Process

→ The process of executing the steps of the test cases on the build is known as Test Execution Process that is in this process we pick each test case and follow the steps written on it.

→ If the expected result and actual result is same that particular step is said to be "passed".

→ If all the steps are passed the test case is said to be "passed."

→ If any one of the step is failed the test case is said to be failed.

→ When the Expected Result and Actual Result deviates that particular step is said to be failed.

NOTE:-

Sometime if one or two minor step is failed that particular test case is said to be "partially passed test case."

Scanned with CamScanner

Test Execution Report

→ Test Execution Report talks about the stability of the appⁿ, that is no. of test cases written, no. of test cases executed, no. of test cases passed, no. of test cases failed, pass % and fail %.

→ Test Execution Report is prepared after the test execution process.

→ This document is handed to the client too.

No	Module	No of TCs written	No of TCs executed	No of TCs passed	No of TCs failed	pass %	fail %

Req No	Module	No of TCs written	No of TCs executed	No of TCs passed	No of TCs failed	pass %	fail %
1	Login	5	5	5	0	100	0
2	Amount Transfer	20	20	19	1	95	5
3	Bal Enq	10	10	10	0	100	0
4	Statement	15	15	13	2	92	8
5	Loan	24	24	23	1	97	3
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
n	Logout	3	3	3	0	100	0
Total		250	250	230	20	92	8

TCs ~~failed~~ passed

TCs failed

Scanned with CamScanner

13/08/21

Software Testing Life Cycle (STLC)

Software Testing Process / STLC / start to end process / Testing process.

System Study — TL/TM

↓

Test Plan — TL/TM

↓

Test Doc — TE

↓

RTM / CRM — TL

↓

Test Execution Process — TE

↓

Bug Life Cycle — TE (Defect Tracking)

↓

Test Execution Report — TL

↓

Retrospect Meeting — testing team

Test Doc:
- Understand the Reqs
- ↓ Write Test Scenarios
- ↓ Write Test Cases
- ↓ Review & Approval
- ↓ Store the TCs in TC Repository

→ STLC
→ It is
and
→ Initia
test
~~Start~~
- and
→ On
prep
the
fac
→ This
so
act
→ The
the
they
a) T
a
th
b) R
T

Scanned with CamScanner

→ STLC stands for S/w testing life cycle.

→ It is a step by step procedure to test an appⁿ, and it is a part of S/w development life cycle.

→ Initially once we get the requirements the test lead and test manage do one round of ~~'Market Analysis'~~ 'Requirement Analysis' to underst - and the complexity of the modules.

→ On the basis of their understanding they prepare a document that consists of the testing activities which acts as a factor or future plan for all TEs.

→ This Test Plan is mailed to all the TEs so that every T.E can start their Testing activities as per test plan.

→ The first job of the TE is to prepare the necessary test doc. During this process they follow some procedure:

a) They understand the requirements (Req Doc). and any doubt on it get it clarified by the concerned person.

b) Based on their understanding the TE first prepare a high level doc known as Test

Scenarios which gives the brief idea about the flow of the appⁿ.

c) After the Test scenarios the TEs prepare in detail doc known as Test Cases which can be reffered by any person to test the appⁿ.

d) The entire test execution depends on the test cases hence it needs to be Reviewed and Approved to identify the mistakes.

e) Once the reviewal and approval is done the TCs are stored in a centralised location known as TC Repository so that anyone can access it.

→ Once the build is ready and before perfor-ming test execution the TEs prepare one more important doc known as RTM/CRM so that we easily identify the skipped/missed TCs.

→ While performing test execution process the TEs may encounter some bugs/defects. All those bugs are logged to the concerned dev

→ The TEs to
until all

→ Finally
known as
about the

→ The appⁿ
the necess

→ During this
encounter s
such proble
meeting k
we discuss
overcome it

Scanned with CamScanner

re in
hich
to test

n the
viewed
istakes.

done
ed
that

perfor
me
/CRM
ed/

the
ll those

→ The TEs track the status of each bug until all bugs are fixed and closed.

→ Finally the TEs prepare an important doc known as test execution report which talks about the stability of the appⁿ.

→ The appⁿ is delivered to the client along with the necessary deliverables.

→ During this entire process the complete may encounter some problems. In order to overcome such problems in future we go for a meeting known as Retrospect Meeting where we discuss all those problems and how to overcome it.

Scanned with CamScanner

15/08/21

Regression Testing

Release-1	Release-2	Release-3	- - - - Release-n
6 New Features	4 New Features	3 New Features	
9 TEs assigned	Same 9 TEs assigned	Same 4 TEs assigned	
(6 months)	(4 months)	(4 months)	
240 TCs (New)	170 New TCs 160 Old TCs 330 Total TCs	160 New TCs 120 old TCs_{R2} 110 old TCs_{R1} 390 TCs Total	

Enhancement by client
So code changes done by Dev

- → Addition of new features
- → Deletion of old features
- → Modification of existing features (bug fixes)

Challenges:-

1. Identifing the Impact TCs (Impact Analysis Process)
2. Time Constrain:
   release by release the TCs are increasing and time reduces same.
3. It is a tedious process:
   Monotonous process and human may get bored.

(Automation Testing process)

Scanned with CamScanner

→ whenever costumer ask for any enhancement, ~~that~~ the developer will do some code changes that is either 'Addition of new code' or 'Deletion of the existing code' or 'Modification of the existing code', it may have some impact area.

→ The TE has to identify the Impact area TCs and re-execute it. "The re-execution of the old TCs in the new/current release is known as 'REGRESSION TESTING'."

→ The (regression) old TCs which need to be reexecuted in the current release is known as "Regression Testing." It is done from 2nd release onwards.

→ Regression TCs are choosen based on the dependency.

Scanned with CamScanner

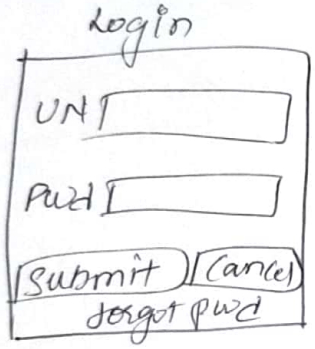
Types of Regression Testing :-

1. Unit R.T:

→ whenever dev does some code changes in any independent indivisual feature in such case we modify the existing TC and re-execute it. There is no impact area [changed features + No impact area]

→ Example :-

- Take a login page of any app.



- Assume that previously the client's require-ments stated the UN must have 4 digits.
- But during next release the client asks for the modification / enhancement that the UN must have 4 chars instead of digits.
- So now based on that when dev does some code changes on Login Module, though Login Module being an independent module and ~~not~~ ~~affe~~ doesnot affect other module, itself can get impacted

NOTE:-

Generally it is a minor changes for which we modify the existing TC but some time we need to write separate new TC-2.

2. Regional R.T:

→ whenever dev does code changes in one feature it may affect a particular region of the appⁿ. Such regression Testing is known as Regional Regression Testing.

[changed feature + Impact area]

→ Example:-

Login
Home
Compose
Inbox
Sent items
Trash
Draft
help
Spam
Logout

- changes done in inbox feature has impact over a particular region.

Scanned with CamScanner

3. Full R.T:

→ whenever the appⁿ is a life critical and machine critical we donot take a risk and once again re-execute the old all existing TCs. It is known as 'Full Regression Testing'. [changed features + complete existing features]

NOTE:-
Donot talk about types of Regression Testing until unless the interviwer ask. RT by default means regional R.T.

Scanned with CamScanner

16/08/21

Regression Testing Process
[Release Level]

Step 1: (when?)

Rel-1

features → 6
TEs → 4
Duration → 6m
TCs → 24

Rel-2 (due to Enhancement)

New features → 4
Old features → 6
Duration → 4m
TEs → 4
New TCs → 170
Old TCs → 160

→ Impact Analysis (say for 15days)

After the product's first release, when the client asks for the enhancement of the product we go for second release. From second release onwards we perform regression testing since there are no old TCs in release-1.

Step-2:

R.A. Meeting ⇒ B.A → dev, TE → Impact

(If the impact is a major risk it is intimated to the client. If it is not a major risk they directly perform)

For the next release we go for Requirement Acceptance Meeting where we discuss the impact of new features on the existing product.

Step-3

After (
Process
by t
① Cl
fe
of
s
S
th
② s
on
a
an
S
a

Scanned with CamScanner

Step-3: (who?)

After RA Meeting we go for the Impact Process (To avoid the Major Risk) where the Impact Analysis is done by three individual teams:

① Client [based on Business Knowledge]

He prepares a report of all the impact features and hands it over to the TL of the company and tells them to be serious/careful about the enhancement.

Say he has prepared a report which states that features [F_1, F_3, F_7] are impacted.

② Developers [Based on Coding Knowledge]

Either it could be old or new devs based on their coding knowledge they also prepare a report of list of features getting impacted and sends it to TL.

Say the devs analysed that features [F_1, F_3, F_6] are getting impacted.

Scanned with CamScanner

③ Test Engineers [Based on Product knowledge]

Since for the previous release ~~whoever~~ whichever TE has worked, they have good product knowledge of the old existing product. So the TEs analyse the impact features and prepares a report based on their analysis and sends it to the TL.

Say TEs analysed that features [F1, F2, F3, F6] are impacted.

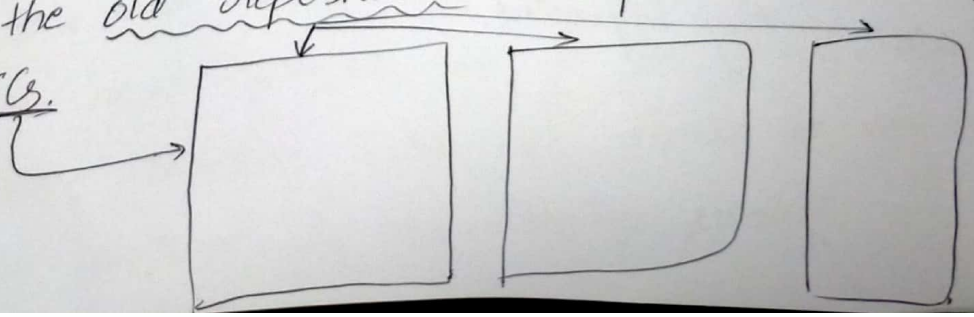
Step-4 :-

All the three individual reports are recieved by the TL. The TL consolidates all the three impact reports into one single impact report.

[F1, F2, F3, F6, F7]

Step-5 :

Based on the Impact Report the TL will go to the old repositories and pick the old Impac -ted TCs.



Step-6 :

Re

Based
Impact T
by the

Step-7 :

As soon
check the
finding
more.

Step-8 :

Once t
execute
by the

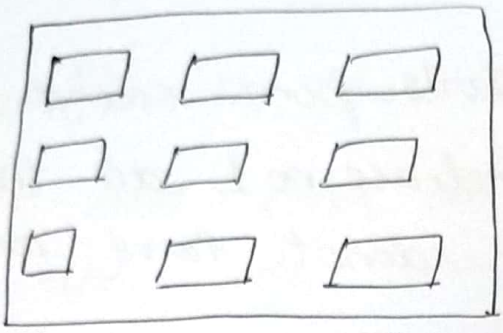
Step-9 :

This p
features

Step-10 : F

Scanned with CamScanner

Step-6:



Regression Test Suit.

Based on the Impact Report all the old Impact TCs are copied into Regression Test Suit by the TL.

Step-7:

As soon as build is ready TEs will first check the new features. Since the probability of finding out defects in new features would be more.

Step-8:

Once the new features are stable, the TEs will execute the old regression TCs which are assigned by the the TL.

Step-9:

This process continues until all the new and old features are stable

Step-10: Finally, we hand over the stable app^n to the client.

Scanned with CamScanner

17/08/21

→ Regression Testing starts from second release onwards. Since in release no 1 all the test cases are new and we donot have any old features to test.

→ whenever costumer asks for some enhance-ment so that the costumer will give new requirements. the costumer, devs and TEs first analyse the requirements and then impact analysis is done.

→ Impact analysis should be done before any code changes at the early stage so that we can avoid the major risk at the initial stage itself.
[Major Risk → main features of an appⁿ getting effected]

IAP (point 4 to point 7)

→ Impact analysis process is generally done by three individual teams:

a) costumer [On the basis of Business knowledge]

- The costumer identifies the features which maybe in a risk for the old features and prepare a impact report and ask the

company to be careful.

b) Developer [On the basis of Coding knowledge].

- The devs either old devs or new devs analyse the code changes and make a plan where exactly the code changes may effect the existing feature's code and prepare an impact area report.

Most Importantly

c) Test Engineers [On the basis of Product know].

- Only the old TEs who worked on the Previous release of the appn has the product knowledge. Hence on the basis of that knowledge the TEs identifies the impact area and create an impact area report.

⇒ All these individual impact area reports are discussed in the R. A. meeting so that the max risk areas are covered.

⇒ All these individual impact area reports are then handed over to the test lead. And TL consolidate these 3 reports into 1 single report.

Scanned with CamScanner

→ Based on the impact area report the TL pick the impacted TCs from the previous release repositories and store it in a separate folder called as Regression Test Suit.

→ Parallely the devs and TEs are busy on new features.

→ As soon as the build is ready the TEs start checking the new features first since new feaultures are newly developed and the probability of finding the bugs will be more in new features.

→ Once all new features are stable the TL assigns the regression TCs to the TEs.

→ This process of executing the old TCs and new TCs continue until all the TCs are passed and the appⁿ is stable.

→ Finally the stable appⁿ is handed over to the costumer.

17/0

Rete

BOO

TE does testi

find

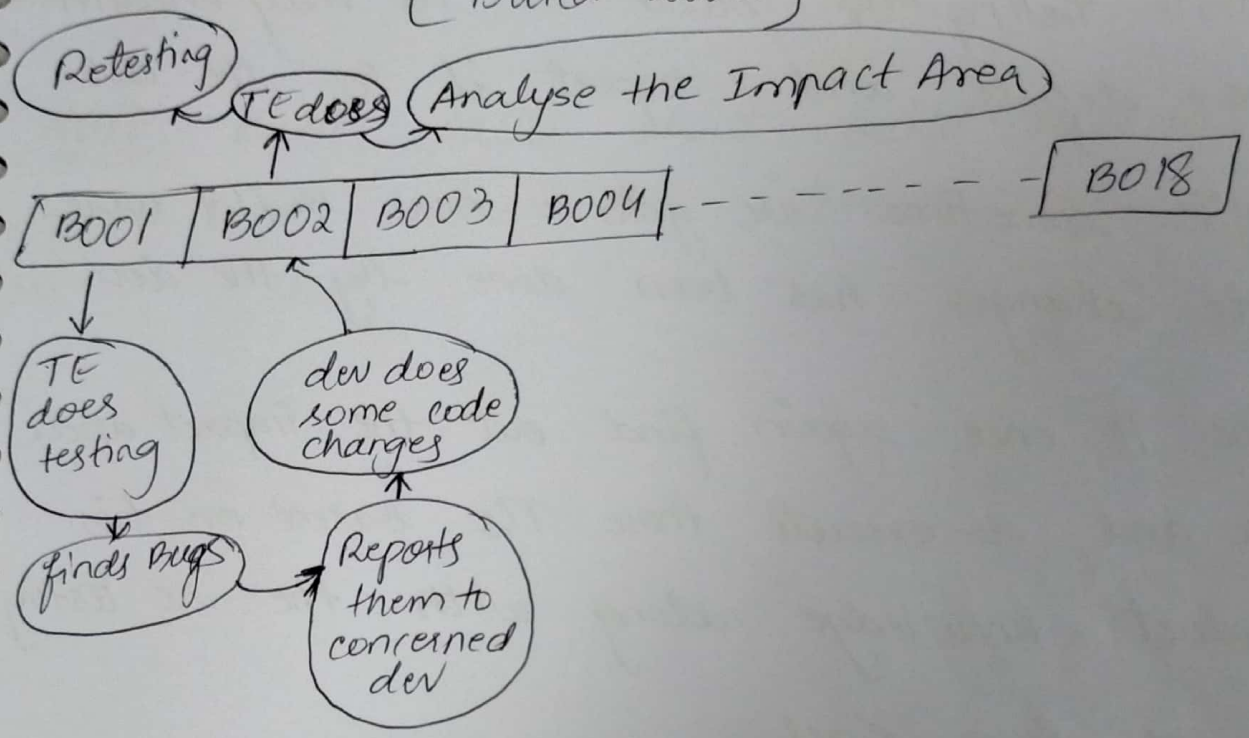
done by TEs only

Scanned with CamScanner

17/05/21

Regression Testing Process

[Build level]



done by TEs only

→ No old TCs [No Regression Test Suit]

→ TE will identify the impact area TCs and re-execute it.

→ No involvement of dev and costumers.

→ As soon as the build is installed the TEs will start testing the build.

→ while testing the build the TEs may encounter some defects and report it to the dev.

→ After sometime we get a new build were code changes has been done by the devs.

→ The TE once again find out the impact area TCs and re-execute those TCs based on his product knowledge along with the re-testing of the bug fixes.

→ In this complete process no separate Regression Test suit is created, that is, it is an informal testing were the TE re-execute the TCs were they found some impact area.

→ In this complete process the devs are not involved since they are completely ~~in bug~~ busy in bug fixes and costumer cannot involve in it since the costumess are not aware of the changes going on inside the company.

→ Hence by the -dge.

NOTE:- D until an

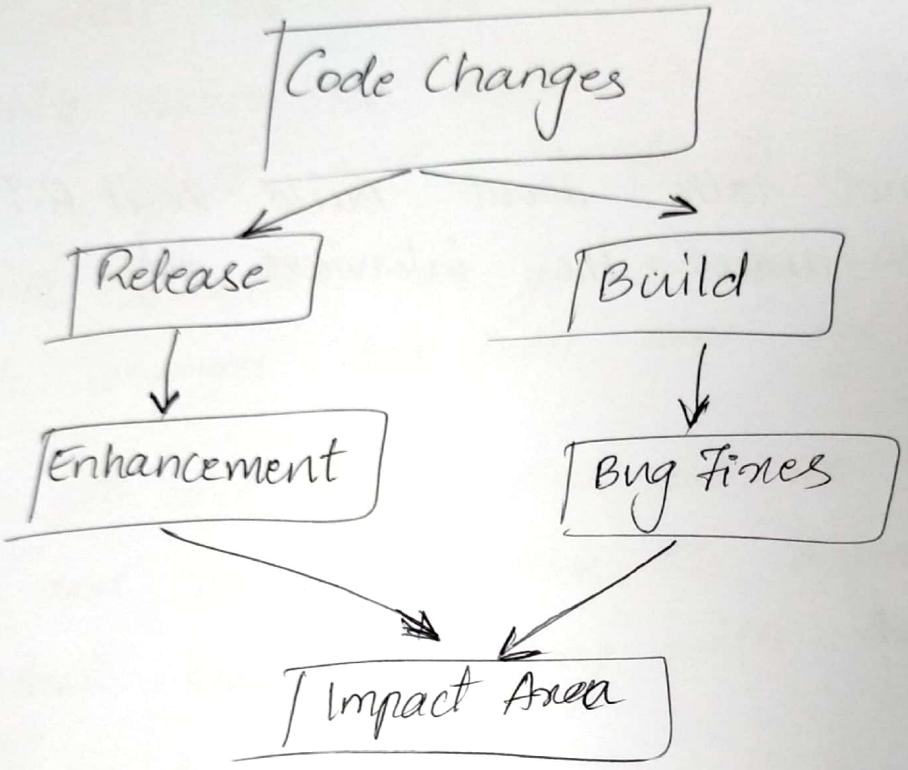
Scanned with CamScanner

→ Hence Build Level Regression Testing is only by the TEs based on their product knowle-dge.

<u>NOTE</u>:- Donot talk about Build Level R.T until and unless the interviwer ask.

Scanned with CamScanner

18/08/21



What is RT?

→ Whenever developer does some code changes across the release or build due to enhancement or bug fixes, it may have some impact area. checking those impact areas is known as 'Regression Testing'.

Why R.T?

→ To overcome the Impact Area problems.

1. What
2. Why
3. Whe
4. Type
5. chal
6. Pro
7. Imp
8. Diff

8.

Re

→ whenev
code ch
release
enhance
it may
impact
impact a
RT
→ In RT
on imp

Scanned with CamScanner

1. What is Regression Testing?
2. Why we go for RT?
3. When do we go for RT?
4. Types of RT?
5. Challenges of RT and how to overcome them?
6. Process of RT?
7. Impact Analysis Process?
8. Difference b/w RT and re-testing?

8.

Regression Testing	re-testing
→ whenever dev does some code changes across the release/build due to enhancement / bug fixes, it may have some impact area. Checking the impact area is known as RT	→ Re-executing the failed TCs to ensure that the bug is fixed or not is known as re-testing
→ In RT we mainly concentrate on impact area	→ In re-testing we only execute the failed TC and it is a part of bug life cycle.

changes
have
pact

Scanned with CamScanner

→ RT can be done with the help of tools (Automation Testing).	→ Re-testing is always done manually.
→ In RT devs and costumers are involved along with the TE.	→ ~~Re~~ re-testing is only done by the TE.

functio
Testing

→ Sele
→ QT
→ Sill

Bug Trac
Tool

→ Jiva
→ Bug
→ Manti

NOTE :- 7

Automation Testing

Functional Automation Testing Tools (By default it is Regression T)

- Selenium
- QTP
- Silk Test

Performance / Non-Functional Testing Tools

- Load Runner
- Jmeter
- Neo Load
- Web-stress

Bug Tracking Tools

- Jiva
- Bugzilla
- Mantish

Test Management Tools

- QC/ALM
- Test Manger
- Test Rail

NOTE:- Functional Automation Testing by default means Regression Testing.

Scanned with CamScanner

① When

6 features 4 TEs Duration 6mo 240 TCs	4 new features 6 old features 4 TEs Duration 4mo 170 New TCs 160 old TCs

→ 3½ months
MTE will work for new features

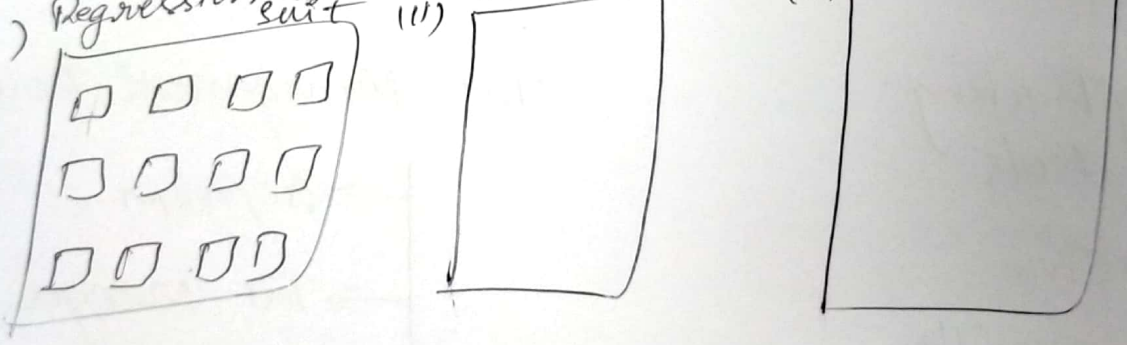
↳ regression test suit.

② who Automation TEs will do the AT

③ Regression Test suit is handed over to ATE.

④ ATE needs three things to convert old TCs into scripts

(i) Regression Test suit (ii) Automation Tools (iii) Previous Release appn



⑤ Convert the regression TCs into script (ATE).

⑥ Once the new features are stable, the ATEs will go for "Batch Execution" [One click execution of all scripts]

⑦ Re

TC
TC
TC
TC
TC-0

⑧ Re

⑨ This
passe

Scanned with CamScanner

⑦ Result

TCID	TSID	status
TC_001	TS_004	passed
TC_002	TS_007	failed
TC_004	TS_009	passed
⋮	⋮	⋮
TC_00n	TS_00n	passed

→ MTE
↓
Verify it manually
↓
If valid bugs present
↓
give it to concerned Dev
↓
code changes are done
(bug fixes
↓
MTE does Impact area analy

⑧ Re-Run the scripts [ATE]

⑨ This process continues until all the scripts are passed.

Scanned with CamScanner

19/08/21

→ Generally we go for Automation Testing whenever there are multiple regression cycle or multiple releases or repetitive task.

→ The process of testing the appⁿ by using a tool is known as Automation Testing.

→ From the release no 2 once we get the enhancement from the client first we do one round of impact analysis and prepare the regression test suit.

→ The regression test suit is handed over to the Automation Test Engineers (A.T.E). Since doing regression testing manually is always time consuming and no accuracy task.

→ From release number 2 all team works parallely like Dev working on new features, MTEs are working on new features and ATEs are busy in writing the test script.

→ In order to write the test script the ATEs needs three things, i.e.;

→ By
finishe
ATEs

→ Once
scripts
starts

→ Bal
of

→ The

→ Once
if
respe
the

→ The
if

1. Regression Test Cases
2. Automation Tool Es
3. Previous Release Appⁿ.

→ By the time Manual Test Engineers (MTEs) finishes working on the new features the ATEs complete writing the test scripts.

→ Once the new features are stable and scripts are ready the Automation Team starts "batch execution."

→ Batch Execution is one click execution of all the test scripts.

→ The tool will generate the report/result.

→ Once we get the automation result and if any of the script is failed the respective test case is handed over to the MTE.

→ The MTE manually verify the bugs and if really a bug exists the MTE report it to the dev.

Scanned with CamScanner

→ The Dev fixes the bug and once again the MTE re-test the bug fixes along with the impact area.

→ Once again the ATE re-execute the script.

→ This process continue until all the new features as well as old features are stable.

Advantages of Automation Testing:-

1. Accuracy exist.
   - Since tool doesnot have any brain so it doesnot get tired/bored of repetitive task and they will not skip any values/scenarios in the test case.

2. Doing regression testing with automation testing is always time saving process. Since execution happen very faster than manual and we can go for batch execution of all the scripts simultaneously.

3. U
use
cod

Limi

1. Au
rele
are

2. N
orde
rele

3. Sor
can
capt

4. Sor
into
man

3. Once the test scripts are ready we can use or execute them whenever Dev does some code changes across multiple releases.

Limitations of Automation Testing:-

1. Automation Testing cannot be performed in release no. 1, since in release no. 1 no old TCs are there.

2. New TCs cannot be automated since in order to write the script the previous release appⁿ is required.

3. Some TCs has human involvement, such TCs cannot be automated such as OTP, dynamic captchas etc.,

4. Some TCs need a lot of time to get converted into script hence these TCs are always executed manually.

Scanned with CamScanner

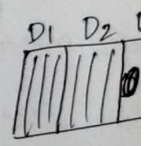
Manual Testing	Automation Testing
→ MT is always done manually, i.e., we give the i/p manually and check the o/p manually.	→ AT is always done with the help of tool, i.e., we give the i/p with the help of script and check the o/p with the help of tool.
→ In order to perform MT, MTE write Test Scenarios and Test Cases.	→ In order to perform AT, ATE writes Test script.
→ MT starts from release 1, once we get the req.	→ AT starts from rel 2, whenever there is repetitive task or multiple releases
→ In order to perform MT very good product knowledge is required.	→ In order to perform AT very good coding knowledge is required.
→ MT is a time consuming process since manually we have to execute all the TCs one by one	→ AT is a time saving process since we can go for parallel execution of all the scripts (batch execution).

20/08/21

Build Ve

Smoke T

Approach-1



Test

Dev → 2 day
fix.

TEs → will

Devs → wil

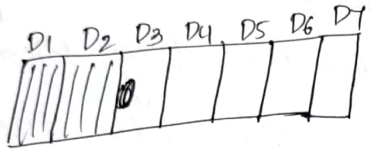
Test Cycle

20/08/21

Build Verification Testing / Dry Run Testing / Smoke Testing / Sanity Testing / Skim Testing

Approach-1

BOOI



Test Cycle (7 Days)

Dev → 2 days of time to fix the Blocker Bug

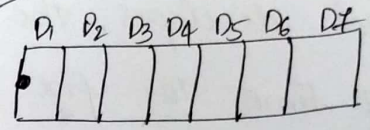
TEs → will sit idle

Devs → will be under pressure

Test Cycle will increase.

Approach-2

BOOI



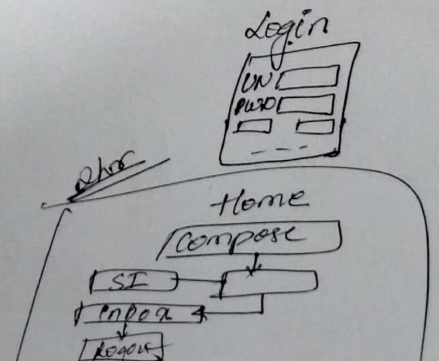
Test Cycle (7 Days)

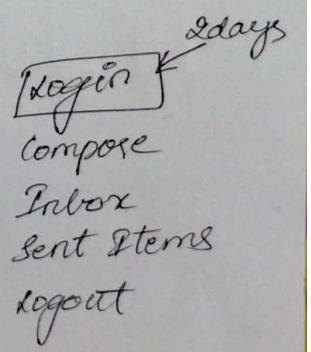
TE found the Blocker Bug in very early stage

TEs will not sit idle

Devs will be not under pressure.

Test Cycle will not increase





Approach-1

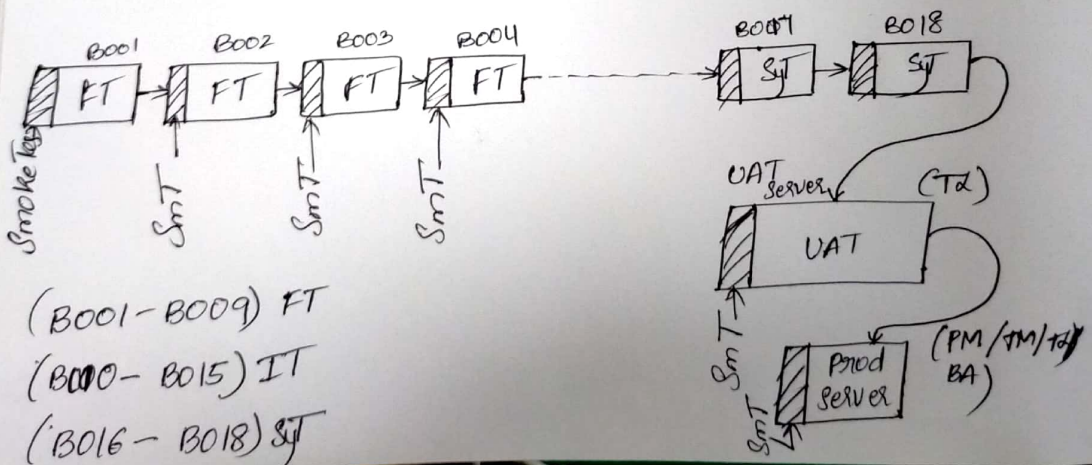
→ When a build is ready with the Test Cycle of 7 days the TE checks for a module (say log in) by applying different combinations of i/ps Say this testing would be done for around 2 days. But after checking for login when the TE checks for compose module but encounters a Blocker Bug, and reports it to the concerned developer. the Dev would take 2 days of time to fix that Blocker Bug. Meanwhile the TEs would be sitting idle due to which there would be extreme pressure on the Dev to fix the B.B as soon as possible. Due to this time lag the Test Cycle too will increase

→ Checking the basic and critical features of ~~an~~ appⁿ before moving into deep rigorous testing is known as Smoke Testing.

→ As soon as the Build is installed the TE start Testing the build with one round of positive i/p to check that the build is testable or not.

→ It is done across all the builds as soon as it is installed, so that we can identify the blocker bug ~~at~~ very early stage.

→ Since in this type of testing we verify the build is testable or not it is also known as Build Verification Testing.



(B001 - B009) FT
(B010 - B015) IT
(B016 - B018) SyT

NOTE :- Sm
avoid the
-Isory typ

Advantages

→ Blocker
initial sta
→ TE will
on other
→ Devs wil
TEs are
→ Test cycle
idle and
time to

NOTE :- (i) Sma
server sin
and it
(ii) Once th
in the
TEs will
ensure

Scanned with CamScanner

NOTE :- Smoke Testing is a good approach to avoid the various challenges, it is not a compu-lsory type of testing.

Advantages of Smoke Testing :-

→ Blocker Bug can be identified at very initial stage.

→ TE will not sit idle since they can work on other independent Testable modules.

→ Devs will not be under pressure since the TEs are not sitting idle.

→ Test cycle will not exceed i.e., no-one will sit idle and we can deliver the product on time to the client.

NOTE :- (i) Smoke Testing is also performed in the UAT server since the environment is getting changed and it may causes Blocker Bug.

(ii) Once the appⁿ is deployed into the production, in the production serve too the BA, PM, TMEs TEs will perform one round of smoke testing to ensure no blocker bugs in the appⁿ.

Scanned with CamScanner

Smoke Testing	Sanity Testing
→ We go for Smoke Testing as soon as the build is installed.	→ We go for Sanity Testing as soon as the bug is fixed.
→ In Smoke Testing we verify the build is testable or not.	→ In sanity testing we verify the impact area due to the bug fixes.
→ Smoke Testing can be documented and scripted since we can identify the basic and critical features of the appⁿ.	→ Sanity Testing can't be documented and scripted since we can't assume or predict the bug.
→ Smoke Testing is shallow (→ valid i/ps) and wide (↳ all basic fea)	→ Sanity Testing is deep (all i/ps) and narrow. (↳ only impact area)

Scanned with CamScanner