

Date ____ / ____ / ____

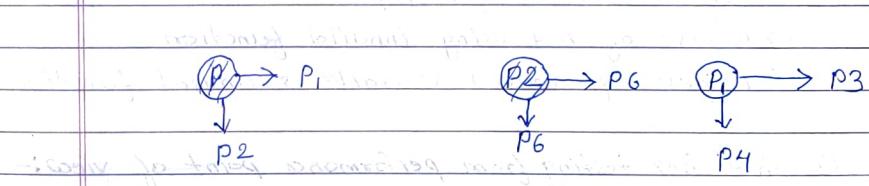
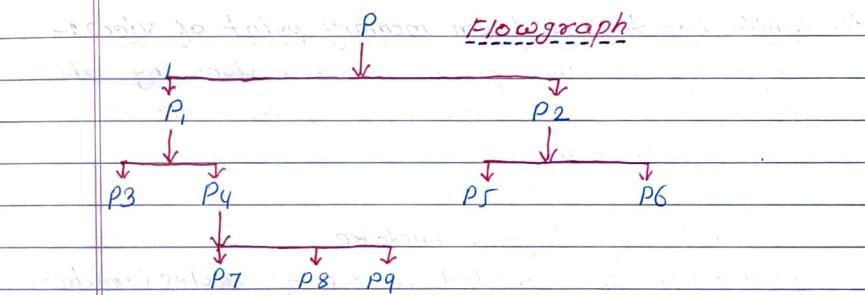
White Box Testing

Testing each and every line of codes is called white box testing. It is done by developer.

Types of white box testing :-

1. Path testing
 2. Condition testing
 3. Loop testing
 4. white box testing from memory point of view
 5. white box testing from performance point of view

1. Path testing:- write the flowgraph and test all the independent paths.



2. Condition testing:- Testing all the logical condition for both true and false values called Condition testing.

Date / /

S/W

Saathi

| | | | |
|---------------|-------|------------------|--|
| \equiv | | | |
| if cond true | | | |
| \equiv | true | Agree & Continue | |
| if cond false | false | Install | |

3. Loop testing:- Testing the loops for all the cycle is called loop testing.

```

1   4
1   3
1 num=9
1 while < 0
1   2
1   n=n+1
1   3
1

```

4. white box testing from memory point of view:- what are all the typical mistakes done by the developer bcoz of which size of the code is getting increased

- a) Because of logical mistake
- b) Because of repeated lines of codes (or) bcoz of not using functions
- c) Because of not using inbuilt function
- d) Because of unused variable & unused function

5. white box testing from performance point of view:- what are all the typical mistakes done by the developer bcoz of which program is consuming more time to perform

- a) Bcoz of not using better logic
- b) whenever we use (OR) in a condition put that condition in the begining which most of the times give result as true.

Date / /

Saathi

c) whenever we use (AND) condition put that condition in begining in which most of the times gives result as false

d) changing the program or condition in order to improve the performance is called Performance tuning

- Q difference b/w white box & Black box testing

white box testing

i) Testing each & every lines of code is called W.B.T

Black box testing

Verifying the functionality of an application against requirement Specification is called B.B.T

ii) It is generally done by developer

It is generally done by test engineer

iii) To do this one should have very good knowledge of programming

To do this no need to have very good knowledge of programming

To do this one should have knowledge of internal lines of codes

To do this no need to have knowledge of internal lines of codes

As early as you catch the bug cost of fixing the bug reduces, If delayed in catching the bug then cost of fixing will increase exponentially

Date / /

SDLC (Software development life cycle)

It is a procedure to develop a Software

Requirement Collection

Analysis

Design

Coding

Testing

Installation

Maintenance

Types of models:-

- 1) waterfall model
- 2) spiral model
- 3) V-model
- 4) Prototype model
- 5) customized / derived model
- 6) Hybrid model
- 7) Agile model

Date / /

1. waterfall model

Requirement Collection

Analysis

DESIGN

CODING

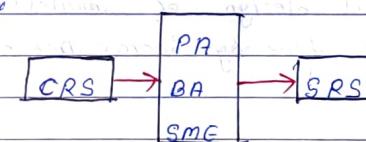
TESTING

INSTALLATION

MAINTAINANCE

a) Requirement Collection:-

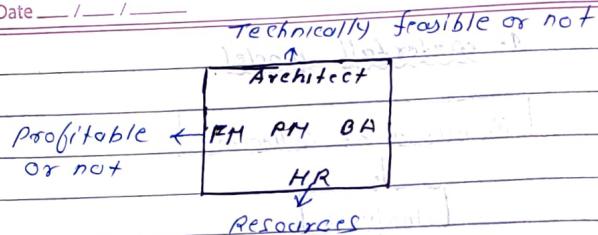
Here either (Business Analyst or) Product Analyst (or) Subject matter Expert will go to customer place and collect the (customer requirement Specification) CRS understand the requirement and convert into Software Requirement Specification (SRS) and explain it to Company



b) Analysis :-

Feasibility study:- It is the study of project where they decide that project is technically feasible or not, financially profitable or not, Resources are available or not.

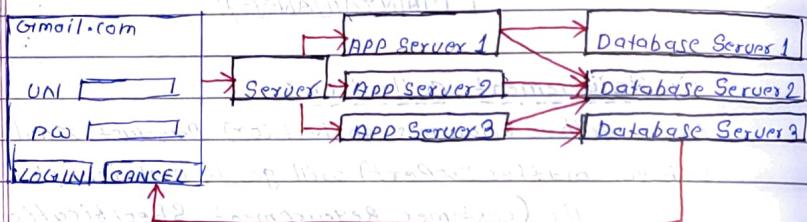
Date / /



c) Design:- Designing of Software is done in 2 steps.

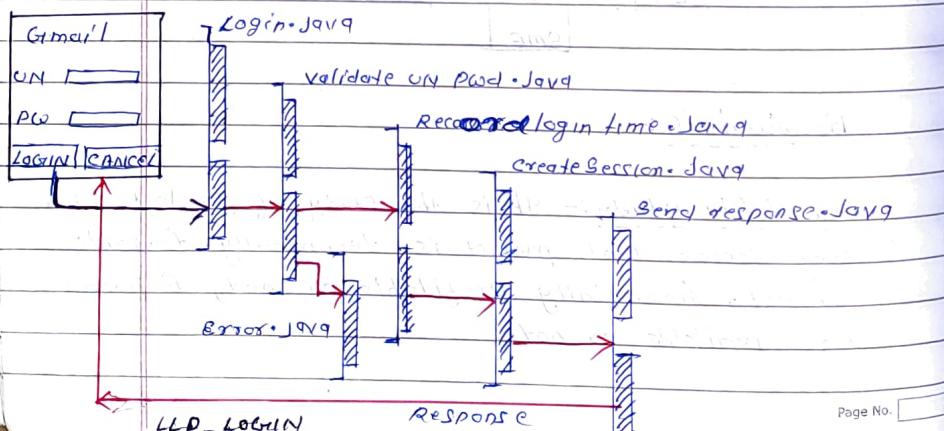
i) HIGH LEVEL DESIGN (HLD)

- It is the design of the architecture of the product.
- It is designed by architect.



ii) LOW LEVEL DESIGN (LLD)

- It is detailed design of smallest feature.
- It is generally done by Senior Developers.



Date / /

LLD Programming :-

Login.java validate UN PWl.java Error.java

| | | |
|----------------------|------------------------|---------------------|
| call validate UN PWl | if UN PWl | throw error message |
| — | { | = |
| — | call record/login time | = |
| — | } | = |
| — | else | = |
| — | { | = |
| — | call error | = |
| — | } | = |

Record/login time .java CreateSession .java Send response .java

| | | |
|---------------------|--------------------|---|
| get Session time | call Send response | = |
| call Create Session | = | = |
| — | = | = |
| — | = | = |
| — | = | = |

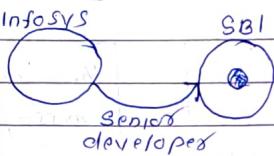
d) Testing:-

Q what are all the drawbacks of developer involve in testing?

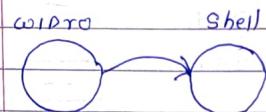
- Ans:-
- i) They will not see the product from negative point of view.
 - ii) They somehow want to build the product.
 - iii) They don't want to see their product breaking.
 - iv) They will consume time allocated for testing to develop S/W & will say no time left for testing.
 - v) Instead of fixing the bug they try to hide the bug.

e) Installation:-

Case-1.

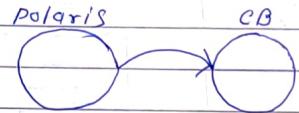


Case-2



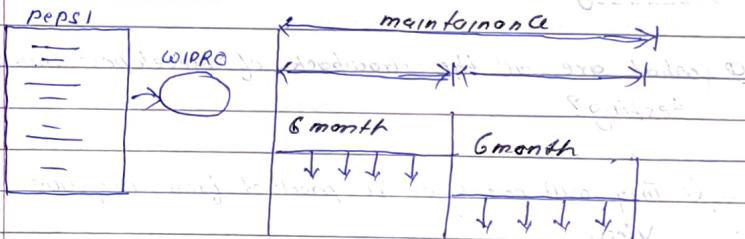
Senior developer → Release Engg., Field engg.
WIPRO → Installation engg.

Case-3



Senior developer → Support engg. or Devops

f) Maintenance:-



The period of fixing the bug free of cost and Period of fixing the bug by paying charges is called maintenance period.

Q In any project why customer keep on changing requirement?

- Ans → 1) Every Software support business & business keeps on changing & when business changes Supporting SW must be changed.
2) Because of market competition.
3) To adapt new technologies.

Q Why we freeze requirement in waterfall model?

- Ans → 1) As requirement changes, corresponding design also changes because of change in design defect will introduced in design.
2) As design changes code changes because of that lots of defects will introduced in code.
3) To avoid it we freeze the requirement.

Q Advantage of waterfall model?

- Ans → 1) It is simple to adopt
2) Initial investment is less
3) We can expect stable product at the end bcoz requirement are frozen in the beginning.

Q Drawbacks of waterfall model?

- Ans → 1) Developers were involved in testing.
2) It is not flexible model.
3) Testing is smallest phase done after coding, if there is any defect in the requirement and design it flows till end and leads to lots of rework bcoz requirement and design phase is not tested.
4) It leads to lots of rework which inturns increase the cost of the project.
5) Time taken to deliver the product to the customer will be more.

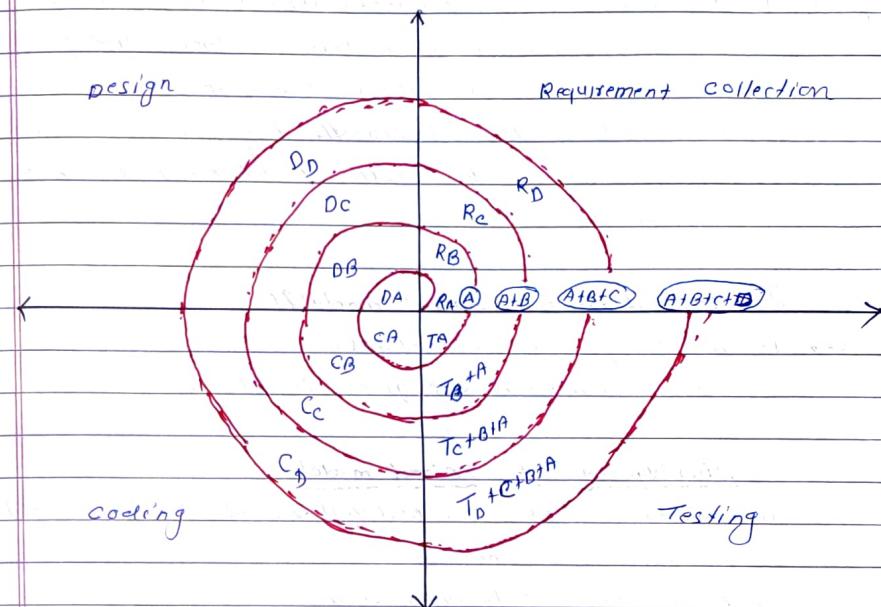
Date / /

Date / /

Q Application of waterfall model?

- Ans →*
- 1) whenever we build smaller or simple applications.
 - 2) whenever we go for short term project.
 - 3) whenever we are sure that customer is not going to change the requirements.

2. SPIRAL MODEL (incremental / iterative)



Q when we deliver project to customer what he will do?

- Ans →*
- 1) He will test the program.
 - 2) He will use it if it is useful.
 - 3) OR else he will keep it and gives requirement for next module.

Q what happens if there is major changes made in B-requirement (60-70%)?

- Ans →*
- 1) Change the design as per requirement changes in B.
 - 2) Coding will be done as per design.
 - 3) Test B features along with bugs.
 - 4) Also check A & C as it may get affected due to change in B.

Date / /

Q what happens if there is minor change in module C?

- Ans. → i) After delivery of product (A/B/C) to customer he may catch some bugs in this product and if he wants to give minor change in C
- ii) The developers have to fix the bug, make the minor changes in C along with development of new module D

Φ Application of Spiral model ??

- Ans. → i) dependency between modules.
- ii) when requirements are given in stages

Disadvantage of spiral model

- i) Each cycle looks like waterfall model so same old drawback comes here
- ii) Testing starts only after coding
- iii) Requirement and design stage is not tested
- iv) Developers are only involved in testing.

Conversion of CRS to SRS

1. C A/C
2. S A/C
3. Loans
 - 3.1 Home loan
 - 3.2 Personal loan
 - 3.3 Vehicle loan
4. Apply loan
 - i) Fixed interest
 - ii) Floating interest
5. Approval loans
 - 5.1 <5000 → Loan manager
 - 5.2 5000 → Branch Manager
 - 5.3 >10000 → Regional manager

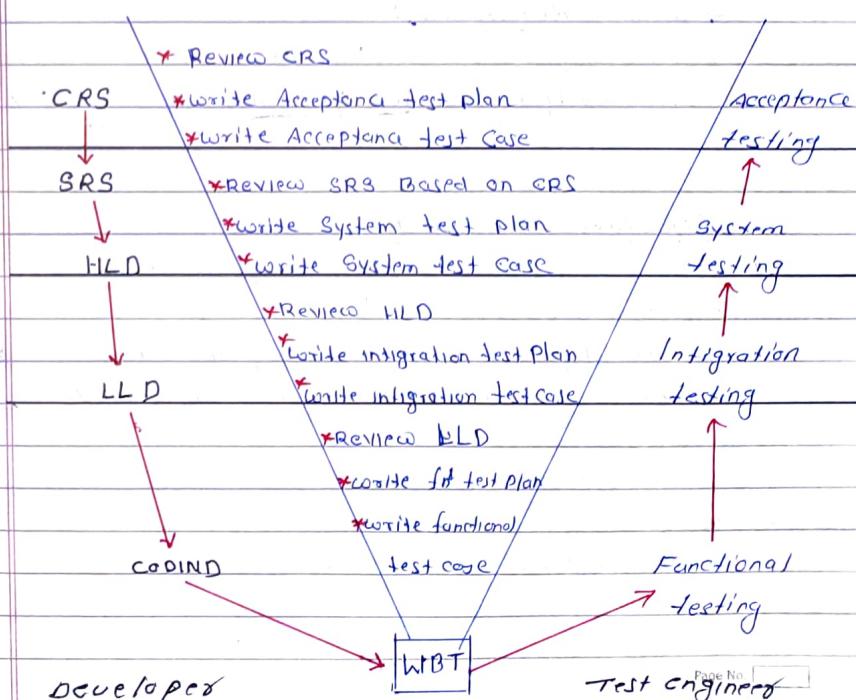
Date / /

| Login page | Home page | Loan page | Home loan |
|------------|-----------|-----------|--------------|
| UN | C A/C | HL | Amt |
| PW | S A/C | PL | i) Fixed |
| | Loans | VL | ii) Floating |
| | | | NPV |
| | | | INTEREST |

| ₹5000 | ₹6000 | ₹10000 |
|----------------------------|----------------------------|--------------------------------|
| loan manager pending loans | Bank manager pending loans | Regional manager pending loans |
| Approve | Approve | Approve |
| Reject | Reject | Reject |

3. V-model

In order to overcome the drawback which was there in waterfall and spiral model we come up with V-model also known as verification & validation model.



Date _____

Date _____

Q) what test engineer reviews in CRS?

Ans

- 1) Conflict in Requirement.
- 2) missing Requirement
- 3) wrong Requirement

1. Conflict in Requirement:

page 10:- Loan manager has got no authority to approve loan

page 15:- if amount < 5000 Loan manager Should approve
if amount > 5000 Branch Manager Should approve
if amount > 10000 Regional Manager Should approve

2. Missing Requirement:

- 1) user can apply for loan ---> approval missing
- 2) user can apply for Insurance ---> " "
- 3) user can apply for OD ---> " "

3. wrong Requirement:

- 1) when apply for loan 10% of fixed interest is charged
- 2) when apply for loan 10% of floating " " "

Advantage of V-model:

- 1) Testing Starts in very early stage of project i.e at Requirement Collection Stage.
- 2) All the stages are tested bcz of this it avoid the downward flow of defect which reduces lots of rework.
- 3) Total cost will be less.
- 4) The O/P is given simultaneously bcz of this project gets completed very fast.

Drawback:-

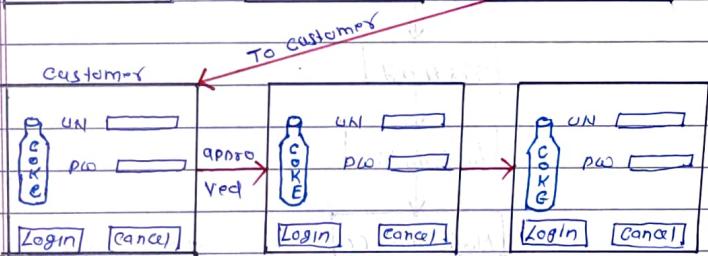
- 1) It involves lots of documentation work.
- 2) Initial investment is more

Applications:-

- 1) whenever we build large or complex application.
- 2) whenever we go for long term project.
- 3) whenever customer is expecting very high quality product & prescribed high time frame.

4. Prototype model:

It is a dummy software (or) an image which looks like a actual software but it's not a actual software.

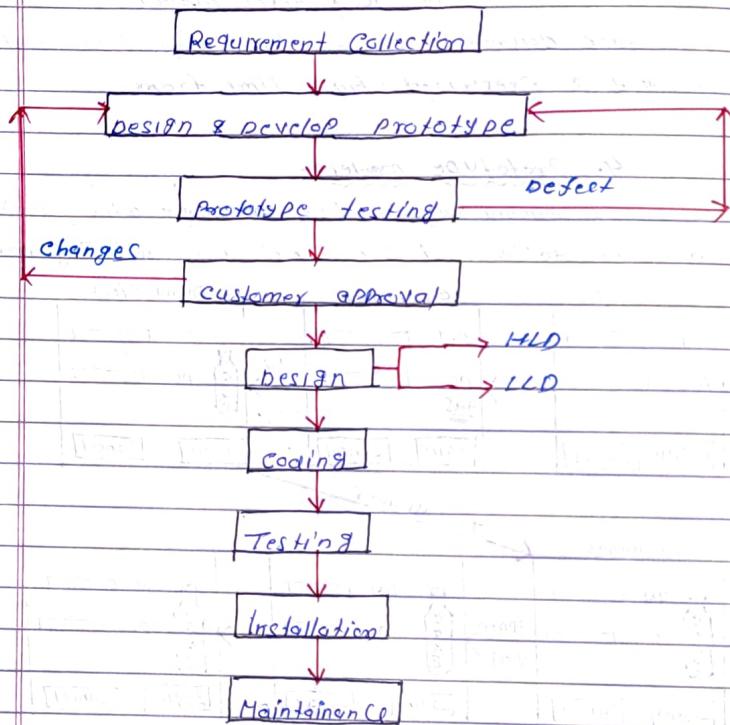


Advantage:

- 1) Improved communication b/w customer, development team and testing team.
- 2) Initially itself customer will get to know what is outcome.
- 3) Initially itself developer will get to know what they have to develop.
- 4) customer gets the opportunity to ask for changes in beginning itself.

Applications:-

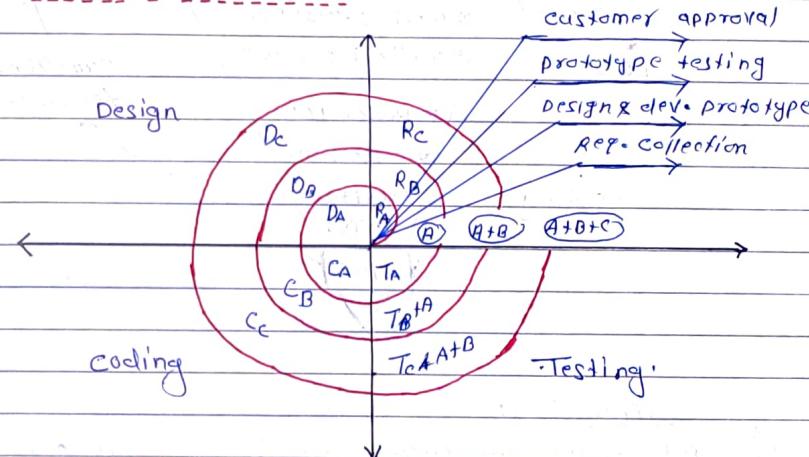
- i) when customer is not clear about explaining his own requirement.
- ii) when customer is new to software development process.
- iii) when developer are new to domain.



they were busy in developing & redesigning prototype.

5. Customized or Derived model

we take basic model and we do some changes as per the Company Standards, and that changed model is called as customized (or) derived model.

6. HYBRID MODEL1. Spiral ModelApplication:-

1. when there is dependency in modules
2. when customer is not clear about explaining his own Requirement

2nd

1. when requirement are given in ~~mod~~ stages
2. when customer is new to Software development process.

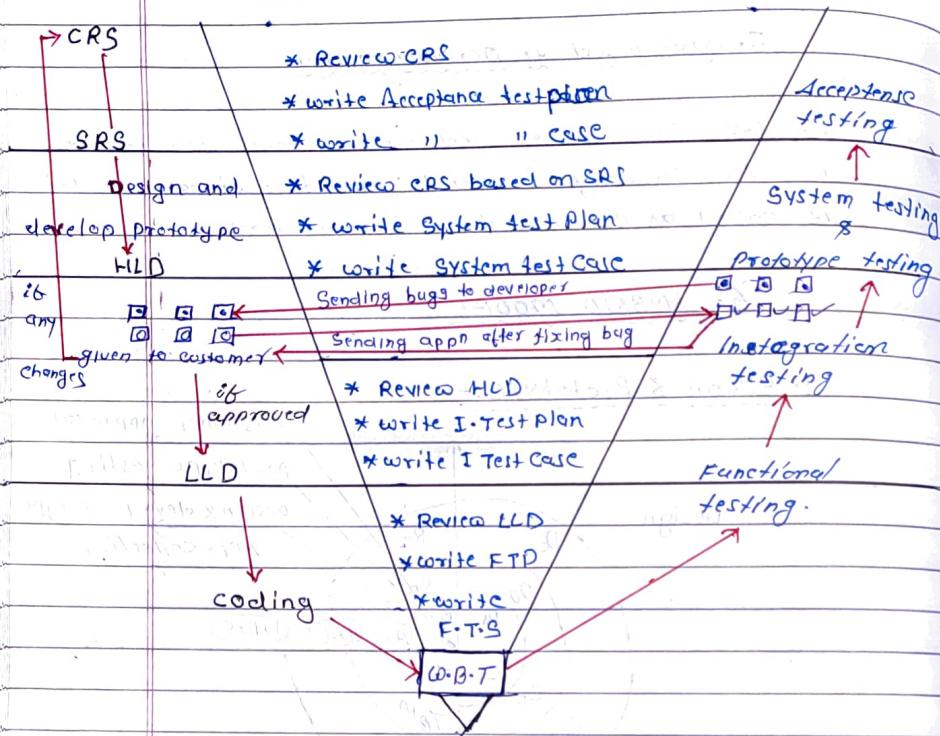
Drawback:-

- i) Investment needed to build prototype.
- ii) Actual development of SW starts late because

Date ___ / ___ / ___

Date ___ / ___ / ___

2. V-8 Prototype:-



Application:-

- 1st
- whenever we build complex or large applications.
 - when customer is not clear about his own requirement.

2nd

- whenever we go for long term Project
- when customer is new to SPM development Process

3rd

- whenever customer is expecting high quality product in a pressurized high time frame
- whenever developer are new to domain

Black Box testing

usability testing / cosmetic testing / GUI / Yellow Box testing

Testing the user friendliness of the application is called usability testing

Q How to do usability testing?

- Ans
- g will check for look & feel of the appn
 - g will check whether appn is simple to understand or not.
 - Important features or frequently used features should be given to user in 3 clicks

Q For what kind of application we should do the usability testing?

- Ans
- Any application used by variety of users
 - Any appn which generates lots of revenue
 - Any appn where we don't provide any training to user about how to use i.e. we are expecting end users to understand appn & use it by themselves

Q When we do usability testing?

- Ans
- Different projects will do usability testing in different ways

a) In some projects they do usability testing when the product becomes functionally stable.

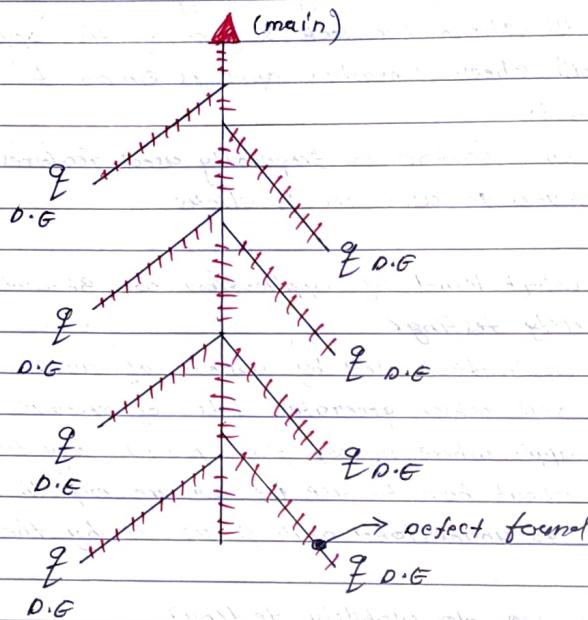
b) In some projects they do usability testing in the beginning itself.

e.g.: Prototype projects

Date ___/___/___

Defect tracking

Test Engineer communicate to Development Engg. through defect tracking tool.

FISHBONE Technique / Ishikawa method

1. while testing S/W if T.E finds any defect, he will communicate with D.F
2. All D.F will sit together and find root causing defect.
3. And they document it and share it in Common folder, so that all developer can access it.
4. And they can present it to everyone
5. This process or technique is called as FISHBONE TECHNIQUE OR ISHIKAWA METHOD.

Date ___/___/___

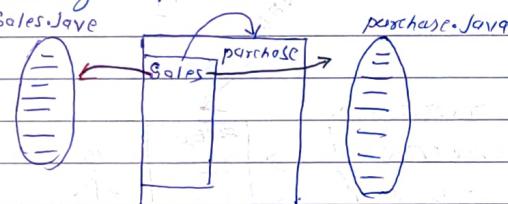
Test efficiency:-

$$T.E = \frac{\text{Total no. of bugs found by T.E}}{\text{Total No. of bugs in S/W}} \times 100$$

Total No. of bugs in S/W
 Total no. of bugs + Total no. of bugs found during Acceptance Testing + No. of defects found by users

Defect:-

1. Any feature which is not working according to the requirement Specification is called defect.
2. Deviation from requirement Specification is called as defect.
3. why we get defect?
 - (Ans) 1) wrong implementation
 - 2) missing implementation
 - 3) extra implementation: when develop an extra feature which was not in Customer requirement Specification



wrong
Coding

Date ___ / ___ / ___

Q what is difference b/w defect, bug, error and failure?

Ans

* **Defect**:- deviation from requirement Specification is called defect.

* **Bug**:- Bug is just an informal name given to defect.

* **Error**:- Error is mistake done by programmer bcoz of which we are not able to Compile or Run the program.

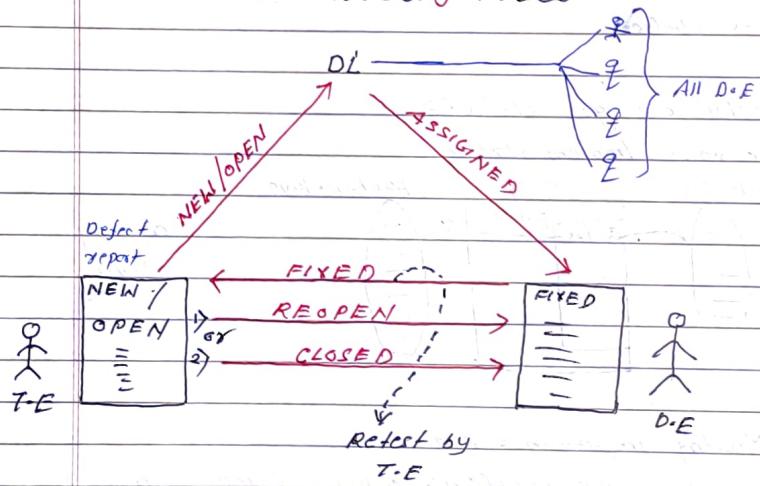
There are 2 types of error

a) CTE:- we get this bcoz of Syntax mistakes

b) RTE:- we get this bcoz of logical mistakes

* **Failure**:- Defect in a SW leads to failure. one defect leads to one or multiple failure.

Defect tracking process



Date ___ / ___ / ___

1. while testing T-E will find the defect.
2. T-E will write defect report on defect tracking tool.
3. Status of Report will be NEW/OPEN and Send to development lead and Cc to team lead.
4. D.L will see the report Search the D.E who was handling this feature & tell him to fix. the status will be changed to ASSIGNED.
5. D.E will go through Report, Then he will go to source code fix the bug and Send back the report to T-E. If status will be changed to FIXED.

6. T-E will test the SW if bug was fixed, he will change the status to CLOSED, but if he found that bug not fixed status will be changed to REOPEN.

Severity

⇒ It is decided based on the impact of defect on the customer business work flow.

It has 4 types:-

1. BLOCKER
2. CRITICAL
3. MAJOR
4. MINOR

1. BLOCKER DEFECT:- (Show stopper)

- a) Assume that there is a defect in the app and 9am 100% Sure that this defect will effect the customer business work flow and also blocks the T-E to test the feature.
- b) Such type of defects are called Blocker defect.

2. Critical Defect:-

- a) Assume that there is a defect in a appⁿ and i am 100% sure this will effect customer business work flow but its not blocking T.E to test the feature.
- b) Such type of defects are critical defect.

3. Major Defect:-

- a) Assume that there is a defect in the appⁿ but i am not sure how this defect will effect customer business work flow
- b) Such type of defects are called major defect.

4. Minor Defect:-

- a) Assume that there is a defect in the appⁿ but i am 100% sure that this defect will not effect the customer business work flow
- b) Such type of defects are called minor defects
eg:- Spelling mistake, object overlapping, alignment issue etc.

Q) Give Severity for ATM machine:-

Ans **Blocker:-** when user insert ATM card into ATM machine and enter valid pin but its not accepting.

Critical:- when user insert ATM Card to ATM Machine and enter invalid pin but its accepting

Major:- If we are not getting any receipt or Screen is blur.

Minor:- If there is any Spelling mistake in the receipt.

Q) Give Severity for given name text field?

| |
|---|
| * NAME <input type="text"/> |
| <input type="button" value="Submit"/> <input type="button" value="Cancel"/> |

Blocker:- Name text field is not accepting any char.

Critical:- Name text field is accepting characters only in the range b/w 8-10

Major:- Name text field not accepting more than 60 characters

Minor:- If there is any mistake in spelling or * Symbol is missing

Priority

If importance given to fix defect.
(or)

How soon defect should be fixed by developer.

There are 3 level of Priority

- 1) HIGH - P1
- 2) MEDIUM - P2
- 3) LOW - P3

HIGH(P1):- If the defect is having priority as P1
then developer should fix the defect immediately

MEDIUM(P2):- If the defect is having priority P2 then
developer then can fix the defect within the
Same test cycle (or) within same build (or)
within release

LOW P3:- If defect is having priority as P3 then
developer can fix the defect in the next
upcoming Release (or) & to 3 release

Q In whatsapp user is not able to video call
Ans → Blocker - P1

Q In flipkart there is Spelling mistake in last line
Ans → Minor - P3

Q In QLA user is ~~not~~ able to book and finish
the ride by using wrong OTP
Ans → Critical - P1

Q whatsapp mate function is not working.
Ans → Major P2

Q youtube user is not able to play video
Ans → Blocker P1

Q In youtube if 1000 Subscriber shows as 500 only
Ans → Critical P1

Q In FB Remember Password feature is missing
Ans → Major P1

Q

1. HIGH SEVERITY & LOW PRIORITY

2. HIGH SEVERITY & HIGH PRIORITY

3. LOW SEVERITY & LOW PRIORITY

4. LOW SEVERITY & HIGH PRIORITY

5. Blocker P3

whatsapp → install → Not able to install 50th time

2. Critical P3

Apply loan → Approve loan → change Seroor →

| |
|-----------------|
| Check Status |
|-----------------|

↓
8%

Date _____

Date _____

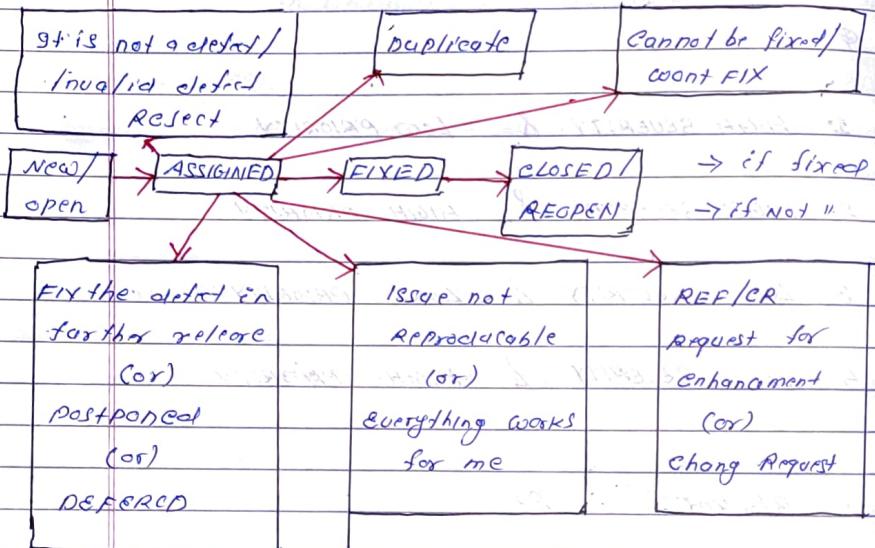
Q who will decide Severity & Priority

Ans → Test engineer will decide the Severity and Priority

VVI → It's impossible to change the priority of the defect after log in into defect tool

VVI → Developer will fix the defect which is high priority rather than High Severity

Defect life cycle (or) Bug life cycle



Q Explain defect life cycle?

1. Test ENGINEER:-

- Test Engg. will find the defect
- Prepare defect report in defect tracking tool
- He will put the status as NEW/OPEN
- He will send the report to development lead

2. development lead:-

- He will read the report & understand the problem
- He will identify the developer who did the mistake
- He will change the status to ASSIGNED
- He will send the report to D.E

3. development Engg. :-

- He will read the report and understand the problem
- He will go to Source Code and fix the bug
- He will change the status to FIXED
- He will send report to T.E and keep CC to D.L

4. Test Engg. :-

- He will read the report and retest the bug
- If bug is fixed, he will change status to CLOSED
- If not fixed, he will change the status to REOPEN
- He will send status to D.E and keep CC to D.L

This process is called as Defect life cycle

Q what is difference b/w NEW/OPEN status?

Ans → For the first time if T.E finds the defect status will be NEW
⇒ when developer starts to work on defect status will be OPEN

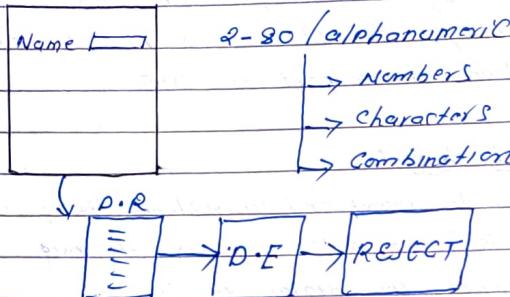
Q what is REJECT Status?

Ans → Test engineer will assume feature itself as a defect
⇒ And send the defect to developer
⇒ developer will say that it's a feature and change the status to REJECT

Date ___ / ___ / ___

Q Why we get Reject Status?

Ans 1. Because of misunderstanding of Requirement



2. Because of Referring old requirement

| Requirement | New Requirements |
|-------------|------------------|
| chat | Coding for |
| status | chat, call |
| calls | Status X |

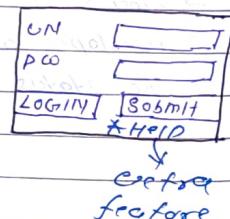
→ After 1 year
Remove the calls status feature

Test Engg - test the app after long leave and he don't see status feature and prepare defect report with blocker and P1 and send to D.E D.E then reject the defect.

3. When you configure S/W wrongly while installing when T.E is not installing the S/W properly and if he get a bug & send to developer then developer will say it is not a defect and tell T.E to install S/W properly

3. Because of extra feature

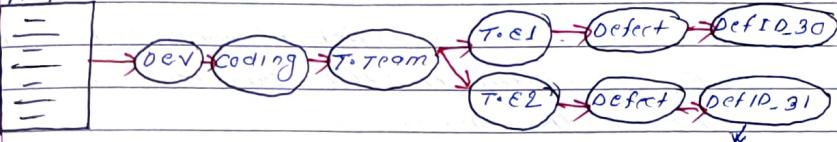
Whenever you consider feature as a bug, developer might Reject. In Such Case T.E should REOPEN the status and ask developer to update requirement



Q What is DUPLICATE Status?

If a defect is sent which is already sent by someone else then developer will change the status to DUPLICATE.

Ans:



It is duplicate of Def ID 30

Q Why developer will say it as duplicate?

Ans 1. To reduce defect count.

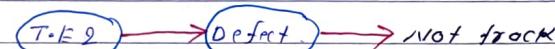
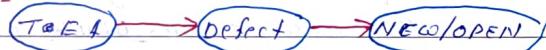
2. To reduce duplicate effort.

Q Why we get duplicate status?

Ans 1. Old T.E would have found too many bugs some of them are still pending of new Test Engg joins the same project and send some old bugs then developer will say it as DUPLICATE

2. Someone might come into your module and catch a bug then send it to developer; If you find same bug and send it to developer he will say it as bug DUPLICATE

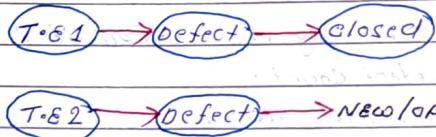
Case 1 -



T.E will find the defect in S/W if the same defect is present already then we should not track a defect.

Case 2.

T.E will find a defect already which is in fixed status in this case T.E should open the defect.

Case 3-

T.E will find defect in s/w and already same defect is present with CLOSED status in this case T.E should track as new defect.

Q How to avoid duplicate bugs?

- Ans 1:** When T.E find bug and communicate to development lead he should keep CC to test lead and all test engg. who are working on the same project.
- 2:** Before T.E log a defect he should search for the duplicate bug in defect tracking tool i.e. in advance search field by entering certain keyword. In this example Sales_Bank page is keyword.

3. T.E when he finds defect before tracking we should cross check with senior T.E or T.L or dev. sometimes so that we can reduce duplicate bugs.

Q what is can not be fixed status?

Ans Developers are accepting that it is a defect and he says can not be fixed due to some reasons.

Q why we get can not be fixed status?

- Ans 1:** If the minor defect is present in the root of the Software and if it is not affecting customer business flow. In such a case developer says can not be fixed.
- 2:** If technology itself is not supporting
- a) If defect is minor or major then developer says can not be fixed if the defect is blocker or critical then developer should change the technology or else you should come up with some alternative solution to develop the same feature.

- 3:** If the cost of fixing the bug is more than the cost of bug.

Q what is cost of the bug?

Ans Loss in the business because of having bug in s/w.

Q what is issue not reproducible (or) Everything works for me status?

Ans T.E is able to see the defect but the developer will be not able to see the same defect in such a case developer will change the status to issue not reproducible.

Q why do we get issue not reproducible?

Ans. 1. Because of platform mismatch.

a) OS mismatch

b) Browser mismatch

c) Browser Version mismatch

d) Browser Setting mismatch

2. Because of improper defect report.

3. Because of incorrect data (or) data mismatch

4. Because of Inconsistent defect:- It means

Some times features will work sometimes Some feature will not work this kind of defect is called as inconsistent defect.

When T.E. when he find this type of defect he should write note in the report saying that defect is inconsistent.

VVI

Q How do you convince developer that it is a defect when developer say issue not Reproducible?

Ans. 1. I will tell the test date, OS, Browser used and also I will write proper procedure.

2. If the defect is not visible I will send Screen shot and also give him demo of how I am getting the bug by asking developer to connect to my computer.

3. Still if the bug is not visible to the developer I will try it in 4-5 computers around me. If it is not visible I will escalate it.

If it is visible I will ask developer to fix the bug.

Q what is postpone (or) defered (or) holding (or) fix the bug in future release?

Ans. developer are accepting that it is a defect but they want to fix the bug little due to some reason

Q why we get postpone status?

Ans. 1. If T.E. find the defect in the end of release and it is minor defect then developer say that they don't have sufficient time & change the status to postpone.

2. If T.E. find the defect in a future which is not required to the customer in the current release then developer say i will fix the bug in future release.

3. If T.E. finds the defect and send it to developer. Developers say that customer is expecting lots of changes in the same feature so, better i will postpone it until I get clarity from customer.

4. If T.E. finds a defect in the feature i.e exposed internal users and if that defect is minor then developer say i will fix it in future release.

Q what is REF/CR?

Ans. There is problem or defect in the app which is not a part of the requirement in such a case they call it as enhancement or request for enhancement or change request.

Date / / Interview Questions:-

1. Defect triage:-

Ans - When time is very less and there are too many pending bugs to be fixed, we will categorize the defect into different groups:-

- we will move all business critical defects to **ASSIGNED status**
- There are some bugs which need not to be fixed on a urgent basis - we move such type of defects to **POSTPONE status**
- There are some defects which need not to be fixed at all as customer is ready to accept the product with those defects. we move such defect to **CANNOT BE FIXED status**

This type of grouping defect is called defect triage

2. Defect masking:-

Ans - One defect hiding or masking another defect is called as defect masking.

3. Defect Seeding:-

Ans Intentionally introducing the defect in the S/W to check the efficiency of T-E is called Defect Seeding.

4. Defect clustering:-

Ans

- when small no. of modules contains most of the bug clustered (or) defect which shows most of the test failures.
- According to paretto principle 80% of the bugs are detected by 20% of the modules.
eg:- HELP, SUPPORT, About us etc.

Date / /

5. Defect pesticide / Pesticide Paradox:-

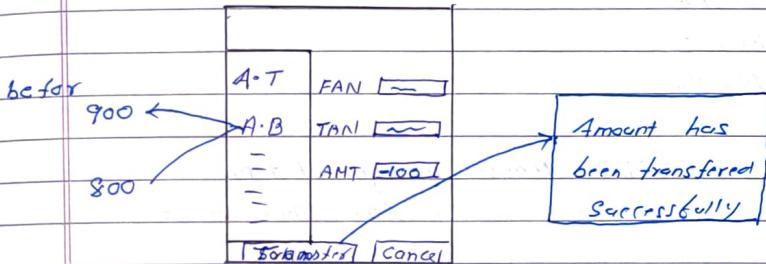
If the same test cases or Scenario's are repeated again and again it will no longer find any new bugs. this is nothing but defect pesticide or pesticide paradox

We can avoid it by

- upgrading the test case
- writing new test cases to the newly added features

6. Defect Cascading:-

Ans One defect or any defect triggering another defect is called defect Cascading

6. Defect leakage:-

Ans If customer or the end user find the defect in the app after release those defects are called as defect leakage

$$\frac{\text{No. of defects found by customer after release}}{\text{No. of defects found by T-E}} \times 100 = \frac{\text{No. of Invalid defect (Reject, duplicate, INR)}}{\text{Total number of defects}}$$

$$\frac{10}{230 - 15} \times 100 \Rightarrow 4.65\%$$

Date / /

Saathi

VVI Defect tracking tools:-

Q What is defect tracking tool?

A Ans- Defect tracking tool is a centralized place where we can track or store the defects and communicate defects to developer in organized way.

BUGZILLA → defect tracking tool

Q CI/ALM → 1. Add requirement 3. Track defect } P.M.T
JIRA 2. write T-CASE }

BUGZILLA

BUGZERO

BUGPRO

BUGNET

Mantis

PMT → Project management tool

ALM → Application Life Management

Q How defect tracking tool works? (BUGZILLA, MANTIS, JIRA)

Ans- 1. Open the browser and enter URL

| | |
|--------------------------------------|---------------------------------------|
| HTTP://BUGZILLA/LOGIN.VSD | |
| UN | <input type="text"/> |
| PW | <input type="password"/> |
| <input type="button" value="LOGIN"/> | <input type="button" value="CANCEL"/> |

2. Enter username and password and click on Login button

Date / /

Saathi

Welcome to BUGZILLA

Dashboard

Search

Create issue Adv. Search

Resolved issue Configured field project names

Open defect project name FB

Closed defect Release Name Gmail

Found by T.E module name Flipkart

Severity Zomato

Priority Swiggy

Summary

Reporter

2. T.E will search for duplicate defects using search text field if he knows defect id if he don't know defect id then he will search for duplicate defect using advance search text field.

| | |
|---------------------------------------|---|
| Project Name | <input type="text" value="Gmail"/> |
| Release Name | <input type="text" value="TIGER"/> |
| Module Name | <input type="text" value="compose"/> |
| Severity | <input type="text" value="Blocker"/> |
| Priority | <input type="text" value="P1"/> |
| Summary | <input type="text" value="Compose-Blank Page"/> |
| <input type="button" value="Submit"/> | <input type="button" value="Cancel"/> |

300 → 150 → 100 → 40 → 10 → 3

| S.No. | Summary | Status | Severity | Priority | Reporter |
|-------|---------|--------|----------|----------|----------|
| 1 | ~ | ~ | ~ | ~ | ~ |
| 2 | ~ | ~ | ~ | ~ | ~ |
| 3 | ~ | ~ | ~ | ~ | ~ |

Date / /

Saathi

Date / /

Saathi

3. click on create issue if there are no duplicate.

| | |
|---|------------------------------------|
| Defect id: | 1234567890 |
| Build Name: | Alpha Build |
| Release Name: | beta |
| Req. No.: | 30.1 |
| Module Name: | Compose |
| Build id: | b04 |
| Status: | NEW/OPEN |
| Severity: | BLOCKER |
| Priority: | P1 |
| T-Name: | Gmail - compose - blank Bandeep |
| Test data: | UN-abc, PWD-xyz |
| Test url: | fshyadnebph |
| Defect summary: | Compose page blank |
| Detail description: | OS, version, browser |
| Observation/Actual result: | Blank Page |
| Expected Result: | compose page should displayed |
| Screen Shot: | [Screenshot] |
| Cc: | [Email] |
| <input type="button" value="Submit"/> <input type="button" value="Cancel"/> <input type="button" value="Reset"/> <input type="button" value="BUG HISTORY"/> | |

4. When click on Submit it will give a unique defect id

Steps to Reproduce defect

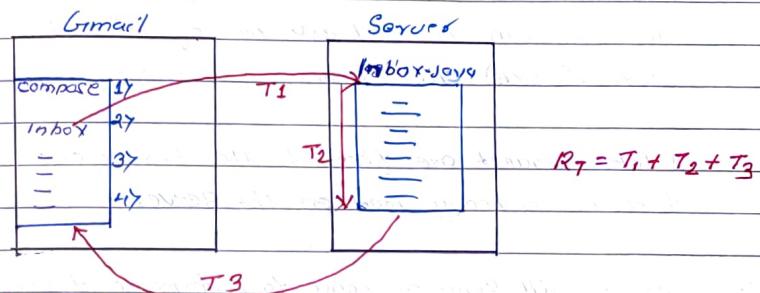
1. Open browser and enter url
2. Enter UN, PWD and click on login
3. Click on compose link

Page No. []

Spikeline / Performance testing / 8 Bottleneck

Testing the stability and response time of an app by applying load is called performance testing.

Response time: It is the time taken to receive the request (T_1) Time taken to execute the program (T_2) Time taken to send the response (T_3)

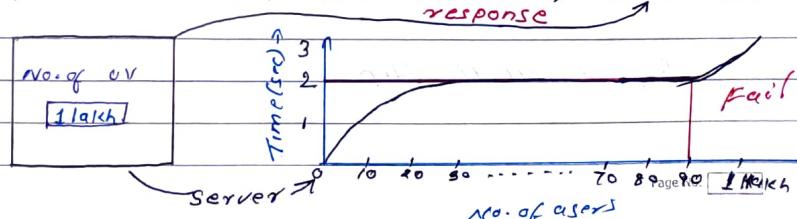
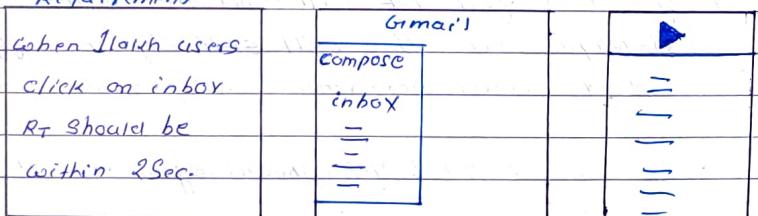


Load: It is nothing but designed no. of users.

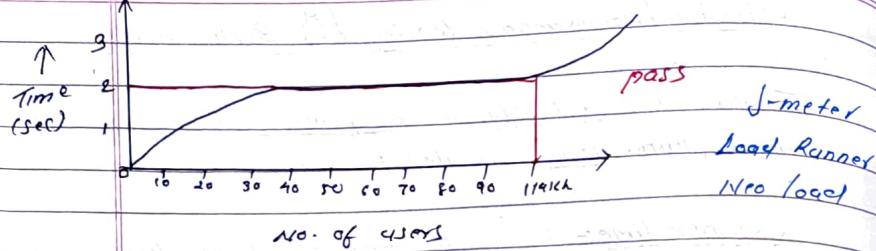
Stability: Ability to withstand designed no. of users or load.

- Q How to do performance testing using performance testing tool?

Ans:- requirement.



Date / /



1. Take any performance testing tool & write Scripts
2. Click on run it will ask no. of virtual users (enter 1 lakh) click run
3. 1 lakh request are fixed to the Server and now there is a heavy load on the Server
4. Server will send response to performance testing tool (J-meter)
5. Tool will analyse the response and gives the result in form of graph
6. T.E will analyse the graph and decide whether test is fail or pass
7. If it is fail T.E will send the defect to developer now they will entirely modify the code to improve performance i.e. performance tuning.
8. Get new build & T.E should again run the Script
9. Repeat all the steps

Date / /

Types of performance testing:-

1. Load testing
2. Stress testing
3. Volume testing
4. Soak testing

1. Load testing:- Testing the stability and response time of an app by applying load which is equal to and less than designed no. of users is called L.T.
2. Stress testing:- Testing the stability and response time of an app by applying load which is more than designed no. of users is called S.T.
3. Volume testing:- Testing the stability and response time of an app by transferring huge volume of data is called V.T.
4. Soak testing:- Testing the stability and response time of an app by applying load continuously for a particular period of time is called S.T.

Load Runner

| | No. of UV | 11akh |
|---|------------|-------|
| | No. of hrs | 24 |
| — | | 48 |
| — | | 72 |
| — | | — |

- Q. For what kind of app we should do performance Testing?

- Ans. 1. Any app which is used by multi users
2. Any app which generates lots of revenue

Date / /

Saathi

Date / /

Saathi

Q) When to do performance testing?

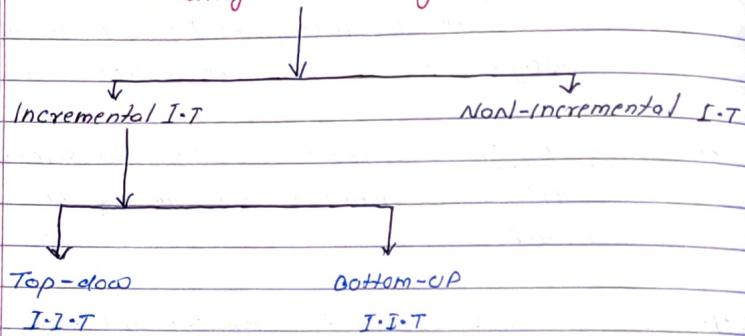
Ans.

In different projects we do performance testing in different ways.

- a) Some project they do performance testing when product become stable.
- b) In some project we do performance testing in SDLC itself.

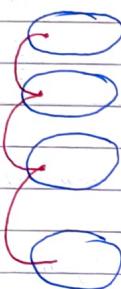
Types of Integration testing

Integration testing



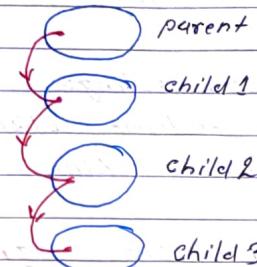
1. Incremental Integration testing:-

Incrementally adding the module and testing the dataflow b/w the module is called as incremental integration testing.



Top-down I.I.T. :-

Incrementally adding the module and testing dataflow b/w modules & make sure that module which you are adding should be child of previous module



Bottom-up I.I.T. :-

Incrementally adding the module and testing the dataflow & make sure that module you are adding should be parent of previous module

CEO

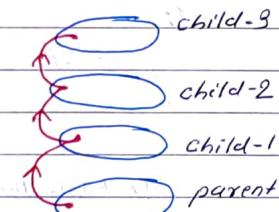
| | |
|-------------|------------------------|
| Add mgm. | <input type="text"/> |
| Edit mgm. | <input type="text"/> |
| Delete mgm. | <input type="text"/> |
| = | |
| Submit | <input type="button"/> |
| Cancel | <input type="button"/> |

Manager

| | |
|--------------|------------------------|
| Add engg. | <input type="text"/> |
| Edit engg. | <input type="text"/> |
| Delete engg. | <input type="text"/> |
| = | |
| Submit | <input type="button"/> |
| Cancel | <input type="button"/> |

Engineer

| | |
|--------|------------------------|
| — | Report |
| — | — |
| — | — |
| — | — |
| Submit | <input type="button"/> |
| Cancel | <input type="button"/> |



Note:- when it is difficult to identify which is parent module and which is child module then we go for
NON-INCREMENTAL INTEGRATION TESTING

Date / /

Saathi

Date / /

Saathi

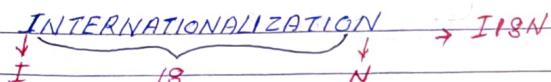
Globalization testing:

Developing the SW for different languages is called as Globalization and testing the SW which is developed for different languages is called as Globalization testing.

Types of Globalization testing:-

Internationalization testing - I18N

Localization testing - L10N



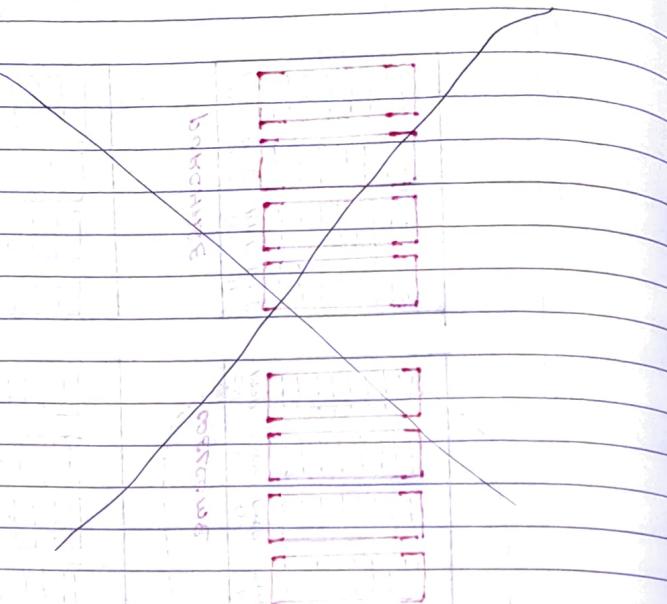
Step 1:-

- user opens the browser enters URL and click go,
- Request goes to Server and request object has 2 things
 - 1) Page name
 - 2) Language code
- In this example page name is welcome and the language code is English.

Step 2:-

- The Server is waiting for the request, as soon as the request object comes in it goes inside the request object & checks for page name
- In this example page name is welcome so it runs welcome.jsp



StepsStep 3:-

- The program starts running. This program has 2 sections
 - a) The first section runs and receives the request object goes inside the request object and checks for language code.
 - b) In this example language code is English so it connects to English property files.
- The connection b/w program & property files established

Step 4:-

- The remaining program copies the content from the property file and displays on the browser.

I18N testing:-

- Testing the UI for different languages is called as I18N testing.
- Here we check whether
 - a) Content is in right language or not.
 - b) Right content is displayed in right place or not.
 - c) Features are broken when the language changed.

How to do I18N testing? (Chinese)

1. Go to Chinese property file
2. Add prefix and suffix to the content
3. Save the property file
4. Open the application Select language (Chinese) corresponding page comes
5. Check for prefix, prefix is correct means the content is in right language.
6. Check for suffix, suffix is correct means content is displaying in right place

PSEUDO Translation:-

- For their understanding purpose in order to test the software they give prefix & suffix for different languages, this concept is called as pseudo translation.

What type of defect we find while doing I18N testing?

Def. 1- chances are there right language might not be displayed

eg:- if you take amazon app user selects a language as Chinese, if it's Shaoing Japanese or blank page it is called as defect.

Def. 2- chances are there right content might not be displayed in right place.

eg:- if u take amazon app, user selects a language as Japanese, entire page displayed in Japanese but contents like copyright displayed in English language.

Def. 3- Alignment Specification problem for languages

eg:- unidirectional & bidirectional

Def. 4- TOOL TIP defect

~~SO~~ what is Tool Tip

when user point cursor on image rectangular box will be displayed which explain about the image this is defect.

Tool Tip should be always

displayed in language which user has Selected if not it will be called as Tool Tip defect.

2. L10N testing (format testing)

Testing the S/W to check whether S/W is developed according to the country Standards or country culture is called as Localization testing.

1. Here we will check currency format is displayed

as per country Standards or not

eg:- INDIA:- RS.100 / ₹100 / INR100

US:- \$ 100 / USD 100

UK:- Pounds

2. Here we will check date format ie displayed as per the country Standards or not.

eg:- INDIA:- DD/MM/YYYY

US:- MM/DD/YYYY

Germany:- YYYY/MM/DD

3. Here we check PIN code format is displayed as per the country Standards or not.

eg:- INDIA:- 560051 (6-digit postal code)

US:- 81254 (5-digit ZIP code)

Germany:- 5/3 digit (ZIP code)

CANADA:- CA1108 (2 alphabet then Number)

4. Here we check the colour of the image displayed as per the country Standards or not

eg:- Colour of national flags

Smoke Testing:-

Alternate names-

1. Sanity testing
2. Skim testing
3. Dry Run testing
4. Build Verification testing
5. Health checkup of the S/W
6. confident testing

Def:- Testing the basic and critical features of an application before we do thorough testing or rigorous testing (F-T/I-T/S-T) is called

Smoke testing:

- It will take maximum $\frac{1}{2}$ day to Smoke testing.
- As soon as T.E gets build he is going to verify build that is why it is called build verification testing.

| | | | | |
|---|------------|----------|--------|--------|
| A | ✓ | (b01) | (b02) | (b03) |
| B | ✓ | 5 days | 4 days | 3 days |
| C | ✓ | | | |
| D | Blank page | 3-4 days | | |

1st day → A2nd day → B3rd day → C4th day → D (defect forced)5th day → F.T

→ According to this example customer will give the requirement for A, B, C, D module & developer will write code, do W.B.T and give the build to T.E

→ T.E will start doing F.T for A, B, C module for 1st 3 days, 4th day when you click on D module it's going to blank page

→ T.E will immediately communicate defect to the developer and developer will tell he need 3-4 days of time to fix the defect so 2nd of this 5th cycle will get postpone

→ Release will be delayed and customer will charge penalty for the company, In order to overcome this problem in every company as soon as we get the build / software you should do smoke testing.

Advantage of Smoke testing:-

1. T.E can find all the blocker defect in the early stage itself
2. Developer will get sufficient time to fix defect
3. Test cycle will not be postpone and release will not be delayed.

Note:-

1. In Smoke testing we will test only basic & critical features
2. We will take every basic & critical features & test for 1 or 2 important Scenario
3. Here we should do only positive testing.

Q) Which testing is Positive testing?

Ans) Smoke testing

4. In the beginning we will not be able to identify basic and critical features, we will learn it after getting very good product knowledge

How to do Smoke testing? (

1. Features to be tested as part of Smoke testing
Compose, inbox, login, Signup, Sent Items, All mails, Forward, Reply All, Search mail, Attachments, Logout, URL

Email-

9. Features not to be tested as Smoke testing
 Trash, Chat, Spam, important, calendar, draft, delete, Setting, Archive, contacts, outbox, Travel, Groups, Social, promotions, Help, Feedback, upload pic, Schedule.

For whatsapp

1. Features to be tested in Smoke testing
 chats, calls, upload pics, status, block

2. Features Not to be tested in Smoke testing
 unblock.

How to write Smoke Scenario?

4. 1. To check that when user clicks on inbox inbox page should be displayed
 2. To check that when user clicks on any of the mail, mail should be opened and the Content Should be displayed.

URL

To check that when user enters url then welcome Page Should be displayed

Assignment:-

Do Smoke testing & also write Smoke Scenario on below mentioned application

1. FB
2. whatsapp
3. Swifgyy
4. OLA
5. MS WORD / EXCEL / NOTEPAD

- How to do Smoke testing on Mobile?
 1. Feature to be tested on Smoke testing:
 power button, calls, camera, Speaker, Fingers print, mobile data, Sim card data, Battery, charging slot, Keypad, Answer button

2. Features not to be tested in Smoke testing:
 volume, message, Louder speakers, Alarm, memory card, Bluetooth, wifi, Hotspot, Earphone, Screen rotation, Flight mode, silent mode, Torch light, Brightness.

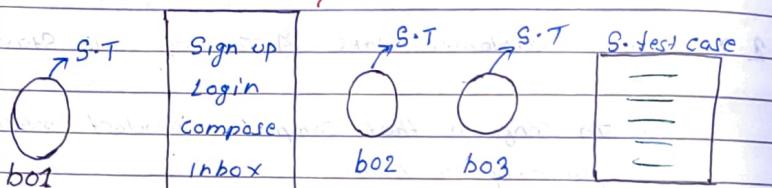
- whenever new build comes from D.T we always Starts with Smoke testing.
 → Adding, modifying or removing features or bug fixes
 → might have impact on basic or critical features
 → To find that in the begining itself we do S.T

1. Before customer does A.T we will start with S.T
 a) To confirm that complete product received properly or not
 b) To confirm that product is carefully installed and configures

4. Before developer give build to testing team, dev should do S.T, so that if too many bugs are there, they need not give that build to testing team
 3. One who install the product in production to confirm that product is installed properly he will do S.T

why we do Smoke Testing?

1. To check whether product is testable or not
In the begining only if we find too many bugs, it means product is not testable, so better stop testing and spend all the time in identifying some more scenario.
2. Do S.T. in the begining itself, if you find blocker or critical bugs send it to developer in the begining only so that dev. will have sufficient time to fix it
3. we do it to ensure that the product is installed properly
4. It is kind of health checkup of Software



Automation Script

| | |
|-----|-----|
| --- | --- |
| --- | --- |

→ we have to check whether we received broken build from dev. team

- Q which is first testing performed on software?
- a) Sanity testing
 - b) Functionality testing
 - c) W.B.T
 - d) Smoke testing

Q what is difference b/w Sanity testing and Smoke testing?

Ans: As per my knowledge there is no difference b/w Smoke and Sanity testing but I have gone through some of the websites & documents and observed there are some difference b/w Smoke and Sanity testing.

- gurug99.com
- Allinterviews.com
- S/W testing help.com
- quora/google (can also be)

Smoke testing

- Shallow - High level testing
- Wide - Try to cover all basic & critical features
- So Shallow and wide testing.
- eg - compose - 2 sec

→ Here we document Scenario & test cases

- Here we do automation
- It is done by both D.E & T.F

Sanity testing

- Deep & narrow testing

Narrow - we take one feature go deep inside & test it thoroughly.

eg - compose

| | |
|------|------------|
| To: | tve |
| Cc: | g |
| Sub: | -vc ip |
| | [Redacted] |
| S | C |

Date ___/___/___

- S.T is both +ve & -ve testing.
- We don't document Scenario & test cases.
- Here we don't do automation.
- ST is only done by T.E.

ADHOC Testing:-

- ST is also called out of box testing.
- b) -ve testing
- c) Monkey testing
- d) Gorilla testing
- Testing the application / software randomly without looking into the requirement is called adhoc testing.
- Why we do Adhoc testing?
 - a) Chances are there that customer & end-users might use the S/W randomly & they might find defects. In order to avoid that we should do Adhoc testing.
 - b) Developer will develop the S/W as per requirement & T.E. test the S/W as per requirement, chances of finding defect will be less if you test the S/W as per requirement so T.E.'s should come up with creative scenario if test the S/W by doing adhoc testing.
 - c) Since requirement are not followed by users we should do adhoc testing.
 - d) Since it is -ve testing Adhoc testing be done.

Date ___/___/___

- e) The intention of doing adhoc testing is to somehow break the product
- Write Scenario for adhoc testing for below app

| | | |
|------------------|---------|----------|
| www.Gmail.com | compose | TO [] |
| UN [] | inbox | Cc [] |
| PWD [] | Sub [] | Send [] |
| [LOGIN] [CANCEL] | | [Cancel] |

1. Login as user, click on Compose
2. Enter values for all the fields
3. Click on Send button and Logout
4. Click on browser back button & check whether Login page is displayed or app should ask to enter UN & Pwd

2. Log in page
1. Open the browser & Enter url.
2. Welcome page should be displayed.
3. Enter valid UN & Invalid Pwd & click login.
4. Proper error message should be displayed.

- Log in page
3. 1. Open the browser & Enter url
2. welcome page should be displayed
3. Enter Invalid UN & Valid Pwd & click login
4. Proper error message should be displayed.

4. Log in page
1. Open the browser & Enter url.
2. welcome page should be displayed.
3. Change the Pwd and enter with old Pwd.
4. Error message should be displayed.

Date / /

Saathi

Login page

Home page

Home page

Home page

Date / /

Saathi

5. 1. Login to same account in 2 different browser.

2. In one of the browser change password and in other browser do any action.

3. Check whether app is asking to enter changed pass or not.

6. 1. Login as user and click on Compose

2. In 'To' field enter wrong email Id and click on Send.

3. Check whether error message comes or not.

7. 1. Login as user and click on Compose.

2. In 'To' field write corong email Id.

3. In 'Cc' text field put correct email Id.

4. Check which displays wrong Email Id error.

8. 1. Login as user and click on inbox

2. Login as user in another browser and click on inbox

3. Delete some messages in 1st browser

4. Try to open the deleted message (of 1st browser) in 2nd browser

5. Check messages Shown deleted or not.

9. 1. Login to Email as user & click on Setting.

2. Click on change pass.

3. Enter new password Same as old pass.

4. Click on Submit button.

5. Check whether proper error msg displayed

→ When we do Adhoc testing?

→ When features and module is stable then we think about Adhoc testing

→ While doing Smoke testing we don't do Adhoc testing bcoz if we do Adhoc testing we will not get time to test basic features.

→ Whenever we are free we should spend time in Adhoc testing (i.e. after the SLO is tested as per the requirement).

→ Whenever we are doing F.T/I.T/S.T, if we get some creative scenario then we should do Adhoc testing.

→ If you get too many scenario then document it so, that to execute when we get free.

→ While doing F.T/I.T/S.T if you get some good Adhoc Scenario stop doing F.T/I.T/S.T and test for adhoc scenario but don't spend too much time. For it immediately switch back to F.T/I.T/S.T

→ Types of Adhoc testing

There are 3 types of Adhoc testing:-

1. BUDDY testing:-

D.E. Test E will sit with Dev, come up with creative scenario & test the SLO

2. PAIR testing:-

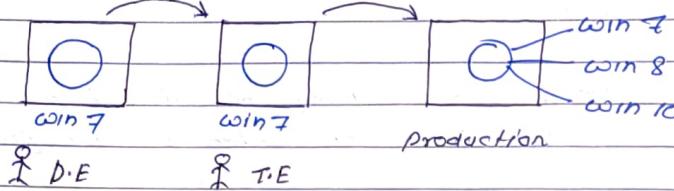
Here T.E will sit with T.E and come up with creative scenario & test the S/W.

3. MONKEY testing:-

Here T.E will test the S/W like a monkey without applying any logic.

COMPATIBILITY TESTING

→ Testing the functionality of an application in different hardcore and S/W environment is called as compatibility testing.



→ Chances are there developer would have developed the application in one platform, T.E would have tested in some platform and when product is launched user might use application in different platform.

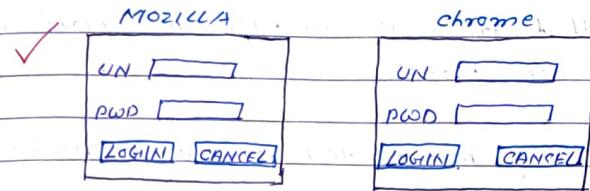
So, S/W which works on one platform may not work on other platform.

→ Due to which bad name spreads in market and no. of users who buy the product reduces.

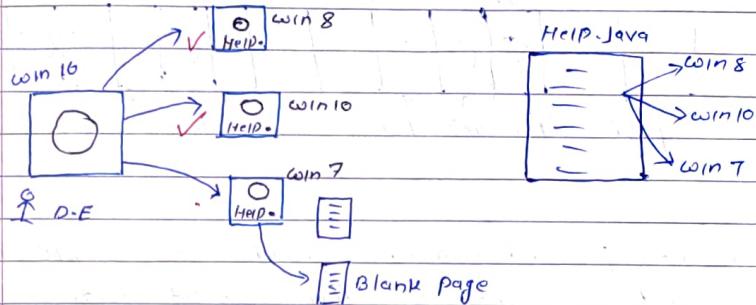
To avoid this we do compatibility testing.

Q why we do compatibility testing?

Ans → To check whether features are working consistently in all the platforms or not.



Q How to do compatibility testing?



Common code

```
if OS == Any
{
    display Help
    =
    work Help
    =
}
```

PLATFORM Specific code

```
if OS == win 10
{
    display Help
    -----
    work Help
    -----
}

if OS == win 8
{
    display Help
    -----
    work Help
    -----
}

if OS == win 7
{
    display Help
    -----
}
```

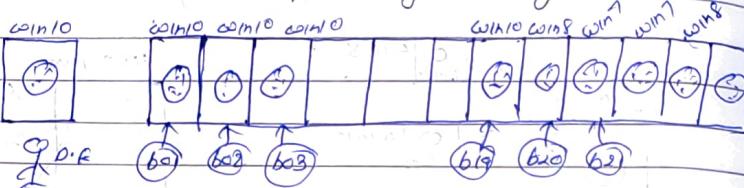
Date / /

Saathi

- Dev would have written Common code & they claim that it works on all platform
- Dev would have written platform Specific code & says that it works in respective platform or Corresponding platform
- So we may have to test in every platform & confirm that it really works

Q When we do compatibility testing?

Ans.



→ When product becomes functionally stable in base platform then we think about testing the application in other platform.

Q How do they choose platform?

Ans.

→ Based on ROI

→ Based on Company & Customer.

- a) Product based → Google analytics - tool, Graph
- b) Service based → customer

Q How to do compatibility testing?

- a) Stand alone
- b) Client Server
- c) web application

Date / /

Saathi

Stand alone application:-

- a) This type of app can be used only one user at a time.
- b) To use this s/w internet & Server is not required

Client Server application

- a) Here will be 2 s/w i.e. client s/w and Server s/w
- b) By using that client s/w we will interact with Server s/w
- c) To use this internet & Server is required.

Web application

- a) It is a client Server application where Server behaves like a client
- b) Any appn that can be accessed by opening browser and entering URL.

→ Stand alone app

i) Desktop based

a) Notepad

b) Paint

c) calculator

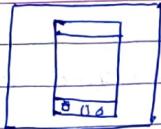
ii) Mobile based

a) calculator

b) Note

c) calendar

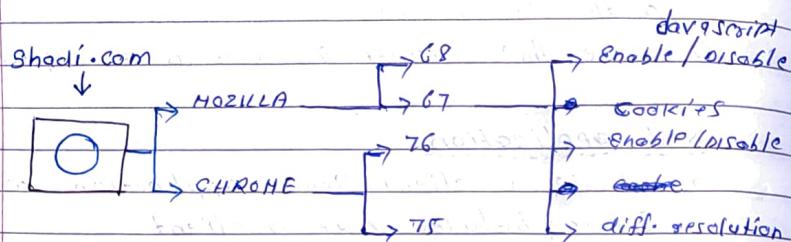
| ① | Windows 7 | Mobile devices | Android version | | | | ios | |
|---|------------|----------------|-----------------|---|---|---|-----|----|
| | | | 10 | 9 | 8 | 7 | 5s | 6s |
| | Windows 8 | SAMSUNG | □ | □ | □ | □ | □ | □ |
| | Windows 10 | OPPO | □ | □ | □ | □ | □ | □ |
| | | VIVO | □ | □ | □ | □ | □ | □ |
| | | MI | □ | □ | □ | □ | □ | □ |
| | | REALME | □ | □ | □ | □ | □ | □ |



② virtual mobile in System

- ③ crowd test platform - it is done by the crowd of people of the company.

client Server appn



** web application

- ④ It is testing the web application thoroughly.

- ⑤ Web application means you will have so many components, we may have to test each and every component thoroughly. i.e. F.T to

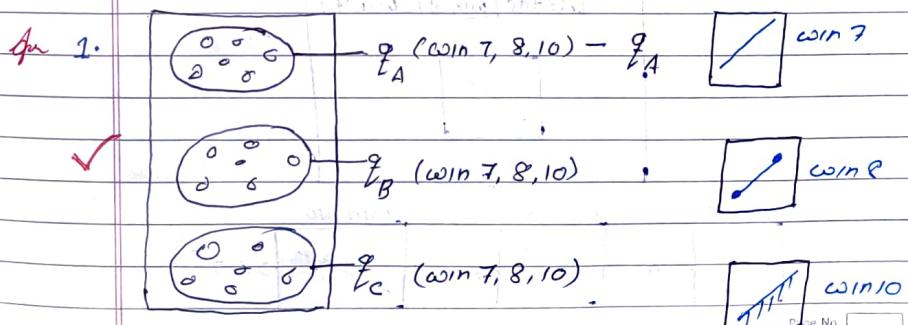
- ⑥ There will be a lot of dataflow b/w the features so we do I.T.

- ⑦ There will be a lot end-end business scenario so we do S.T.

- ⑧ End users might use the app in different Platform so we have to do C.I i.e. we may have to test the application in

different OS and, each OS test in different browser, in each browser test on different version, in each version test on different settings like -

- i) Enable / disable JavaScript
- ii) Enable / disable Cookies
- iii) Test in different resolution
- f) Variety of users might use the appn so it should be user friendly so we do usability testing
- g) Web appn means multiuser can use, so load on the web will be high so we have to do performance testing
- h) If devlopers has got Secured data we have to do web security testing.
- i) If app is developed for multiple language then we have to do I18N & L10N testing.
- j) To check whether it is user friendly for physically challenged people so we have to do accessibility testing.
- k) How will they allocate work while doing C.T?

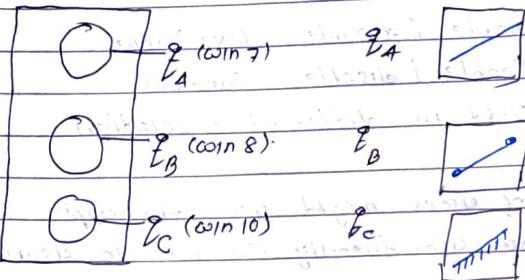


Date / /

Saathi

If you test your feature in different platform
you will get a chance to compare

2.



X

If different T.E. test in different platform
you will not get a chance to compare

Q How will they manage different platform for a
single T.F.

win 7 win 8 win 10

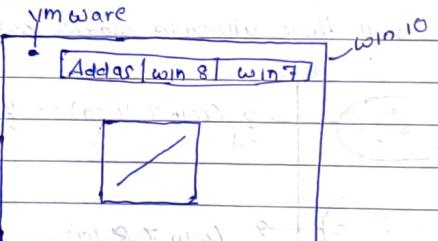
for

1. [] [] []

2. i) C:\drive - win 7 ii) D:\drive - win 8

iii) E:\drive - win 10

3. Vmware



win 7 win 8 win 10

→ F.T.

→ F.T.

→ F.T.

Date / /

Saathi

Doing FT / I-T / S-T in different platform is called
as C-T

HARDWARE COMPATIBILITY TESTING

a) Testing the functionality of an application in
different hardware environment is called hardware
compatibility testing

b) Here we test in different hardware like

1. Different Processors

a) different speed (GHz - 2.4, 1.0, 0)

b) different make (intel, asus)

c) different bit size (64 bits, 32 bits)

2. Different Mother board

3. Different VGA card

4. Different Monitor Resolution.

Q what kind of bugs you can expect while doing
compatibility testing.

An-

a) Alignment issue



b) Object overlapping

c) Scattered content.

d) change in look and feel of the application

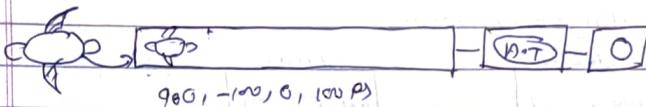
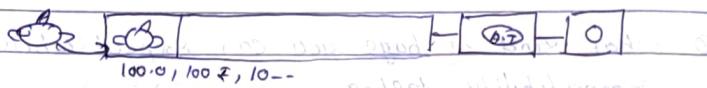
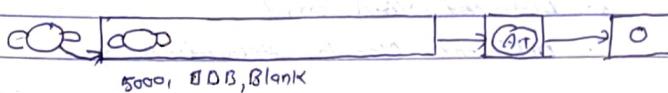
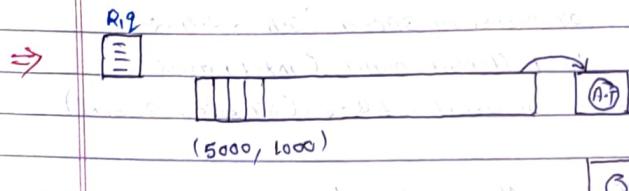
→ Image with certain format may not be
displayed in certain browser

Date / /

Saathi

- Q) Scroll bar issue
→ unable to scroll down or scroll back
→ Y →

| Amount Transfer | | 1) 500 8) 100.50 |
|-----------------|-----------------|---------------------|
| Amt Transfer | FAN | 2) 6000- 9) Hundred |
| Amt Balance | TAN | 3) -100 10) 1,000 |
| 0) | Amt | 4) #? 11) 90 |
| Transaction | | 5) 10-- 12) 100 ₹ |
| Logout | Transfer Cancel | 6) 0 13) BLANK |
| | | 7) 100R |



CBD_SRC 30. Amount Transfer

30.1 FAN-T-F

30.1.1 Should accept only 10 digit integer

30.1.2 Should accept no created by manager

30.2 TAN-T-F

30.2.1 Should accept only 10 digit integer

30.2.2 Should accept no created by manager

30.3 Amount T-F

30.3.1 Should accept only five integer from
(100-5000)

30.3.2 Should not accept more than balance

Date / /

Saathi

- Q) If you see the requirement and test the product what are the drawbacks.

- Ans. 1. There will be no consistency in test execution
2. Quality of testing over the period of time depends on memory power of test engineer.
3. Quality of testing varies from person to person
if engineer is smart he will try all the scenario otherwise he will try less scenarios.
4. To overcome this we document the scenario called as test case

- Q) What is Test Case?

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

It got different sections like

- | | |
|--------------------|------------------|
| 1) Step No | 6) Actual result |
| 2) Description | 7) Status |
| 3) Input | 7) Comment |
| 4) Expected result | |

- Q) When do we write test case?

- Ans. 1. When dev are building the product one req is given, testing team will be writing test cases.
2. When dev gives product for testing, testing team will test the product based on test cases.
3. When customer says modify or test the change requirement, dev will modify or change

Date ___/___/___

the features and also test team will modify or change the test cases.

Q. Why do we write Test Cases?

Ans.

1. To write T.C to have better test coverage
→ when req. comes in dev. are busy in building the product. Same time T.E are free so they identify all possible scenario and document it, when they build comes T.E can spend time in executing the scenario that T.E are covering will be same

2. To have consistency in test execution
→ It means if you have documented the scenario you can make sure that you are executing all scenario in all the test cycles or sprints or module

3. It depends on process rather than on person.

4. To avoid giving to every new engineer on the product or on the requirement

5. T.C is the only document which acts like proof for customer, dev team and manager that we have covered all possible scenario

6. T.C acts like a base document for writing the automation scripts, if you refer test case and write automation script you can ensure the same kind of coverage even in automation script.

Date ___/___/___

7. If you have documented the test cases, you don't have to remember the scenario

8. If you have documented test cases, execution happens in a very organized way.

9. If you have written test cases time taken to execute will be very less.

Test Case Format

1. Test Case Name
2. Requirement No.
3. Test data
4. Pre-Condition
5. Test Case type

Comments

| S.L.NO | Description(Action) | INPUT | EXPECTED RESULT | ACTUAL RESULT | STATUS |
|--------|--|---------|--|--------------------------------|-------------|
| 1. | Enter character into Amount text field & click on transfer | Hundred | Appropriate error msg. Should be displayed | Error msg. Should be displayed | In Progress |
| 2. | Enter -ve integer into Amount text field & click on transfer | -100 | Appropriate error msg. Should be displayed | Error msg. Should be displayed | In Progress |

Author: Sandeep

Approved by:

Reviewed by:

Approval date:

Date / /

Saathi

Date / /

Saathi

→ Test Case design Techniques:

1. Error guessing
2. Equivalence partitioning
3. Boundary Value Analysis (BVA)

→ 1. Error Guessing:

Amt Amount Text field

1. Character 5. 1,00
2. Decimal 6. Blank Space
3. Special char. 7. Alphanumeric
4. zero 8. \$ 100

Here we guess all possible errors based on that we drive the Scenario

We guess error based on:-

- a) Based on reg.
- b) Based on Experience
- c) Based on intuition

→ 2. Equivalence Partitioning:

- i) Pressman
- ii) Practice

i) Pressman:

Rule 1.

If the I/O is in Range of values then design the test cases for 1 valid and 2 invalid values

eg:-

Amt → 100 - 5000

Valid → 500

Invalid → 90

6000

eg:-

INSURANCE

Age → 5-55

[Submit] [Cancel]

Valid → 50

Invalid → 4

60

Rule 2

If the input is in Set of values, then design the test case for 1 valid and 2 invalid values.

eg:-

online shopping.com

Product

[Submit] [Cancel]

Printer - 10

Scanner - 20

Webcam - 30

Valid → 10

Invalid → 11

21

eg:-

$$B = \{ -6, 4, 2, -8 \}$$

Valid → 4

Invalid → 5

3

Date _____

Rule 3:

If the I/P is boolean, then design the test case for both true and false value.

eg:-

| | | |
|--------|--|--------|
| male | <input type="checkbox"/> / <input checked="" type="checkbox"/> | → True |
| Female | <input type="checkbox"/> / <input checked="" type="checkbox"/> | |
| SI | <input checked="" type="checkbox"/> | |

eg:-

| | | |
|-------------------|----------------------------|----------------|
| Fixed Interest | <input type="checkbox"/> ✓ | Valid Interest |
| Flexible Interest | <input type="checkbox"/> ✓ | |

→ 1. valid

ii) Equivalence partition (practice)

Input range [Amount] → 100 - 5000 or Invalid = All other values & more than 5000 → Valid → 90% of the range
Invalid → 6000

Partition

| Range | Test Data | Result |
|-------|-----------|--------|
| 90 | 90 | Valid |
| 1000 | 1000 | Valid |
| 2000 | 2000 | Valid |
| 3000 | 3000 | Valid |
| 4000 | 4000 | Valid |
| 5000 | 5000 | Valid |
| 6000 | 6000 | Valid |

Divide the range into equivalent parts and test for all the values make sure that you are testing the app for atleast 2 invalid values.

Date _____

when there is a deviation in requirement we go for practice method

eg:-

Reg: Amt T.F → 100 - 5000

Deviation:- 100 - 2000 → 1% & transfer

2000 - 5000 → 2% & transfer

other than this → Error

Dev-

if (Amount 100 - 2000)
{Deduct 1%
Transfer
}if (Amount 2000 - 5000)
{Deduct 2%
Transfer
}

if (Amount

else
{throw error msg.
}

3. BVA (Boundary value Analysis):

If the input range of the value between A to B then design the test cases for 'A', 'A+1', 'A-1' and 'B', 'B+1', 'B-1'

Amt: 100 5000

| | |
|-------|--------|
| → 101 | → 5001 |
| → 100 | → 5000 |
| → 90 | → 4999 |

Functional Test Case

Date ___/___/___

Saathi

- Q How to write F.T.C?

| | | |
|-----|----------|----------|
| A-T | FAN | [] |
| | TAN | [] |
| | Amt | [] |
| | Transfer | [cancel] |

Ans) option for test case documentation

| FAN | TAN | Amt | |
|-----|-----|-----|---|
| [] | [] | [] | X |

| | |
|-----|--|
| FAN | |
| | |
| | |
| TAN | |
| | |
| | |
| Amt | |
| | |

- L ① Before we actually start writing Test cases we should come up with option and select best option.

- L ② Always start writing test cases with Navigation steps-

- L ③ Always use Should be or must be in the Expected result table don't use maybe, can be, might be etc..

Date ___/___/___

Saathi

- L ④ Always write generic test cases, don't hardcode the test case.

- L ⑤ Elaborate only those steps in which you have the focus, don't elaborate all the steps.

- L ⑥ whenever we are writing the test cases, we should imagine the application
from 7 --- *

- Q Test case format differ from company to company.

continue further -- *

- Q Approach to write F.T.C

- Ans - 1. Go to body of test case
2. Start with navigation steps
3. Take 1st field
 a) Start with valid data
 b) write E.G. Scenario
 c) write E.P Scenario
 d) write B.V.A Scenario
4. Take 2nd field
 a) Start with valid data
 b) write E.G. Scenario
 c) write E.P Scenario
 d) write B.V.A Scenario

eg:- F.A.N - []

T.A.N - []

Amt - []

Date / /

Saathi

4. Following are the steps to test F.A.N T.F

Saathi

Date / /

| SL.NO | DESCRIPTION | INPUT | EXPECTED RESULT | ACTUAL RESULT | STATUS | comment |
|-------|--|------------------------|--|--|---------------|---------|
| 4.1 | To test Enter the valid 10-digit A/c no. into F.A.N T.F and click on Transfer button | valid 10-digit A/c No. | Error msg' Should not be displayed for F.A.N T.F field | Error msg' Should not be displayed for F.A.N T.F field | Inappropriate | |
| 4.2 | Enter character into F.A.N T.F and click on transfer button | ABCDE | Error msg' Should be displayed | Error msg' Should be displayed | Inappropriate | |
| 4.3 | Enter Special char. # ? into F.A.N T.F and click on transfer button | # ? | Error msg' Should be displayed | Error msg' Should be displayed | Inappropriate | |
| 4.4 | ----- | ----- | ----- | ----- | ----- | |
| 4.5 | ----- | ----- | ----- | ----- | ----- | |
| 4.6 | ----- | ----- | ----- | ----- | ----- | |
| 4.7 | ----- | ----- | ----- | ----- | ----- | |
| 4.8 | ----- | ----- | ----- | ----- | ----- | |
| 4.9 | ----- | ----- | ----- | ----- | ----- | |
| 4.10 | ----- | ----- | ----- | ----- | ----- | |
| 4.11 | Enter 10-digit Invalid A/c no. into F.A.N T.F and click on Transfer button | 22222 | Error msg' Should be displayed | Error msg' Should be displayed | Inappropriate | |
| 4.12 | Enter 9-digit Invalid A/c no. into F.A.N T.F and click on Transfer button | 21212 | Error msg' Should be displayed | Error msg' Should be displayed | Inappropriate | |

Date / /

Saathi

5. Following Steps is to test TAN T.F make sure that F.A.N T.F is having valid data.

| SL.NO | Description | Input | Expected Result | Act. Step Res. | SL.NO |
|-------|---|----------------------|--|----------------------|-------|
| 5.1 | Enter valid 10-digit A/c no. into T.A.N T.F field & click on Transfer button. | Valid | Error msg. Should not be displayed | Step 743 Res: +ve | 6.1 |
| 5.2 | Enter character into T.A.N T.F & click on transfer button | ABCDE | Appropriate error msg. Should be displayed | | 6.2 |
| 5.3 | Enter Special characters into T.A.N T.F and click on transfer button | # % | Appropriate error msg. Should be displayed | | 6.3 |
| 5.4 | - - - - - - - - - - | - - - - - - - - - - | - - - - - - - - - - | | 6.4 |
| 5.11 | Enter 10 digit Invalid A/c no. to T.A.N T.F and click on transfer button | 33333 33333 | Appropriate Error msg. Should be displayed | | 6.5 |
| 5.12 | Enter 9-digit Invalid A/c no. to T.A.N T.F and click on transfer button | 21312 1312 | Appropriate error msg. Should be displayed | | 6.15 |
| -- | -- - - - - - - - - - | -- - - - - - - - - - | -- - - - - - - - - - | | |

Date / /

Saathi

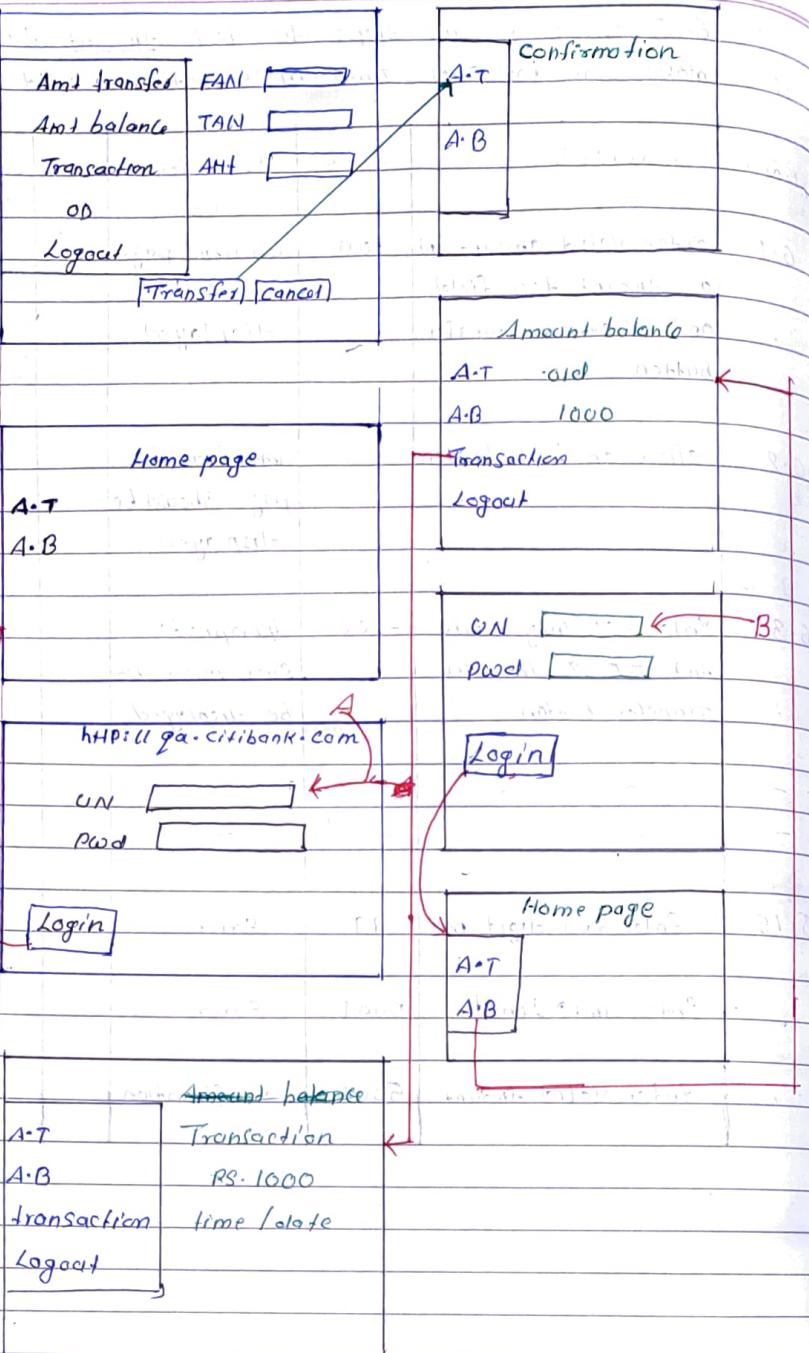
6. Following are the Steps to test Amount text field. make sure that FAN & T.A.N T.F Should have valid data.

| SL.NO | Description | Input | Expected Result | Act. Step Res. | comm |
|-------|---|--------------|---|----------------|------|
| 6.1 | Enter valid amount int o Amount text field and click on Transfer button | 100 | confirm page should be displayed | | |
| 6.2 | click on back button | | Amt transfer page should be displayed | | |
| 6.3 | Enter -ve integer into Amt T.F & click on Transfer button | -100 | Appropriate error msg. shd be displayed | | |
| 6.15 | Enter two digit no 99 Enter Amt > Limit 5001 Enter Valid Amount 5000 | 99 5001 5000 | Error Error Confirmation | | |

Integration Test Cases

Date / /

(Saathi)



Date / /

(Saathi)

| SL.NO | Description | Expected Result | Status | Comment |
|-------|---|---|--------|---------|
| 1. | Open the browser and test url (- url -) | Welcome page Should be displayed | | |
| 2. | Login as user A with valid UN & Pwd (UN - XYZ123 Pwd - ABC123) | Homepage should be displayed | | |
| 3. | Click on Amt link | Following should be displayed F-A-N, T-A-N etc | | |
| 4. | Enter valid A's Ac No. into Amount transfer F-A-N T.F B's Ac No. in T.A.N | Confirmation page Should be displayed | | |
| 5. | Click on Amount balance link | Amt. balance should display or Balance amt. Should be equal to old bal. - transfer amount | | |
| 6. | Click on transaction link | ① Transaction page should be displayed ② Transferred amt date, time, B's account should be displayed | | |

Date / /

Saathi

7. Click on Logout link welcome page
Should be displayed
8. Login as user B with valid UN & Pwd Homepage should be displayed
9. Click on amount balance
 ① amount balance Pg
Should be displayed
 ② Balance amt should be equal to old balance plus amount transferred.
10. Click on transaction link
 ① Transaction Pg
Should displayed
 ② Transaction amt, date, time & A/c No. should displayed

Date / /

Saathi

Test data

It is the data which is written before executing Step no. 1.

Eg:- T.E Should have UN & Pwd & A/c No. of user 'A' and 'B'.

Precondition

It is the condition which is written before executing Step no. 1.

Eg:- T.E Should have check the balance of 'A' & 'B'

① How to fill Test Case Header

Ans. ① Test Case Name :- CBO-AT- All Integration Scenarios
Q39A
(project name + module name - Scenario)

② Req. No. :- 30.1 Amount transfer

③ Test data:-

a) you can create - Gmail.com - Signup, 'A' & 'B' bal. check
b) depending on 'someone':

→ DE → S/W

→ UN - XYZ123, Pwd :- abc123

④ Authorised....

Licence Key []

Licence management team

} depend on
Licence ring.
from

⑤ FLIPKart :- during Credit amount

↳

→ T.E - buy - C.C.NO.

→ pay pal

} depend on
customer

Q How to write integration test case?
Ans.
 1) Take one feature, identify all possible Scenarios.
 2) prioritise the identified Scenarios, and document the Scenario

3) Go to the body of test case

4) Start with navigation Steps

5) Cover Scenario 1, Scenario 2,

L 8) One test Case document contains one Scenario or 'n' no. of Scenario.

To write one Scenario, it takes one step or n. no. of steps

Date ___ / ___ / ___

Q) How to manage test data?

- 1) few of the project will create test data manually
- 2) In many of the projects they will Auto Script, as soon as the build comes they run the Scripts, The Script creates the test data & Send mail to T-E

4) Precondition

It is set of action/setting which is done before Step no. ①

Eg:- T-E should check bal. of user A & B

| | | File | Edit | Tools |
|--|--|------|------|-------|
| <input checked="" type="checkbox"/> ABCD | | | | ABCD |
| <input checked="" type="checkbox"/> XYZ | | | | ijk |
| <input checked="" type="checkbox"/> ijk | | | | |

[Next]

ijk : precondition:- while installing software you should
↓
Select ijk check box

1) open the app

2) click on tool menu

3) click on ijk

5) Severity:

Writing Severity in test Cases helps you to prioritize your test execution.

6) Test Case type:

Depending on what we are writing we have to fill this

Date ___ / ___ / ___

7) Brief Description:

Following Test Cases cover all Information Scenario for amount transfer feature

8) FOOTER:

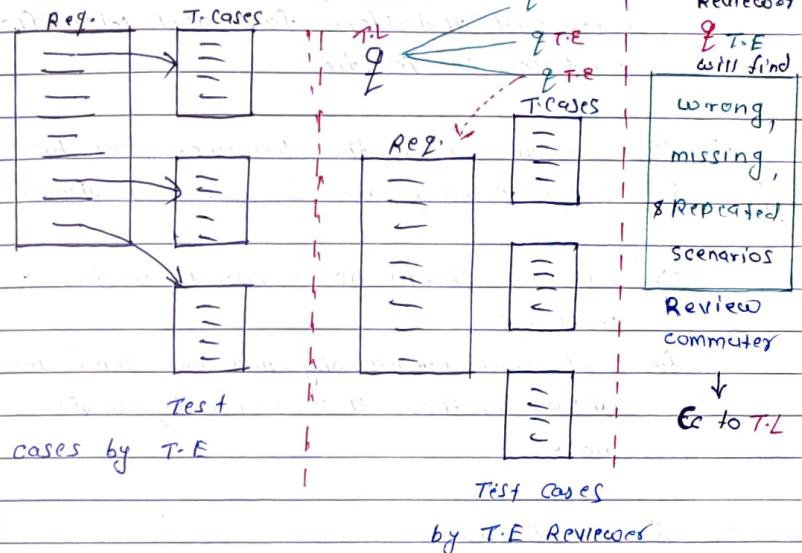
Author:

Approved by:

Reviewed by:

Approval date:

→ Test case Review process:-



Q) On what basis they assign Test Cases to Review

Ans:-



T.T.C given to Reviewer to do the T.E



who understand the project well or



Smart enough to find more mistakes



in less time.

Date / /

Saathi

Date / /

Saathi

- 1) we assign to that fellow working on similar/ related module on Project.
- 2) we assign to that fellow who worked on same module in previous project.
- 3) we assign to that fellow working in same project Since from beginning he knows every corner about the project.
- 4) we assign to that fellow who is responsible, who will understand fast and identifies more mistakes in less time.

Q How do you ensure that Reviewer does his Job?

Ans. 1) Assign primary and Secondary Reviewer.

- 2) Test lead should also randomly review & find mistakes.
- 3) Test lead intentionally introduce some mistakes & check whether it is found by reviewer or not.

Q Reviewer Ethics

- Ans.
- 1) Always Review the Content not Author.
 - 2) while Reviewing Spend time in identifying mistakes not in finding Soln for it.
 - 3) Even after review if there are any mistakes, both author and reviewers are responsible.
- Q why we review Test case? or
How will you review test case? or

what will you do while reviewing Test Cases?

1) I will look into the header of the Test Case, and first understand the requirement for which test case is written, then goto body of the test case and try to find

a) Missing Scenario

b) wrong Scenario

c) Repeated Scenario

2) I will check whether it is organised or not, so that when executed take less time.

3) I will check whether it is simple to understand or not, so that when given to new T.E, he should be able to execute it without asking any question

4) I will look into the header of the test case and try to find

a) All the attributes are covered or not.

b) Check whether content in all the attributes is correct or not

c) Check whether Test Case format is according to standard defined in the project.

Q Test Case Review document format?

| Sl. No | Test-Case name | Step No pre-cond. | Reviewer comment | Severity | Author comment |
|--------|--------------------|----------------------|---|----------|----------------|
| 1. | CBO_AT_AllScenario | 55- Pre- cond. | Transfer more than the bal. Scenario's missing. | Critical | P |

Date / /

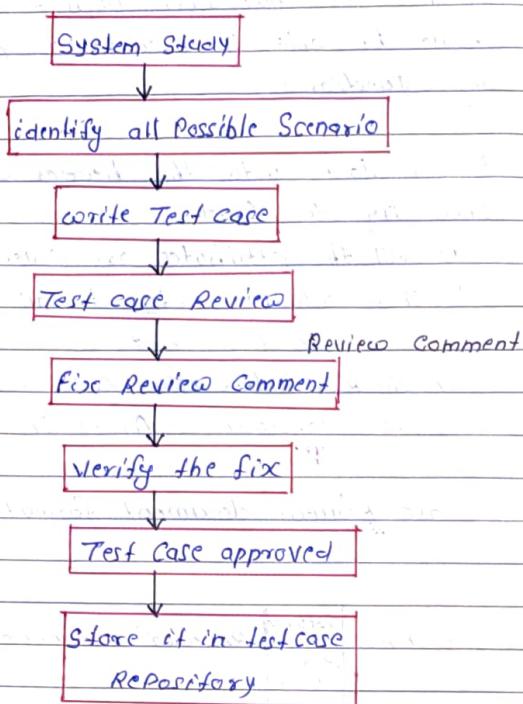
Saathi

Date / /

Saathi

| | Pre cond: | Pre-condn is missing | Minor |
|------------------------------|-----------|---|----------|
| 2. CBO-Ins-Age field testing | 7. | Insufficient ^{coverage} test coverage apply test case design techniques and design some more scenario 10.50 10 0 0 10 yrs. = | critical |

9. Procedure to write Test Case?



1. System Study:

Read the requirement and understand it - if you have any queries then you can interact with

- a) Developer
- b) Business Analyst
- c) Customer

and get it clarified.

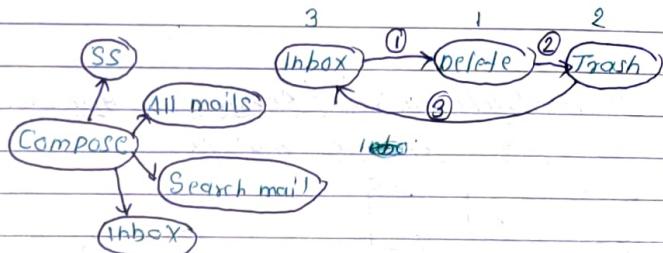
2. Identify all Scenario:

Brain Storming Session

- a) present how feature is allocated
- b) Improves product knowledge
- c) present all identified Scenario
 - i) MISSING
 - ii) wrong
 - iii) Repeated
- d) Measure efficiency of Brain Storming Session

3. Write test case:

- a) Document the identified Scenarios
- b) Group the Identified Scenarios
- c) Prioritize the Scenarios
- d) Apply Test Case design techniques
- e) use Standard test case format.

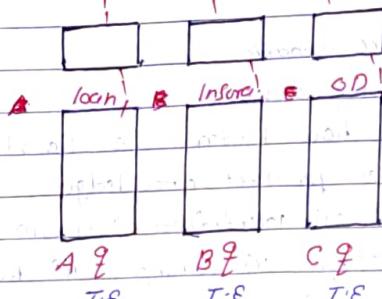
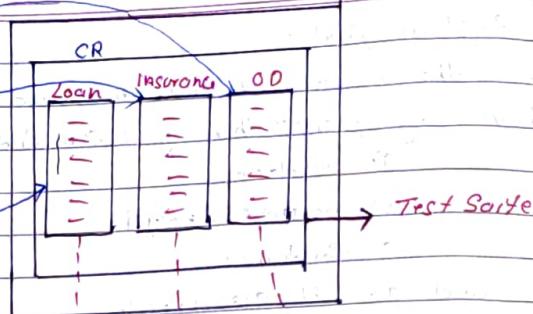
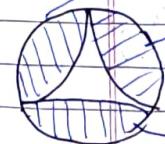


8. Store it in test case Repository

It is a Central Place where we store all the test cases

Test Suites: It is the collection of related test cases.

Date / /



Date / /

| Req. | T-C |
|-------|-------|
| ----- | ----- |
| ----- | ----- |
| ----- | ----- |
| ----- | ----- |
| ----- | ----- |
| ----- | ----- |

It is a document which we prepare to make sure that every requirement has got at least one test case.

Q. How do we do RTM?

| MODULE NAME | HIGH LEVEL REQUIREMENT | Detailed Requirement | TEST CASE NAME | AUTOMATION SCRIPTS |
|-----------------|------------------------|---|-----------------|--------------------|
| LOANS | 1.1 PERSONAL LOAN | 1.1.1 ----- 1.1.2 ----- | CBO-PL-APPROVAL | CBO-PL-APPROVAL |
| | 1.2 HOME LOAN | 1.2.1 ----- 1.2.2 ----- | CBO-HL-APPROVAL | |
| AMOUNT TRANSFER | 2.1 F-A-N T-F | 2.1.1 Should accept only 10 digit integers 2.1.2 Should accept only no. created by Manager | | |
| | 2.2 T-A-N T-F | 2.2.1 ----- 2.2.2 ----- | | |
| | 2.3 Amount T-F | 2.3.1 Should accept only 10 integers from 100 - 5000 2.3.2 Should accept <bal | | |

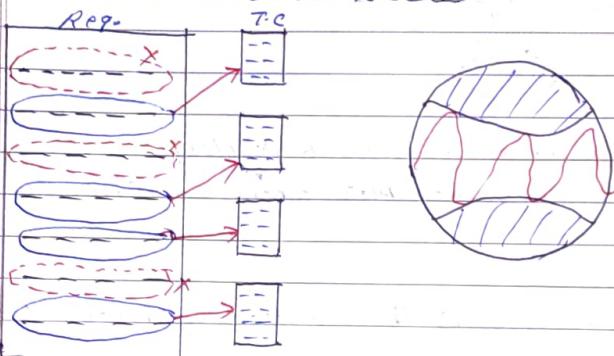
Q. Where do you write Test Case?

- 1) Excel / Word files
- 2) Test Management Tool (ALM, Test link)

Q. Where do you store Test Case?

- 1) In a Shared folder
- 2) In a Test Management Tool

→ Traceability matrix / Requirement traceability matrix or Cross Reference matrix



Date / /

ADVANTAGE

1. It ensures that every req has got one test case which indirectly assures you that you are testing every feature atleast once.
2. It gives traceability from high level req. till Automation script.

DISADVANTAGE

1. It will not ensure that you have got 100%.

| Traceability matrix | Test Case Review |
|--|---|
| 1. Here we check every req. has got atleast 1 T.C | 1. Here we check whether all possible Scenario are covered or not |
| 2. Here we don't check whether every req. has got all possible Scenario. | 2. Here we don't check whether every req. has got atleast 1 Test Case |

Q If there is no req. how do we write Traceability matrix?

Ans. Explore the application, understand app, identify document it, write Test cases, once after writing question is what is the proof that you have written test cases for all the features.

To ensure that we write the Traceability matrix.

There we list all the feature that are there in the product & map it to the T. cases.

Date / /

| | |
|-----------------|--|
| LOG IN | |
| Forgot Password | |
| Reset password | |
| Compose | |
| Inbox | |

Types of Traceability matrix

1. Forward traceability matrix:

If you are mapping from root document to derived document it's called as F.T.M

Eg:- Requirement → Test Cases → Automation Scripts

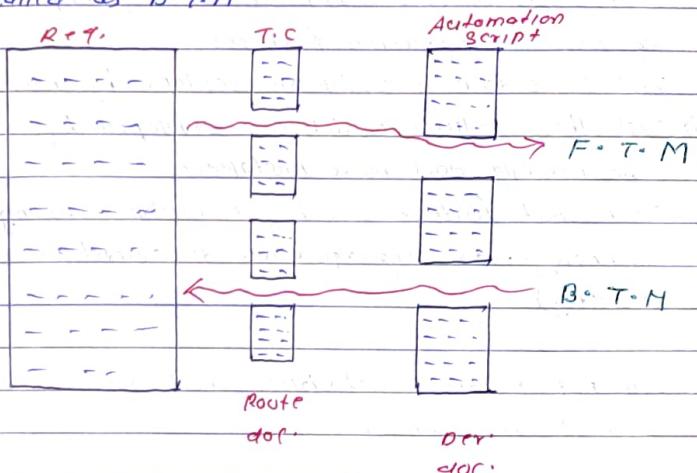
2. Backward traceability matrix:

If you are mapping from Derived document to root document it's called as B.T.M

Eg:- Automation Scripts → Test Cases → Requirement

3. Bidirectional traceability matrix:

If you do both 'F.T.M' and 'B.T.M' then it is called as B.T.M



only answer Def. and Advantage
No. disadvantage & types

Q Last day when you are about to release the product to the customer, if there is a critical blocker bug, what will you do? Will you release the product to the customer?

Ans. As a T.E i am not a decision maker, So i will list all the bugs which are pending and the impact of the defect on the business, and also i suggest whether to release or not and send it to P.M.

It is the manager who should trace a calculated decision and decide.

Q When time is very less what would be your approach?

Ans. 1. Take only very focus approach to test the product.

a) Try to test all the business critical features first.

2. Test those features which gives more coverage in less time

3. Come up with all the trick's to find more bugs in less time

a) Try to test all the new or modified features first.

b) Test those features which are having impact bcoz of new or modified features.

c) Test those features which are developed by new dev.

d) Test those features which are developed by developer who normally does more mistakes.

e) Test that feature which got developed using new technology or language (new for developer not for industry)

4. I will not do those activities which are really not important or critical

5. I will not do more negative testing.

6. I will not do adhoc testing.

7. I will avoid using usability testing

8. I will not test it in all platforms but only in important platform.

9. I will stretch myself for more hours & try to complete the testing.

Q How do you convince your customer or management or developer that you have tested everything?

Ans. 1. My test coverage is good bcoz my testcases are good. My test cases are good bcoz I followed a strict procedure (different technique to write testcase).

I followed strict procedure means

i) I did a thorough System Study bcoz of which I had a deeper knowledge of the product, bcoz of that I was able to find out more no. of Scenario.

ii) I identified all possible scenarios, that then I did a brainstorming session, bcoz of which I was able to find more no. of scenarios.

iii) While writing test case I applied testcase design techniques bcoz of that there was an improvement in the coverage.

iv) I got all my testcases reviewed at that stage we got some more missing scenario and that result improvement in the coverage.

v) While executing test cases I identified some new scenario, I added them back into the testcase bcoz

Date ___ / ___ / ___

Saathi

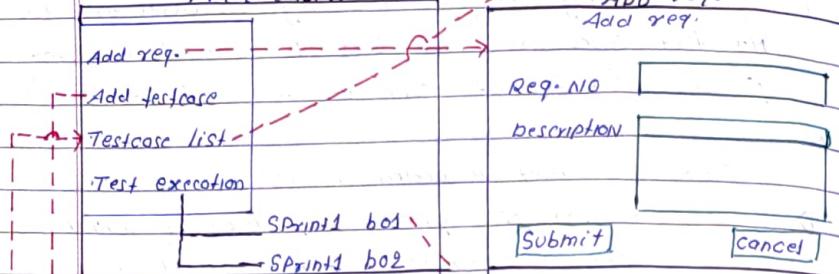
of this there is improvement in test coverage.

6) ALSO I perform adhoc testing which resulted in improvement in the coverage

7) I created traceability matrix and ensured that each req. has got atleast one test case was written bcoz of that there was an improvement in test coverage.

Test Management Tools (ALM)

T.M.T (4LM)



ADD Test case

Test case name CBO-AT-AIIScenario □
CBO - INS - APPROVED

Req NO

| |
|--------|
| 30 AT |
| 31 INS |

Test data
Pre-condition

| Step No. | Description | ER | AR | Status | Comment |
|----------|-------------|----|----|--------|---------|
| 1 | | | | | |
| 2 | | | | | |
| 3 | | | | | |
| 100 | | | | | |

Date ___ / ___ / ___

Test_case_list

| Req. NO. | Test-Case name | Edit | Del |
|----------|--------------------|------|-----|
| 30 AT | CBO-AT-AllScenario | Edit | Del |
| 31 INS | CBO-INS- Approved | Edit | Del |

Create copy / Name- Sprint 1, but

Suhm

Here you can
execute and
edit or delete
test case

→ It will Create
Sprint, bug in
Test execution

6

→ Excel Sheet

| Test Execution Sprints 6-01 | | | | | |
|-----------------------------|------------------|--------|------------|------------|---------|
| Req No | Test Case name | Status | Pass Count | Fail Count | Comment |
| 30 AT | CBO AT ALIASCEN | Pass | 1 | 0 | |
| 31 INS | CBO-INS-APPROVED | fail | 0 | 1 | |
| 32 | | | | | |
| 33 | | | | | |
| 34 | | | | | |
| 35 | | | | | |
| 2000 Test Case total | | | 2000 | 1 | |

Here you can execute t-cases but can't edit or delete.

| Test Case name | CBO-AT-All Scenarios |
|---------------------------------------|---------------------------------------|
| Requirement NO | |
| Step No. | Description |
| | E.R |
| | Status pass/fail |
| | Comment N/A |
| <input type="button" value="Submit"/> | <input type="button" value="Cancel"/> |

Date / /

Saathi

Test Management tool Contains:

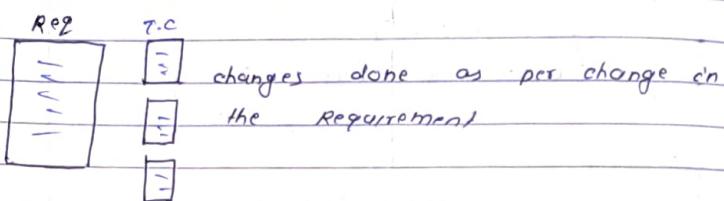
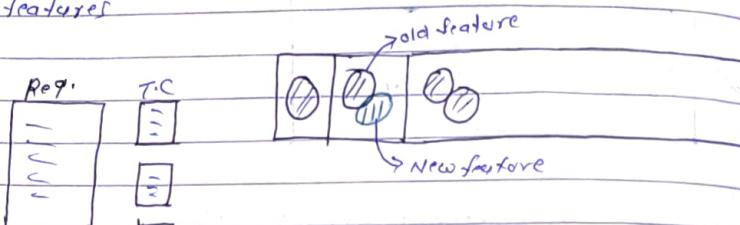
- Requirement management feature - Add, Edit, delete
- TestCase Management feature - Add, Edit, delete, execute
- Defect tracking features

Regression testing

Def 1:- Testing the unchanged feature to make sure that it is not broken/affected bcoz of changes is called as Regression testing.

Changes can be addition, modification, removal or bug fixing.

Def 2:- Re-execution of same test cases in different cycles/sprint/release to make sure that changes (addition, modification, removal or fixing bug) is not affecting the old features.



Types of R.T

- UNIT R.T
- REGIONAL R.T
- FULL R.T

1. UNIT R.T:- Testing only the changes made or bug which is fixed is called as UNIT R.T

Date / /

Saathi

eg:-

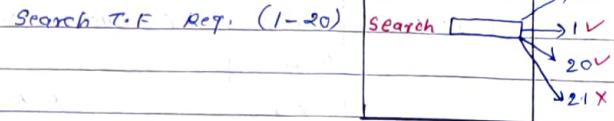
| Add user | |
|---------------|----------------------|
| Add user | <input type="text"/> |
| | <input type="text"/> |
| | <input type="text"/> |
| Submit | Cancel |

After filling all details click on Submit which gives the confirmation msg. This msg is full of grammar mistake.

After dev. corrected the conf. msg. bug are test only the conf. msg. grammatical mistake nothing else. This is called unit R.T.

eg:-

| | |
|-------------------|-------------------|
| confirmation msg. | → grammar mistake |
|-------------------|-------------------|



eg:-

| | |
|-----|---------|
| b01 | Notepad |
|-----|---------|

| | | |
|------|------|------|
| File | Edit | TOOL |
|------|------|------|

Help is missing

| | |
|-----|---------|
| b02 | Notepad |
|-----|---------|

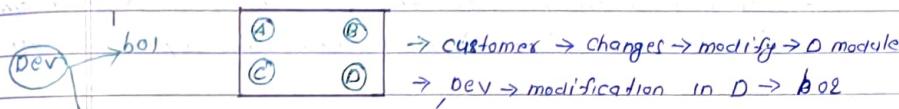
| | | | |
|------|------|------|------|
| File | Edit | Tool | Help |
|------|------|------|------|

Help appears after bug fixed

2. REGIONAL R.T:-

Testing the impacted area (affected area) due to some change is called as Regional R.T

Dev



b02

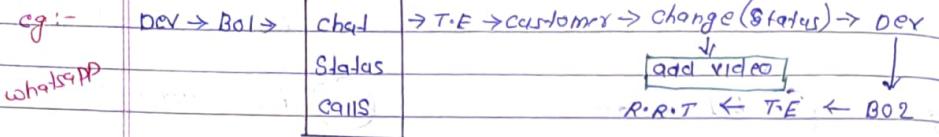
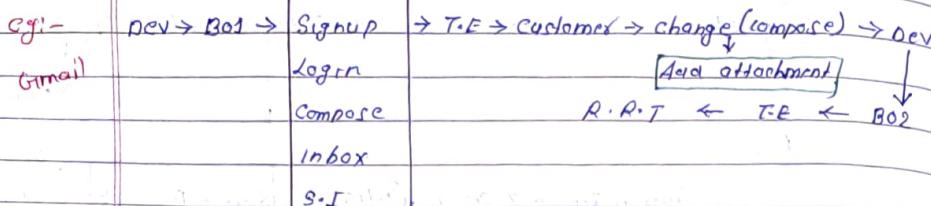
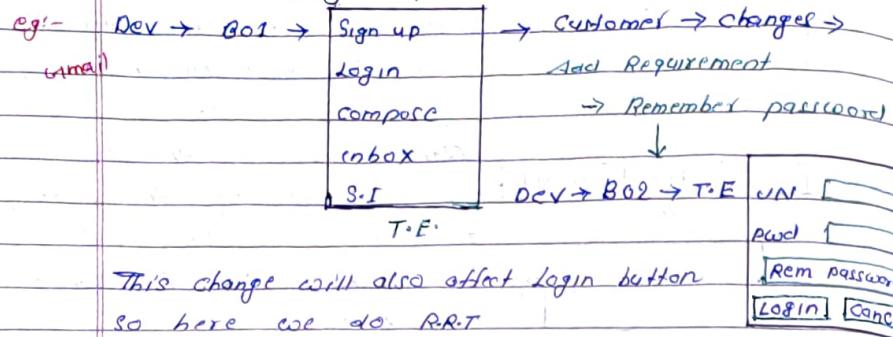
| | |
|---|---|
| A | B |
| C | D |

Test D and affected areas (A,B,C,D). This is R.R.T.

Date / /

Saathi

Q How to identify impacted areas? (Gmail)



If someone will reply to status video it will go to chat. So we have to do R.R.T.

→ T-E will interact with Senior T-E, T-L, Sen. Dev., BA & Customer. Sometimes discuss about the impacted areas & document the impacted areas, this is called an impact analysis meeting.

→ Customer when he give the req. he will only inform to dev. and test engineer about the impacted areas.

→ Business Analyst when he convert CRS to SRS, in the SRS there will be section called Impacted area, and BA will mention list of impacted area

Date / /

Saathi

→ Dev will give the information about impacted areas bcz he is the person who is doing code changes and he will be aware of which on the areas the code has been modified.

→ T-E will interact with T-L (or) T-M (or) Sr. T-E based on their product knowledge they will give list of impacted areas.

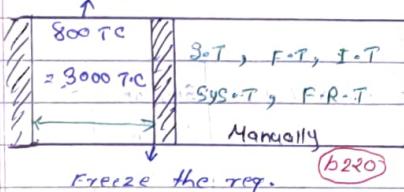
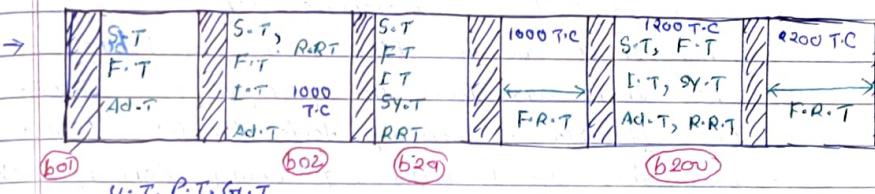
→ T-E will conduct Impact Analysis meeting.

3. Full Regression testing:-

Testing the changes and all remaining features of an appn is called F.R.T

Some companies after every 10-12 cycles they go for F.R.T, but it is not standard, it varies from Company to Company & project to project.

Customer → Req → Compagny → Dev → Coding → Q.B.T →



Date / /

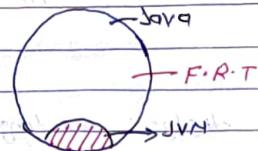
Saathi

Q) When to go for F.R.T?

Ans. a) When the changes are more, don't spend time in doing impact analysis, test the entire product by doing F.R.T.

b) When changes are alone in root of the product, then test the entire product by doing F.R.T.

Cy:-



Q) What is difference b/w Retesting & Regression testing?

Retesting

Regression testing

Ans. 1) Testing only the bug which is left after testing the unchanged is fixed or changed which features to make sure that it is not broken.
2) made in parallel because of changes is called as Regression testing.

Q) When to go for Automation Testing?

1. When product is functionally stable.
2. When there are no blockers or critical bugs.
3. When there are more no. of Regression T.Cases.
4. Once after 8/10 is manually tested for 1-2 releases.
5. When there are no major req. change done by customer.

Req. → T.E → Test.Cases

Dev → Code → C.B.T → (bal) → 8 months

(4) yrs

| | |
|---------|-------|
| Sign up | → T.E |
| Login | |
| A.T | |
| A.B | |

T.E → manually → customer → Production → End user
by writing A.T
T.Cases

Date / /

Saathi

Req. → T.E → T.Cases

Req. → Dev → Copy Prev. → write code → C.B.T → (2-4) months
Code for new features

| |
|------|
| Loan |
| H.L |
| Y.L |
| H.L |

(3-4) months → 4 hrs. manual (500 T.C.)
8 hrs/day → 4 hrs A.T (500 T.C.)

Req. → Newly added old features } manually → Defect → Dev → 4-months
500 T.C. within 1 yr
6-hrs - M.T customer → prod.

2-hrs → old T.Cases → convert to
Automation Scripts (700 T.C.)

Req. → Insurance → Dev → Code → C.B.T → T.E → 6 months

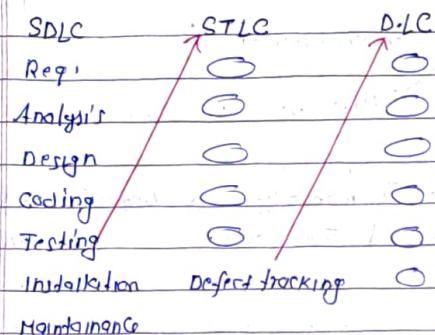
→ T.E → write T.Cases → Execution

Manual +
Automation

4-months

Software Testing Life cycle (STLC)

It is a procedure to test the SW.



Q difference b/w SDLC & STLC?

SDLC

1) It is the procedure to develop the SW.

STLC

1) It is procedure to test the SW.

2) It got Stages → Req., Analysis, Design, Coding, testing, installation, Maintenance

2) It got Stages → System Study, Prepare test Plan, write T. case, prepare T. matrix, Test exec., Defect tracking, Test execution Report, Retrospect meeting

STLC

System Study

Prepare Test Plan

write Test cases

Prepare Test matrix

Test Execution

Defect Tracking

Test Execution Report

Retrospect meeting

1. System Study:- Here we read the requirement and try to understand the requirement.

If we have any question, we have to interact with developer, BA, or customer and get it clarified.

2. Prepare test Plans:- Once we understand the requirement we should Prepare the test plan. Test Plan is a document which drives all the future testing activities. Here we decide

- How many Engg. needed to complete the testing.
- what each Engg. Should do in each stages of testing.
- what types of testing we have to conduct in future.
- what features to be tested and not to be tested.
- what Should be testing approach.
- what exactly we should start any activity and end any activity.

Date / /

Saathi

Date / /

Saathi

3. Write test cases:- Once planning job is done we start writing the test cases. This activity has got different stages like

- i) Identify all possible Scenarios.
- ii) Write the test case.
- iii) Test case review.
- iv) Fix the review comment.
- v) Verify the fix.
- vi) Test case approval.
- vii) Store it in test case repository.

4. Prepare Test matrix:-

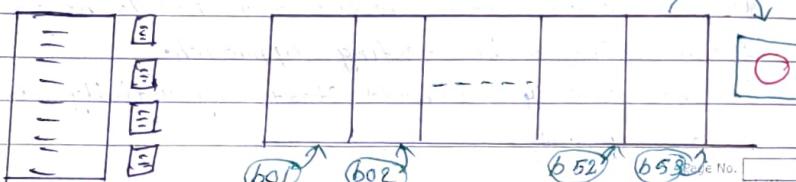
| Req. | Test-Case-name | Once test case are ready the biggest question is what is the guarantee that you have written test case for all the module. To ensure that |
|------|----------------|---|
| - | ~~~ | |
| - | ~~~~ | |
| - | ~~~ | |
| - | ~~~ | |
| - | ~~~ | |

We prepare traceability matrix

It is a document which ensures that every req. has got atleast one test case

5. Test Execution:- This is the stage where we test the product for 3-7 cycles, if it is agile and 40-60 cycles if it is non-agile model.

Here we conduct all the types of testing and we find bugs and help the developer to improve the quality of the product



6. Defect tracking:- We are executing the test means, definitely we are going to catch lot of bugs. Every bug we are trying to catch should be tracked in a very organized way that is called defect tracking.

7. Prepare test execution Report:- At the end of every test cycle we prepare an execution report. It is a document which covers

- i) How many test cases are there.
- ii) How many test cases are executed.
- iii) How many test cases are not executed.
- iv) How many test cases are Pass.
- v) How many test cases are fail.
- vi) Pass %.
- vii) fail %.

We send this report to customer in last test cycle and that is the end of project or release, from customer point of view.

But from project point of view, we have got one more activity i.e., retrospect meeting.

8. Retrospect meeting:- Here entire testing team meet and discuss about the list of achievements and mistakes (right and wrong process followed).

We document all the mistakes and achievement & this document is called 'Retrospect document'.

In next release / sprint / project at planning stage we open the old retrospect document and plan it in such a way that mistakes not repeated & right steps adopted.

Date / /

Saathi

when we do retrospect meeting at the end of the meeting we realize that no. of mistakes you have done is very less.

That is how we continuously improves the test lifecycle.

Q) what are the types of testing? (or)

classification of S/W testing?

Ans. → 1st classification basis

1. Static 2. Dynamic

↓
Functional

- 1> Functional T
- 2> Integration T
- 3> System T
- 4> Regression T

↓
Non-Functional

- 1> Usability T
- 2> performance T
- 3> Globalization T

→ 2nd classification basis

1. White-box testing

- 1> Path T
- 2> Loop T
- 3> Condition T
- 4> memory T
- 5> performance T

2. Black-Box-testing

- 1> Functional T
- 2> Integration T
- 3> System T
- 4> Regression T

→ 3rd classification basis

1> Manual Testing

2> Automation Testing

→ 4th classification basis

1> E-G 2> E-P 3> BVA

Date / /

Q) what are the levels of testing?

1. UNIT TESTING
2. FUNCTIONALITY TESTING
3. INTEGRATION TESTING
4. SYSTEM TESTING
5. ACCEPTANCE TESTING

Ans:

Q) what is diff. b/w verification and validation?

Verification:- It is also called as Review, walkthrough or inspection.

It is the process of collecting info and verifying or reviewing.

Validation:- It is actual testing of the S/W on basis of Ref. Specification.

| Verification | Validation |
|--|--|
| 1> It involves review, walk-through, inspection | 1> It involves actual testing |
| 2> It involves activities like a> req. review b> design review c> test case review d> test plan review e> code review | 2> It involves activities like a> F-T b> I-T c> S-T d> Smoke T e> C-T |
| 3> Here we check whether we are building right product | 3> Here we check whether we are building the product right |

Q Difference b/w static and dynamic?

Ans.

Static

Dynamic

1) It is a verification process 2) It is validation process.

- 3) It involves activities like
 a) Requirement review
 b) Design
 c) Test case
 d) Test plan
 e) Code
- 3) It involves activities like
 a) Functional Testing
 b) Integration
 c) System
 d) Smoke
 e) Adhoc

3) To do this we need not execute a program

3) To do this we should run the program

4) To do this we should have check list

4) To do this we should have test cases

5) We do this to prevent defects

5) We do this to find defects

Q What is diff b/w Functional & Non-Functional testing?

Ans.

Functional

Non-Functional

- 1) Here we test whether the app works according to Functional req. Specification
- 1) Here we test whether the app works according to non-functional req. Specification

- 2) Here we check whether feature is working or not
- 2) Here we check whether look, performance good or not.

- Q What is testing like?
 a) Functional testing
 b) Integration
 c) System
 d) Smoke
 e) Adhoc
 f) Acceptance
- 3) What is testing like?
 a) Load testing
 b) Stress testing
 c) Soak testing
 d) Volume testing

Q Diff. b/w Test Case and Test Scenario?

Ans.

Test Case

1) It is step by step procedure to test a feature

2) It contains:- Step No., description, input, E-R, A-R, Status, Comment

Test Scenario

1) It is one-line description of complete test

1) One Scenario can contain one test-case or multiple test-cases.

Login as user A apply for 5000, Logout → Scenario

1) Login as 'A' click on OD link

2) OD page will display

3) Click on apply OD and enter the amount as 5000

4) Confirmation msg. Should be displayed

5) Logout as 'A'

Test Cases

Date / /

Test PLAN

It is a document which drives all future testing activities.

It has got different Sections like :-

1. OBJECTIVE:- This Section covers aim of preparing the test plan.

2. SCOPE:- This Section covers what all the features to be tested and what all features not to be tested.

2.1: Features to be tested

inbox, compose, Login, Signup, ...

2.2: Features not to be tested

Help, antivirus, ...

Note:- Help is a feature developed by one technical writer and tested & reviewed by another technical writer so we will not test help feature.

→ If there is any 3rd party component we will not do any F.T, we will do only E.T, End-to-end, ...

→ Whenever you are preparing Test plan you have to consider certain features built in previous release to test, because adding new feature might have impact on old feature

→ Certain old features we might move it to outer scope because new feature might not have any impact.

There are certain features we might move it to be tested section, because it is exposed to

Date / /

end users and it needs Compatibility Testing

There are features which are used by internal users of the product, we may have to move it to not to be tested in different platform (to be tested only in base platform)

3. APPROACH:- This Section covers how we go about testing the product in future

3.1 By writing High Level Scenario

→ a) Login to Gmail

b) click on Compose

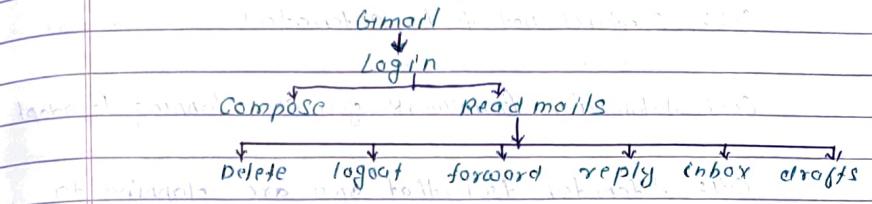
c) check it in S.I

→ a) Login to Gmail

b) Delete all mails from inbox

c) check in fresh

3.2: By writing flow Graph



4. SCHEDULE:- This Section covers when exactly, which activity should start and which activity to be completed.

| System Study | prepare T-Plan | write Test Case | Test Execution | Release date | Customer requested |
|--------------|----------------|-----------------|----------------|--------------|--------------------|
| Date 0/0/0 | Date 0/0/0 | Date 0/0/0 | Date 0/0/0 | Date 0/0/0 | Promote |

Test PLAN

It is a document which drives all future testing activities.

It has got different Sections like:-

1. OBJECTIVE:- This Section covers aim of preparing the test plan.

2. SCOPE:- This Section covers what all the features to be tested and what all features not to be tested.

2.1: Features to be tested
Inbox, compose, login, Signup,

2.2: Features not to be tested

Help, antivirus,

Note:- Help is a feature developed by one technical writer and tested & reviewed by another technical writer so we will not test help feature.

→ If there is any 3rd party component we will not do any F.T, we will do only E.T, End-to-end,

→ whenever you are preparing Test plan you have to consider certain features built in previous release to test, because adding new feature might have impact on old feature

→ Certain old features we might move it to outer scope because new feature might not have any impact.

There are certain features we might move it to be tested section, because it is exposed to

end users and it need compatibility Testing

There are features which are used by internal users of the product, we may have to move it to not to be tested in different platform (to be tested only in base platform)

3. APPROACH:- This Section covers how we go about testing the product in future

3.1 By writing High Level Scenario

→ a) Login to Gmail

b) Click on Compose

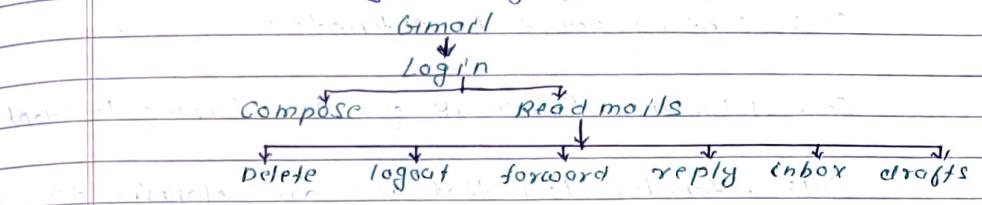
c) Check it in S.I

→ d) Log in → a) Login to Gmail

b) Delete all mails from inbox

c) Check in trash

3.2: By writing flow Graph



4. SCHEDULE:- This Section covers when exactly, which activity should start and which activity to be completed.

| System Study | prepare T-Plan | write T-case | Test Execution | Release date |
|--------------|----------------|--------------|----------------|--------------|
| Date 0/0/0 | Date 0/0/0 | Date 0/0/0 | Date 0/0/0 | Date 0/0/0 |

customer requested
Promise

5. TEST METHODOLOGIES:- This Section covers all types of testing techniques which we are going to conduct on product in future

| | | |
|---------------|------------|-----------------|
| Smoke Testing | common | → I18N |
| Functional " | Testing | → L10N |
| Integration " | techniques | → compatibility |
| System " | for | → usability |
| Adhoc " | all | → performance |
| Regression, | Prospects | |

6. TEST AUTOMATION:- This Section covers what are the features to be automated & what are the features not to be automated and the complete automation strategy.

6.1: Features to be automated

≡

6.2: Features not to be automated

≡

6.3: Automation framework you are planning to adopt

≡

6.4: Automation tool that you are planning to use

≡

7. EFFORT ESTIMATION:- In this Section they estimate how long it takes to complete the project, also they estimate how many Engg. & cost of testing needed to complete the task.

8. DEFECT TRACKING:- This section covers in future when we find defect how we should track, it also covers what should be the procedure of status by severity by priority

8.1:- Procedure to track defect

≡

8.2:- Severity

Blocked

Critical

Major

Minor

8.3:- Priority

High

Medium

Low

8.4:- Defect tracking tool

BUGZILLA

9. Automations:-

10. RISK:-

11. Mitigation Plan/Backup plan/Contingency plan:-

→ we are planning means we are assuming that we are going to do lot of tasks, based on that we promise the customer, because of few reasons, if we are not able to do certain task then that become RISK

→ To face the risk we have a plan that plan is nothing but mitigation or Backup plan

eg:- 1. we assume that everybody will be there in the project

Date / /

Saathi

2. If Someone quits, new fellow might miss lot of bugs that become risk
3. The Mitigation Plan is now 'engg' Should refer the f-cases and test the product with Sec. owner help
12. Test Environment: This Section covers how we go about Setting up the test environment in future.
- 12.1: Procedure to install the build

12.2: Hardware

12.2.1: Server Side

HP Start cat 1500

12.2.2: Client Side

6 computers with following Configuration

→ 1.93 Ghz, intel

→ 16GB RAM

12.3: Software

12.3.1: Server Side

OS → Linux

Web Server → Tomcat Version

App Server → Tomcat Version

DB Server → Oracle Version

12.3.2: Client Side

OS → win 10, win 8

Browser → mozilla, chrome

13. Deliverables: This Section covers what in the o/p that should be given by the testing team at end of the test cycles

13.1: Test Cases

Date / /

Saathi

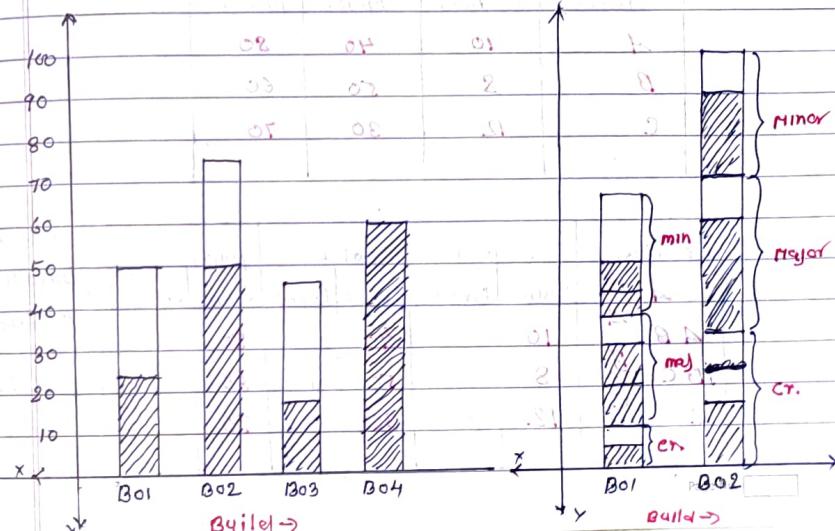
- 13.2: RTM / CRM
- 13.3: Test Execution Report
- 13.4: Defect Report
- 13.5: Graph & matrices
- 13.6: Release note

Graphs: Graphical representation of data to quickly analyse

a) Defect distribution graph

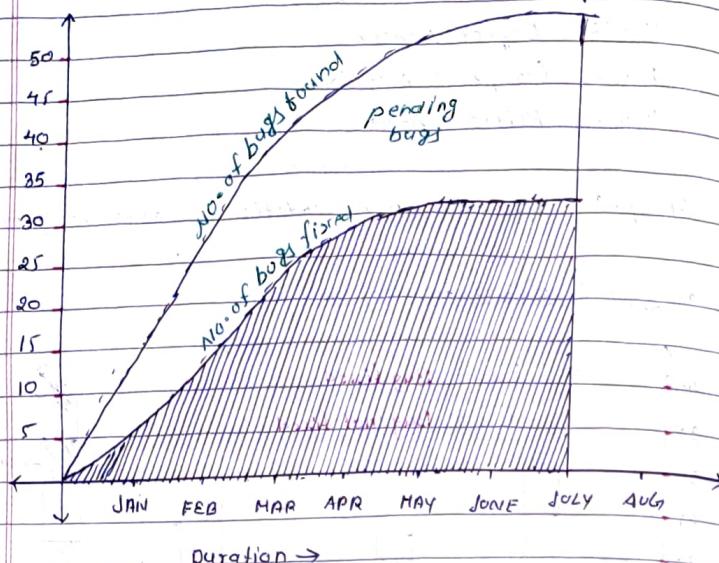


b) Build wise Defect distribution Graph:



Date / /

Saathi

c) Defect trend Analysis Graph:-

(Reports) / Matrix:- It is a past data, which is gathered b/w 2 parameters so that we can do proper analysis and take decision for future testing activity.

a) Defect duration matrix:-

| Features | Critical | Major | Minor |
|----------|----------|-------|-------|
| A | 10 | 40 | 80 |
| B | 8 | 50 | 60 |
| C | 12 | 30 | 70 |

| Feature | Critical | | Major | | Minor | | Defect Duration matrix's in detail |
|---------|----------|-------|-------|-------|-------|-------|---------------------------------------|
| | Found | Fixed | Found | Fixed | Found | Fixed | |
| A | 10 | 8 | 40 | 25 | 80 | 40 | |
| B | 8 | 8 | 50 | 40 | 60 | 40 | |
| C | 12 | 10 | 30 | 20 | 70 | 50 | |

Date / /

Saathi

b) Test Efficiency matrix:-

| T.E Name | Critical | Major | Minor | |
|----------|----------|-------|-------|----------------|
| A | 10 | 40 | 80 | → 130 |
| B | 15 | 30 | 85 | → 130 |
| C | 12 | 50 | 90 | → 152 |
| D | 15 | 60 | 80 | → 155 Hardwork |

c) Build wise Defect distribution matrix:-

| Build ID | Critical | Major | Minor |
|----------|----------|-------|-------|
| B01 | 10 | 40 | 80 |
| B02 | 8 | 50 | 60 |
| B03 | 12 | 30 | 70 |

| Build ID | Critical | Major | Minor | |
|----------|----------|-------|-------|----|
| found | fixed | found | fixed | |
| B01 | 10 | 8 | 40 | 25 |
| B02 | 8 | 8 | 50 | 40 |
| B03 | 12 | 10 | 30 | 20 |
| | | | | 70 |
| | | | | 50 |

d) Productivity matrix:-

$$\text{Test. Efficiency} = \frac{\text{No. of defects found by testing team}}{\text{Total no. of defects found by testing team + found in Acceptance testing + found in production}} \times 100$$

Total no. of defects found by testing team + found in Acceptance testing + found in production

13.6: Release Note:- Along with product we release one note to customer i.e. called release note

Date / /

Saathi

gt Contains

- 1) List of open defects (or) known defects that are still there in product.
- 2) List of platforms in which product is tested.
- 3) List of platforms in which product is not tested.
- 4) List of new features added, modified or removed.
- 5) No. of bugs found in previous release production and fixed in current release.
- 6) Version no and installation steps

14.1 Entry and Exit criteria:-

Entry:- gt is list of criteria that should be met to start the activity.

Exit:- gt is the list of criteria that should be met to say that activity is over.

14.1 Entry criteria for F.T:-

- 1) W.B.T Should be over.
- 2) Build Should be installed with proper testing environment.
- 3) Test cases are ready.
- 4) Test data Should be available.
- 5) Resources Should be available.

14.2 Exit criteria for F.T:-

- 1) % of test cases executed 90%.
- 2) % of test cases pass 85%.
- 3) Should not cross > 10 critical bugs.
- 4) Should not cross > 40 major bugs.
- 5) Should not cross > 80 minor bugs.

Date / /

Saathi

Entry criteria for I.T:-

- 1) Should met with exit criteria of F.T.
- 2) W.B.T should be over.
- 3) Test cases are ready.
- 4) Build Should be installed with proper Testing environment.
- 5) Test data Should be available.
- 6) Resources Should be available.

14.4 Exit criteria for I.T:-

- 1) % of test cases executed 95%.
- 2) % of test cases pass 90%.
- 3) Should not cross > 5 critical bugs.
- 4) Should not cross > 30 Major bugs.
- 5) Should not cross > 60 Minor bugs.

14.5 Entry criteria for S.T:-

- 1) Should met with exit criteria of I.T.
- 2) upto whatever we developed all features should be functionally stable.
- 3) Minimum bunch of features must be ready (critical-features).
- 4) Testing Environment which is similar to production environment must be ready.
- 5) W.B.T Should be over.
- 6) Test cases are ready.
- 7) Build Should be installed with proper testing environment.
- 8) Test data Should be available.
- 9) Resources Should be available.

Date / /

Saathi

14.6 Exit Criteria for S.T:-

- 1) % of test cases executed - 100%
- 2) % of test cases Pass - 98%
- 3) Should not be any critical bug
- 4) Should not cross > 20 major bugs.
- 5) Should not cross > 40 minor bugs.

15. Roles and Responsibility: It talks about work allocation done for Engg. for project.

15.1 Rules and Responsibility of Test Manager :-

- 1) Write and review test plan
- 2) Interact with testing team, development team, project Manager & if needed customer.
- 3) Handle issues and escalation.
- 4) Effort estimation of resourcing.
- 5) Approve release note.

15.2 Test Lead:-

- 1) Write and review test Plan
- 2) Allocate work to test Engineer & make sure that they are going to complete the task within the schedule.
- 3) Consolidate the reports which are sent by every test Engg. and communicate with testing team, development team, Project Manager & customer.
- 4) Conduct Impact Analysis meeting.

15.3 Test Engineers:-

- 1) Review test Plan.
- 2) write test Cases for associated feature.

Date / /

Saathi

- 3) Review test Cases of other test Engineers.
- 4) Execute test Cases written on allocated feature.
- 5) perform Compatibility testing.
OS → win 7, win 8, windows
web browsers → I.E., mozilla, chrome

16. Template:- This Section covers formats for all the document that you are planning to prepare in entire test life cycle.

| Test Cases | RIM | D.R | Test Case Review Report | Test Execution Report |
|------------|-----|-----|-------------------------|-----------------------|
| == | == | == | == | == |
| == | == | == | == | == |
| == | == | == | == | == |

17. Test Stop Criteria:-

→ we should stop testing if the product quality is very good or very bad.

17. Product quality is good means:-

- 1) All the features requested by customer should be ready.
- 2) If all end-end business Scenario are working fine.
- 3) There are O-O, O-C and few bugs left out but they are all less than limit set by customer.
- 4) You should have tested the product in env. Similar to production environment.

Date _____

Saathi

- Q) Product quality bad means
 or There are too many blocker/critical bugs.
 b) If many end-end Scenario's (critical) not working.
 c) If it's crossing the budget.
 d) If it's crossing the Schedule.

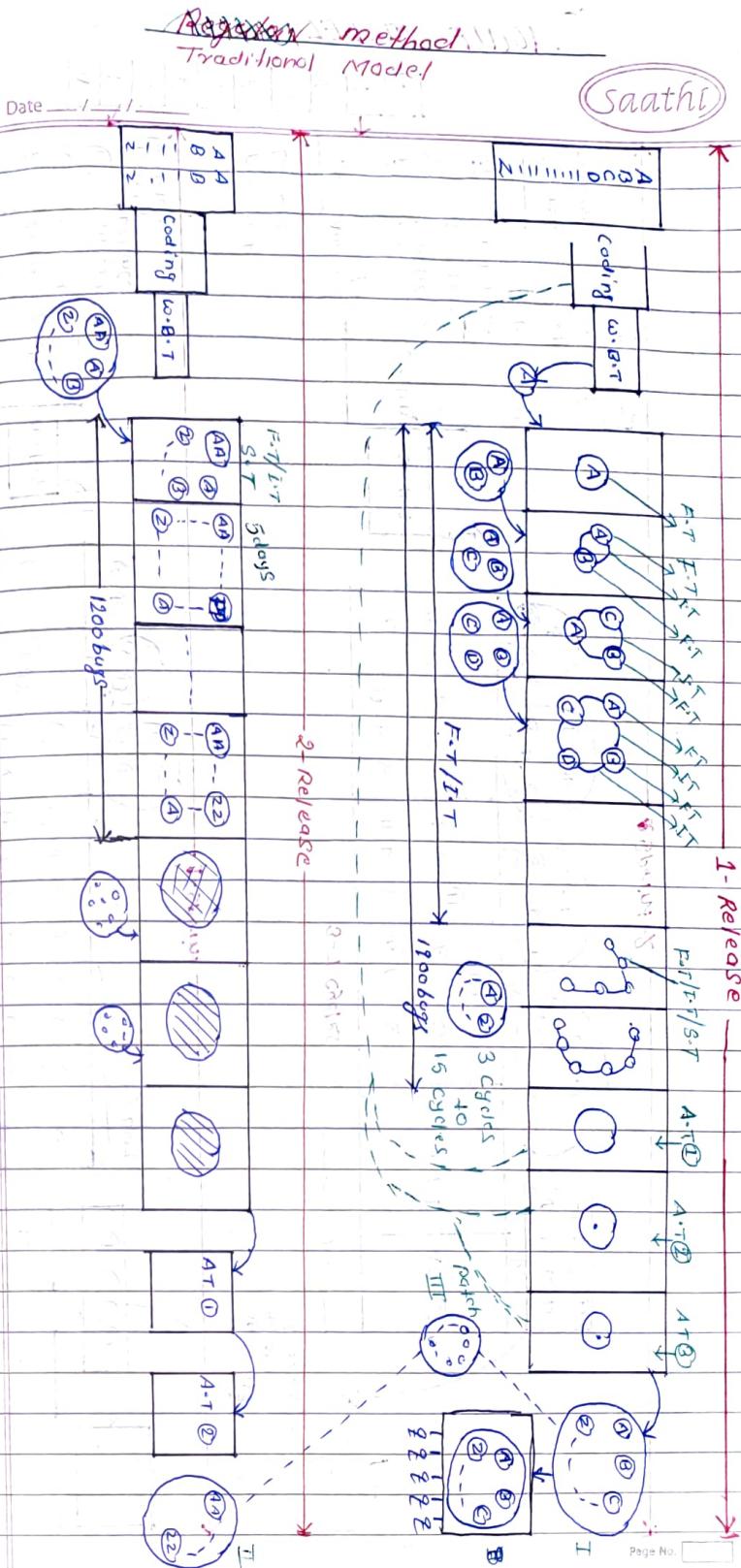
AGILE METHODOLOGY

Hotfix or Incident Management :-

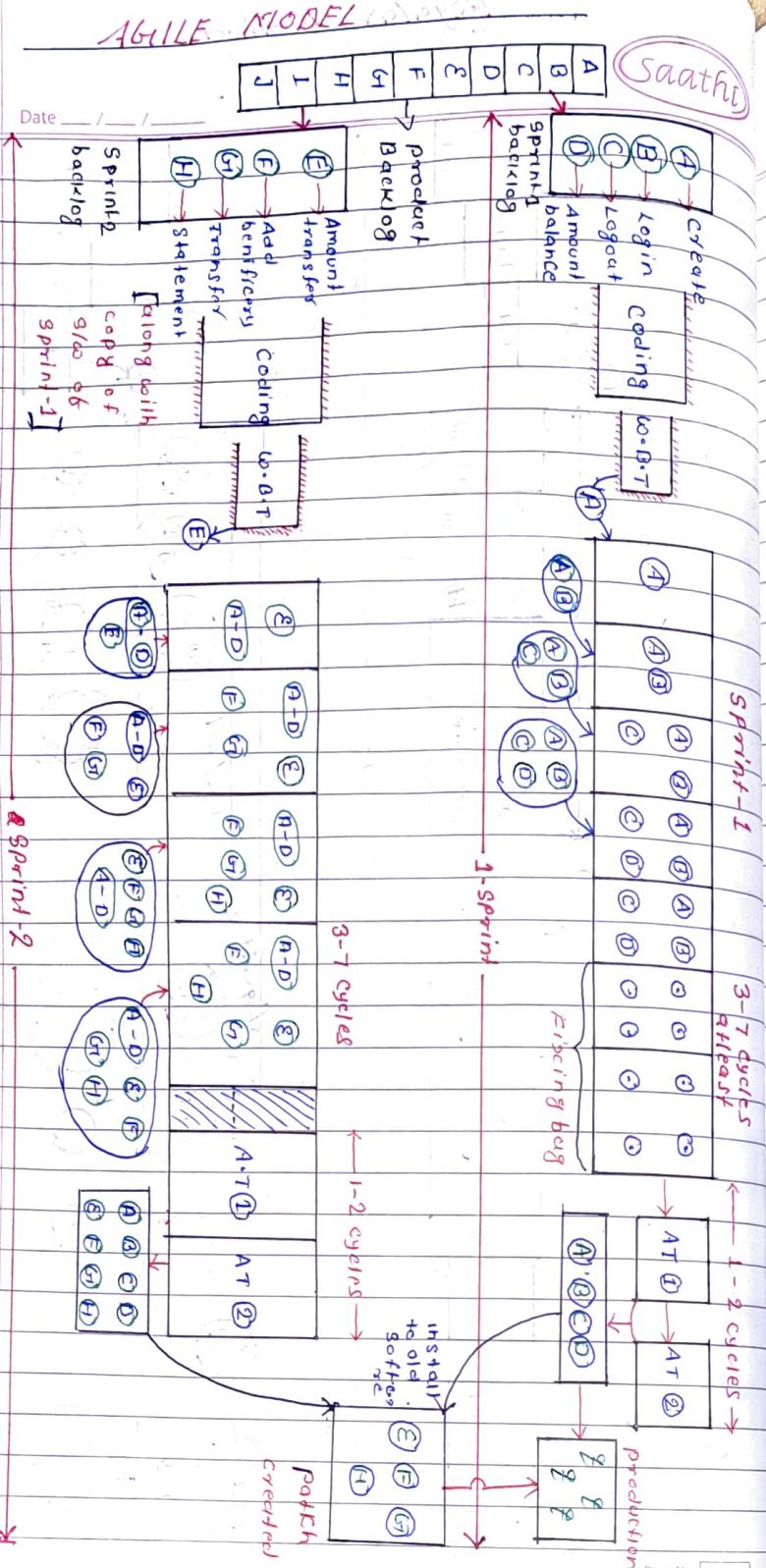
- a) If end-user will find blocker/critical defects while using customer will say to fix it.
- b) Company will take prev. released module, take few D.F & T.F from running project and fix the bug and test the bug.
- c) After fixing and testing product given to customer, & they will do Acceptance testing.
- d) After A.T the patch will be created and moved to production.
- e) This whole process is called as hotfix (or) incident management and this take 3hr to 3 days.

R.C.A.:-

It is root causing Analysis in which Project Manager calls all members of Dev. and testing for a meeting to discuss the reason causing defect. This is also called as FISHBONE Technique (or) ISHIZAWA Method.



AGILE MODEL



Date / /

PRODUCT BACKLOG

LOAN

| HOMELOAN | PERSONAL | VEHICLE |
|----------|----------|---------|
| APPLY | APPLY | APPLY |
| APPROVE | APPROVE | APPROVE |
| Balance | Balance | Balance |
| Repay | Repay | Repay |

→ ARCHITECT
HLD

INSURANCE

| | LIFE | HOME | VEHICLE |
|---------------|--------------------------|--------------------------|--------------------------|
| APPLY INS | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| APPROVE INS | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| CLAIM INS | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| APPROVE CLAIM | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

CORE BANKING

| | |
|-------------|----------------|
| Create A/c | Amount Balance |
| Approve A/c | Amount Tr. |
| Login | Transaction |
| Logout | Statement |

SPRINT BACKLOG

| | |
|----------------|---------|
| 1. Create A/c | LID |
| 2. Approve A/c | UID |
| 3. Login | Coding |
| 4. Logout | Testing |

SCRUM TEAM

| | |
|---------------|-------------|
| core team | Shared Team |
| Scrum master | O/I/B admin |
| Product owner | Architect |
| Developer | B.A |
| T.E's | Devops |
| | N/W Admin |

SPRINT Planning:-

1. Assign the Stories/tasks to Engg.
2. Each Engg. Prioritizing the Stories/tasks.
3. Estimate how long each tasks takes to complete.
or Each Stories takes to build 8 test

| Stories | Time taken to build by Dev. | Time taken to test | Total Time | Story point |
|-------------|-----------------------------|--------------------|------------|-------------|
| Create A/c | 30 | 10 | 40 | |
| Approve A/c | 20 | 6 | 26 | |
| Login | 10 | 3 | 13 | |
| Logout | 8 | 3 | 11 | |

AGILE MODEL:-

- a) Agile is a model where in we develop the S/W in a incremental & iterative process.
- b) They came up with this model in order to overcome the drawbacks of traditional model.
- c) Here they build large products in shorter cycles called Sprint.

SCRUM:-

It is the process used to build the app in agile model

SPRINT PLANNING:-

- i. Here entire Scrum team sit together and pull the Stories from product backlog.
- ii. Scrum master assign each Story to D.E & T.E's.
- iii. Each Engineer drive the tasks to be completed to build each story.
- iv. Each Engineers estimate time taken to complete their each tasks, i.e. they derive story points.

SCRUM TEAM:-

1. It is the group of Engineers working towards completing committed features or stories.
2. Generally Scrum team will have 5-12 people.
3. It includes Shared team and Core team members.
4. Core team includes Scrum master, Product owner, Dev & T.E's.
5. Shared team includes O/I/B admin, Architect, B.A, Devops UI & UX designers.
6. Scrum master leads this entire Scrum team & he facilitates Engineers to complete their tasks.

Following are the roles played by different people in Sprint planning meeting:

1. SCRUM MASTER:-

- a) This complete meeting driven by Scrum master.
- b) His prime role is to facilitate the complete meeting and co-ordinate b/w all Stake holders (relevant people).

eg:-

- a) Getting architect support to the developers.
- b) Getting QA Support to testers
- c) Getting product owner support to tester to understand the requirement.

2. PRODUCT OWNER:-

- a) He clarifies if any question are there related to stories or req.
- b) He will set acceptance criteria for every Story.

3. DEVELOPMENT ENGINEER:-

- a) He should derive the task for build of every Story.
- b) He prioritize which Story to be build first & which which Story to be build later in the Sprint.
- c) He prioritize the task.
- d) He derive the Story points (estimate for each task)

4. TEST ENGINEER:-

- a) He will derive the tasks to be completed to test each feature/Story.

Create A/C → write test cases

Review test cases

Prepare traceability matrix

Execute test cases

Defect tracking

Q what is Acceptance Criteria?

- A. a) It is a criteria which should be met in order to move the product to production or to customer.

b) It is generally set by product owner or customer.

c) Generally they set this in the beginning of the project Sprint i.e. in Sprint planning meeting

Approve A/C acceptance criteria:-

eg:- Create A/C should go to pending approval page

Create A/C acceptance criteria:-

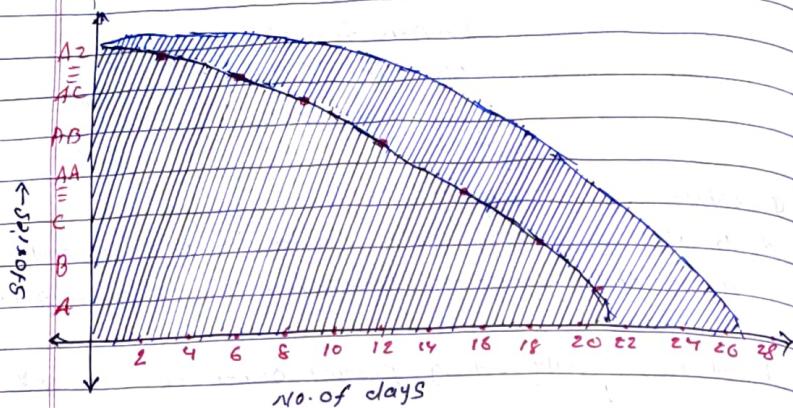
- a) User should be able to login
- b) Should be listed in RBL list
- c) Should be listed in Income tax list.

5. SPRINT BACKLOG:-

- a) It is list of Stories & task (Associated) committed by Scrum master/Team that must be delivered within one Sprint.
- b) It is actually committed or agreed by Scrum team.

6. PRODUCT BACKLOG:-

- a) Generally product owner, QA, customer, architect, Scrum master will be involved while building it.
- b) Here Stories need not to be detailed.

7. BURN DOWN CHART:-8. RETROSPECT MEETING:-

- a) Here entire scrum team meets, sometimes customer also joins & discuss about our achievements & Right process followed and mistakes & document it, this document is called Retrospect document.
- b) When next release/sprint starts while doing the sprint planning we refer this document & we plan it in such a way that old mistakes have not repeated & good activities are once again adopted.

9. STAND UP MEETING / SCRUM MEETING / DAILY SCRUM / ROLL CALL MEETING:-

- a) Here entire scrum team meets
- b) This meeting completely driven by Scrum Master
- c) Here Every Engineer Should Explain:-
- 1) what they have done yesterday.
 - 2) what are the implements or hurdles faced.

- e) 3) What are the activities he is planning to do today.
- f) 4) What are the implements he is expecting in order to complete the job.
- d) Scrum master tries to solve certain implements in meeting, if it takes too much time then Scrum master note it down and solve it later.
- e) Generally this meeting should be completed within 10-15 min.
- f) This meeting should be conducted in the beginning of the day.

- g) Here everybody should stand up in meeting so the people only talk to the point.

10. STORY BOARD:-

Contain list of tasks or stories.

- a) In progress
- b) Completed
- c) Pending

11. CHICKEN:-

Chicken is the one who observe and try to understand how sprint is going on & he will not be doing any task.

12. STORY POINT:-

- a) It is the estimation of how long task would take to complete
- b) How long story is going to take to develop & test

13. They will use a word called DONE DONE DONE i.e., W.B.T is done by DEV, Q.B.T is done by T.E and A.T is done by product owner.

14. There are 4 Scrum ceremonies :-
 a) Sprint planning meeting
 b) Daily Scrum
 c) Sprint Review
 d) Sprint Retrospect meeting

SPILL OVER:-

There are certain features/stories which you are not able to build in this sprint & you are postponing to next sprint that is called Spill over.

Q How do you test a pen?

- Note:-
 a) Tell only scenario not type of testing
 b) Don't worry about filling the table, worry about filling the scenario
 c) Always tell basic scenario first.

Smoke testing:-

- Check whether comfortable to hold or not.
- Check whether properly writes or not.

Functionality testing:-

- Check the height, weight, diameter & thickness of the pen.
- Check height, diameter, thickness of each component.
- Check height of refill, height of ink filled, height of nib.

Integration testing:-

- Will take the cap and then pen whether it properly integrates or not.
- Thread and unthread the pen, check whether it integrates properly or not.
- Remove the refiller and put it again & check if it fits properly or not.
- Remove the nib and put the nib again.

ADHOC testing:-

- Throw out the pen & check whether it writes or not.
- Write opposite to gravitation power.
- Write at different angles.
- Throw pen in water then write.
- Write on wet surface.
- Write on oily surface.
- Keep pen under sunlight & then write.

usability testing:-

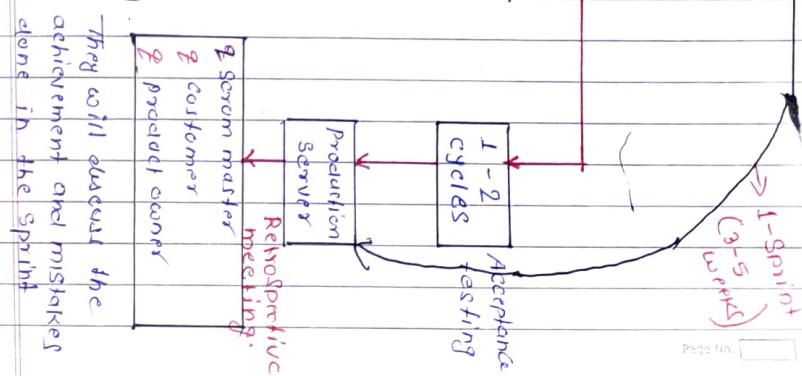
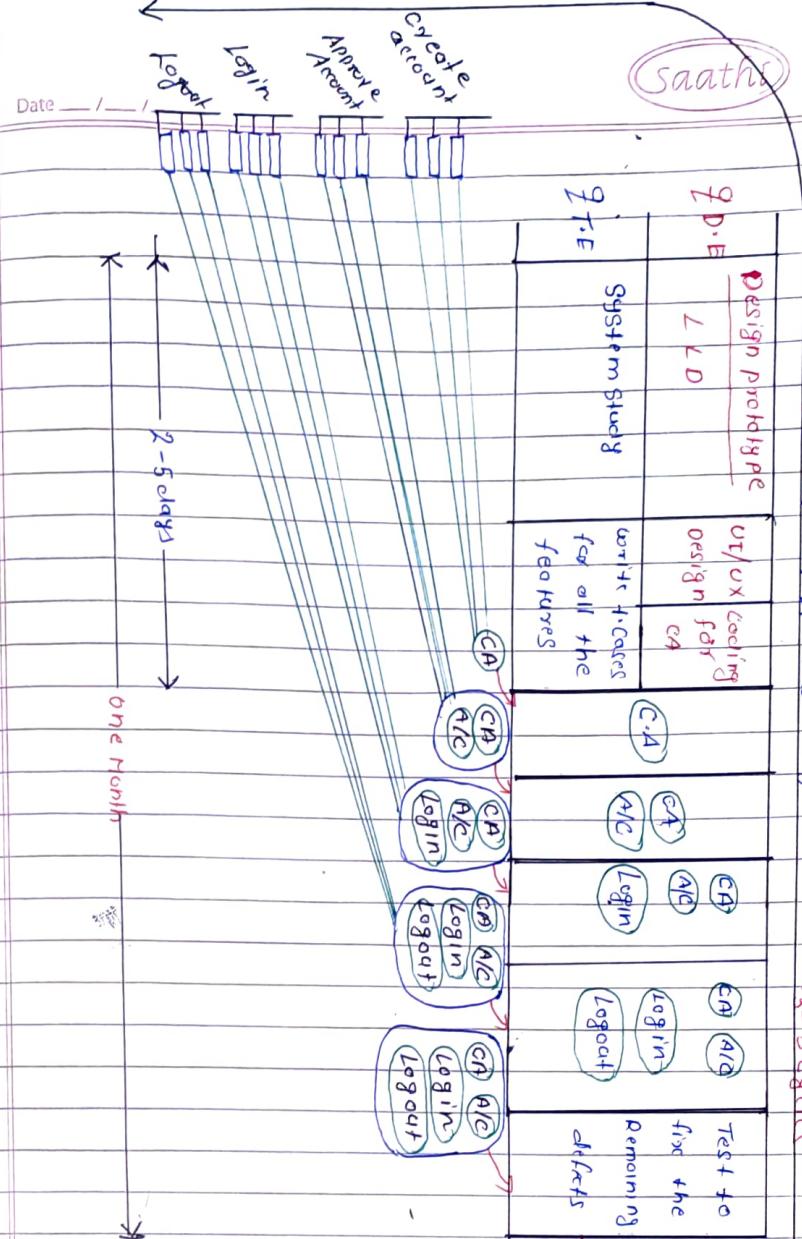
- Check whether it is user friendly or not.
- Check look and feel.
- Check is it comfortable to hold.
- Check is it pains if held for long time.

Accessibility testing:-

- Check whether complete body manufactured in rect. & Bixen bcz it affects color blind people.
- Check whether it is user friendly for physically challenged people.

Compatibility testing:-

- Check pen on different surfaces:-
- a) paper b) wet paper c) skin d) glass e) wall



Date / /

*(Saathi)*ISO testing:-

- checks whether the text printed on pen is in right language

LION testing:-

- check whether cost, pin code, date of manufacturing according to company standards or not.

Performance testing:-

- keep more pressure while writing and check that nib withstand the load
- keep writing with pressure on nib & check if what extent it withstand the load

Regression testing:-

- If any design changes are there in pen after manufacturing, test the unchanged feature to ensure that changes are not having impact.

Comparison testing:-

- Compare the pen with competing pen & list all the advantage and disadvantage of given pen.

Recovery testing:-

- Drop the pen and then write - observe how fast it recovers from crash.

Reliability testing:-

- continuously write for long duration and check whether quality remains the same.