

Notes

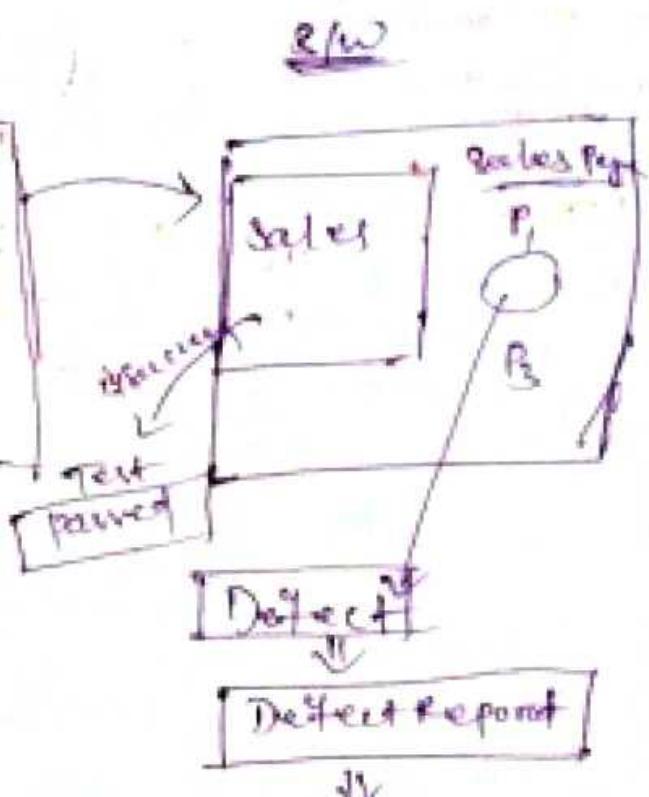
Software Testing

27/10/22

- ↳ The process of finding defect in a software we call it as Software Testing.

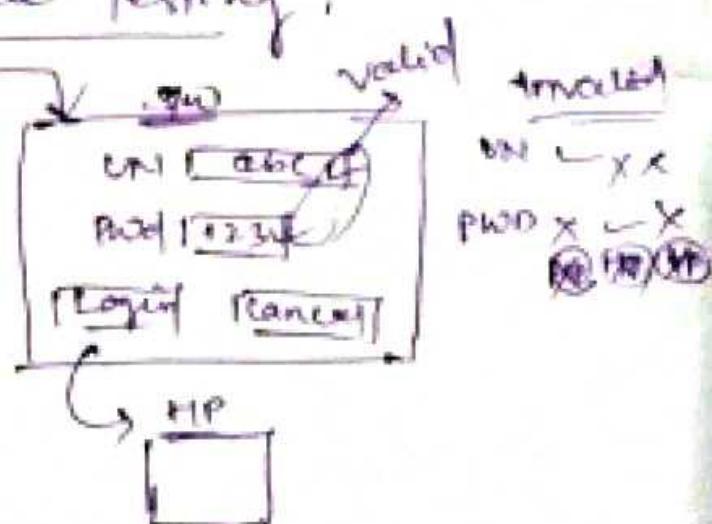
Requirement

- click on sales link
- Sales Pages should be displayed.
- In Sales Page P₁, P₂, P₃ should be displayed

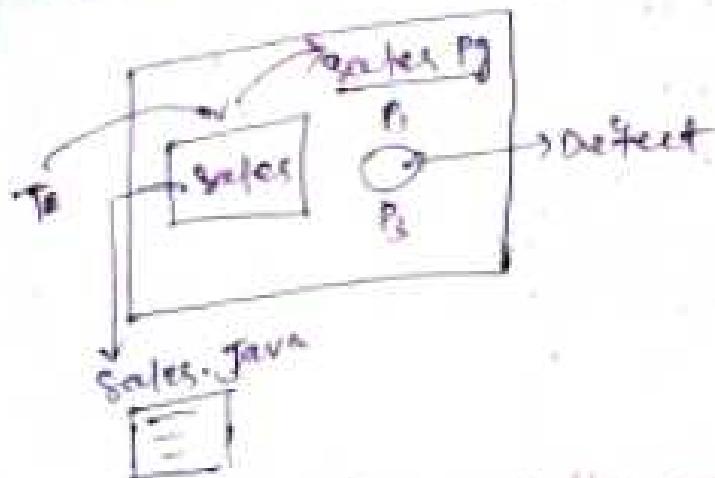


- ↳ Verifying the functionality of an application against the Requirement Specification we call it as Software Testing.

- Req
- Enter the Valid UN & password
 - Home Page should be displayed



→ Execution of a Program with the intent to find defect in a Software is called it as Software Testing.



→ Why we do software Testing?

- ↳ Every Software is Developed to support one or the other Business. If the Software has defect then the customer will under go huge loss due to being a Test Engineer we have to identifying all that defects.
- ↳ To Avoid Customer Loss we do software Testing.
- ↳ We do software Testing to improve the quality of the software.
- ↳ We do software Testing to check whether the software is matching the requirement or not.

Types of Software Testing

28/07/22

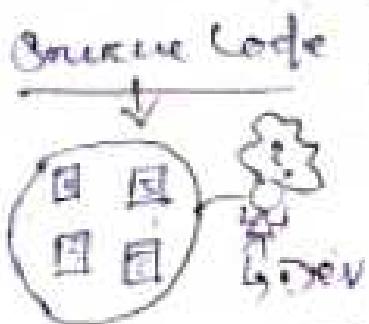
→ We have 2 different types of Software testing.

- (1) White Box Testing (or Unit Testing)
Open box testing
- (2) Black box Testing
Closed box Testing
Glass box Testing
Functional Testing
Non-functional Testing
Opaque Testing

① White box Testing

→ Testing each & every line of the Source Code we call it as White Box Testing.

→ It is done by Development Engineers.



→ White box Testing is also called as Unit Testing.

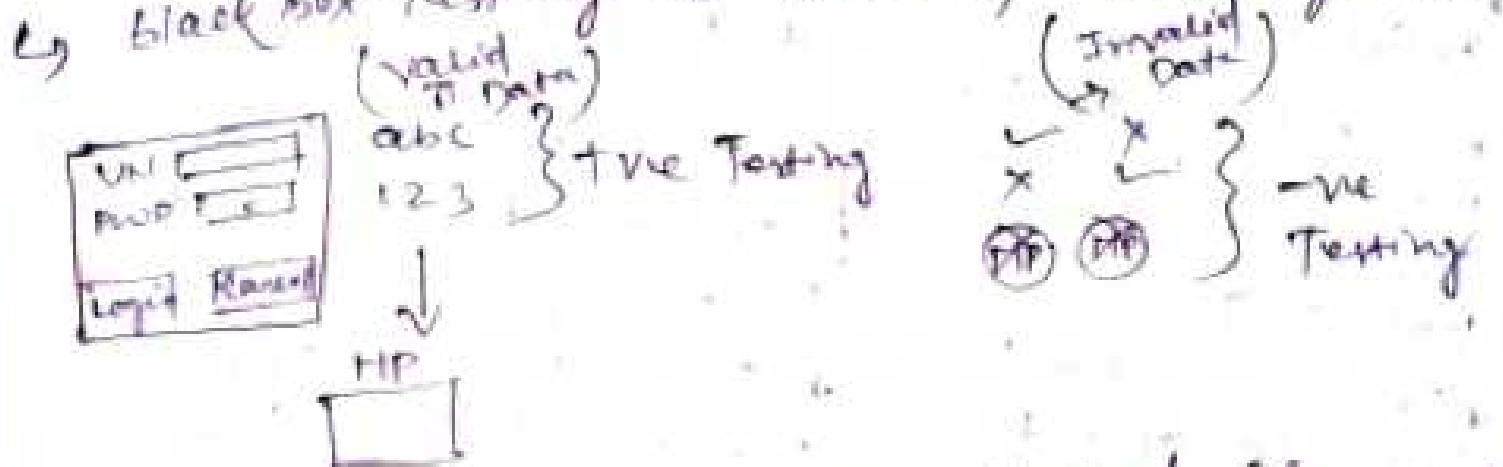
→ We call it as Unit Testing because we consider each & every line of the code as single unit.

→ White Box Testing is also called as Glass box Testing / Open box Testing, because we are able to see the source code.

Black Box Testing / Functional Testing

↳ Black Box Testing means Verifying the functionality of an application regard to the requirement specification. we call it as Requirement Testing.

black box testing is done by Test Engineers.



↳ Black Box Testing is also called as Closed Box Testing / Opaque Testing. Because we can not see the source code.

Types of Black Box Testing :-

- 1) Functionality Testing 2) Usability Testing
- 3) Integration Testing
- 4) System Testing
- 5) Acceptance Testing
- 6) Smoke Testing. → in short
- 7) Adhoc Testing → random
- 8) Regression Testing
- 9) Compatibility Testing / Configuration
- 10) Performance Testing / Non-functional
- 11) Exploratory Testing
- 12) Globalisation Testing

White Box Testing

- * Testing Each & Every line of the Source Code, we call it as White Box Testing.
- * It is Done By Developer.
- * To do White Box Testing we need to have good programming knowledge.
- * To do White Box Testing we should able to see the source code.
- * It is called as Unit testing.

Black Box Testing

- * Verifying the functionality of any application against the requirement specification, we call it as Black Box Testing.
- * It is Done By Test Engineer.
- * To do Black Box Testing need not to have good programming knowledge.
- * To do Black Box Testing need not to see the source code.
- * It is called as Functional Testing.

Interview Question of Software Testing :-

- 1) What is Software Testing?
 - ↳ Tell 2 diff's
 - ↳ Tell why we do
 - ↳ Tell types.
- 2) Explain White Box Testing in Detail?
 - ↳ Tell Diff's
 - ↳ Tell types of white box Testing
- 3) Explain Black Box Testing in Details?
 - ↳ Tell Diff's
 - ↳ Tell types of Black Box Testing
 - ↳ Explain each type quickly.

- Q) What is the Diff b/w White Box Testing & Black Box Testing?
- Q) How we do Perform White Box Testing & Black Box Testing Manually as well as automation of?
- Q) What are the tools that is used to perform White Box Testing?
- Q) What are the tools that is used to perform Black Box Testing?
- Q) Should the technology should be same for writing Automation Script in White Box Testing?

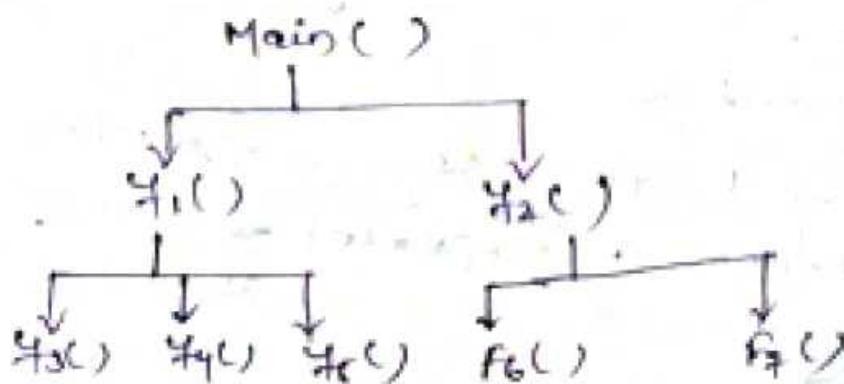
29/10/22

Types of White Box Testing :

There are 5 types of White Box Testing

- (1) Path Testing
- (2) Conditional Testing.
- (3) Loop Testing
- (4) Memory Point of view
- (5) ~~functional~~ Performance Point of view

(1) Path Testing



Flow chart of Program

- ↳ In Path testing we write the flow chart of all the programs written by Developers.
- ↳ Here we do not Miss Testing Each One of the Program.
- ↳ It prevents the testing of same code again & again.

(2) Conditional Testing ↗

Ex

<u>Program</u>
<pre>If (Rating > 112) { placement } else { sit and study }</pre>

- ↳ In Conditional Testing We check the Condition of the Code both true & false values.

(3) Loop Testing ↗

- ↳ In this we test the Looping Statement of the Code.
- ↳ Looping Statement ↗

↳ We use Looping Statement when we have to Perform a ^{repetitive} ~~repeating~~ action.

Types of Looping Statement :-

- (1) - Do While Loop
- (2) - While Loop
- (3) - for Loop

Ex White Loop

```
int a=1  
while (a<11)  
{  
    print(a)  
    a=a+1;  
}
```

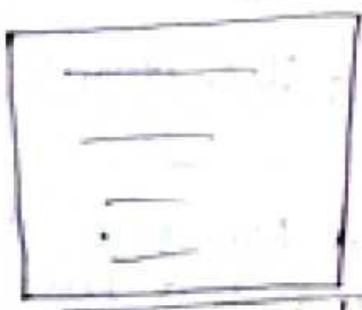
(4) White Box Testing from Memory Point of View

↳ Developers while writing the code will make mistake this will increase the size of the program on the memory of the program.

(i) Memory point of view because of lack of proper logic.

Not using logic

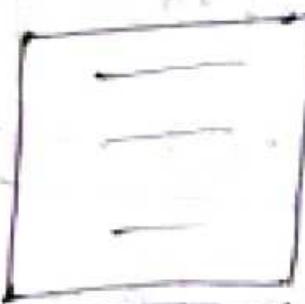
Ex-



Basic Code

Engines-1

(Not using looping statement)

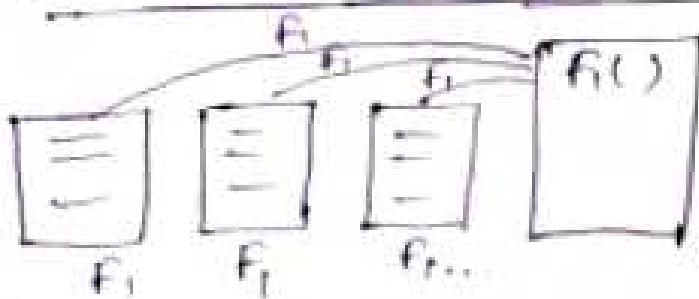


Good logic

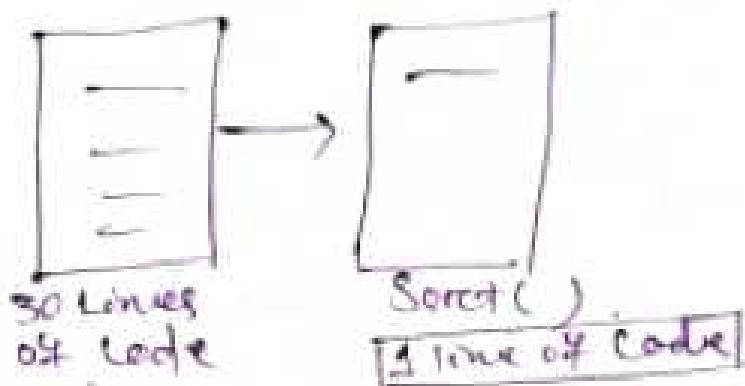
Engines-2

(using looping statement)

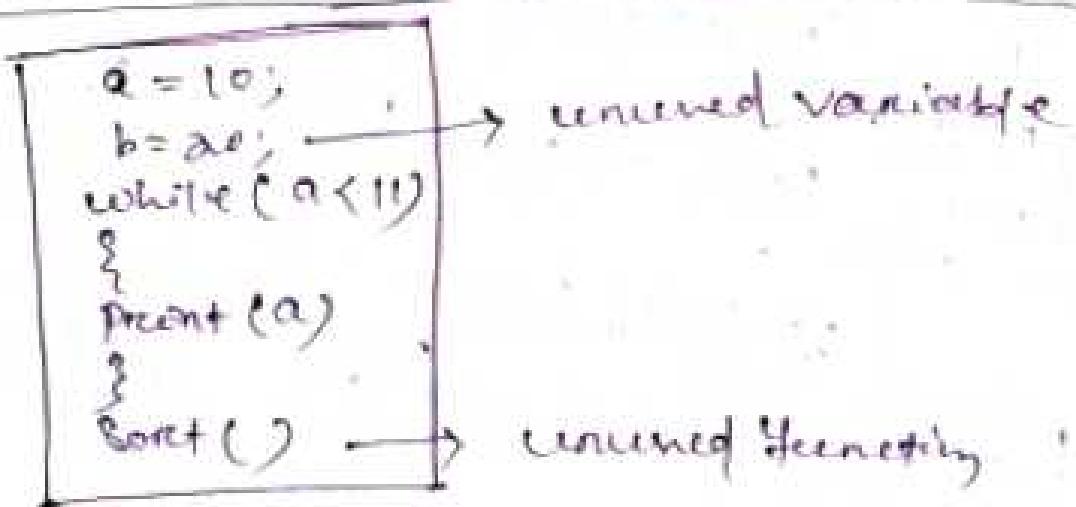
(ii) Because of Not Using Functions :-



(iii) Because of Not Using Built-in Functions:-



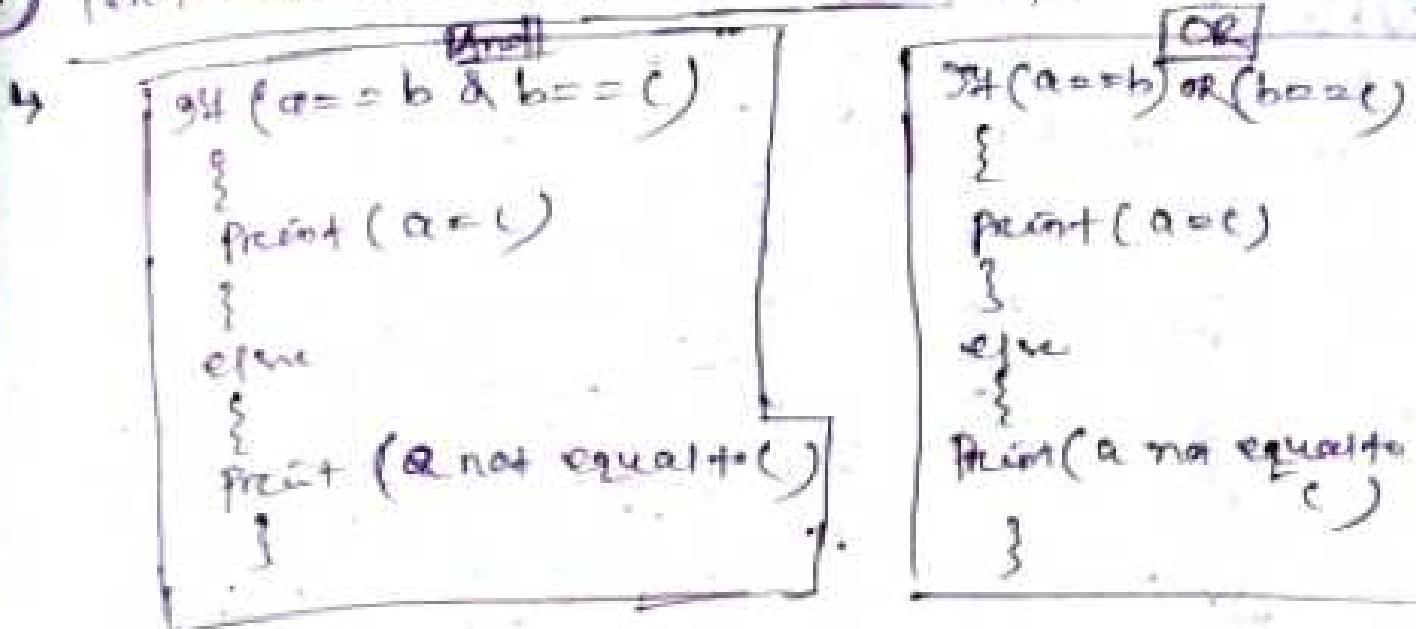
(iv) For Not Using or Unnamed Variables (functions).



↳ By we have repeated lines in the code we will forget variables or line in a diff file & call other file whenever we needs to used it.

5

Performance Point of View \Rightarrow



(i) Because of Inappropriate Logic \Rightarrow

* Whenever we are using OR Operator
we must see that the 1st Operand should
be false rest all are fine.

* whenever we are using And Operator
we must see that the 1st Operand should
be Value for Max number of times.

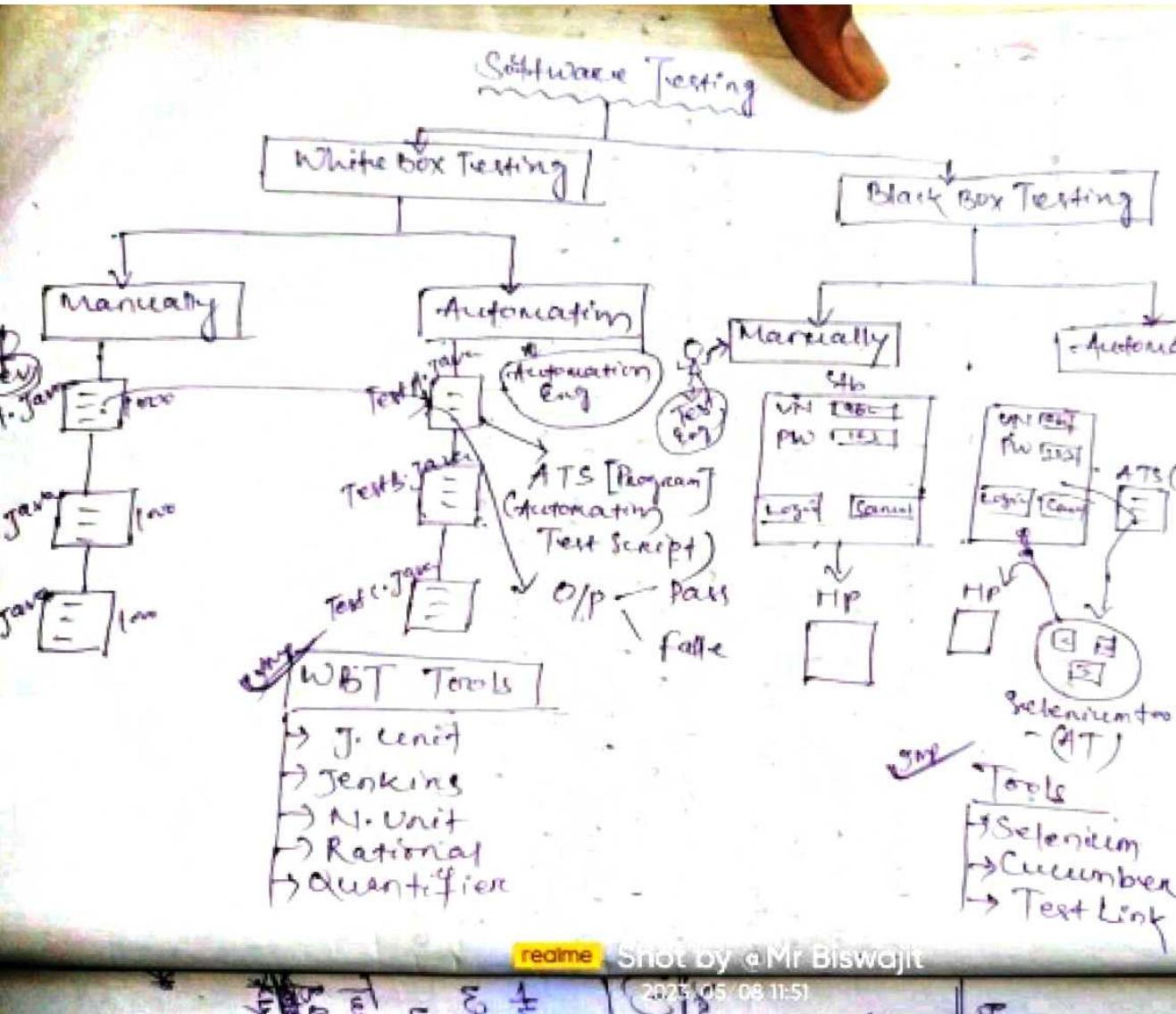
* Adding the program with the bottlenecks to
speed the execution of code we call it
as Performance Bottleneck Point of view.

Q How we do perform White box testing manually & Black box testing manually using Automation

W/B T Tools

- J. Cenit
- Jenkins
- N. Unit
- Rational
- Quantifier

→ Selenide
→ Cucumber
→ Test Link



White Box Testing Tools :-

We have multiple white box testing tool like :-

- ① J-Unit
- ② gherkin
- ③ N-Unit
- ④ Rational Quantifier
- ⑤ Load Runner.

Black Box Testing tools :-

We have multiple black box testing

- tools like :-
- (1) Behaveiro
 - (2) cucumber
 - (3) Test Link
 - (4) QTP (HP)

Black Box Testing

- (1) Functional Testing / field level Testing / component Testing
 - ↳ Testing each & every component individually against the requirement specification we call it as Functional Testing.
 - ↳ functionality Testing should be done by Test Engineers as it is the 1st type of Black Box Testing.

Component Means → Text Box, Radio ~~Box~~ button, Check Box, etc.

- * Threading means → By entering all the possible user scenarios (Valid, & Invalid)

* Whenever we are doing testing we should never deviate from our requirement.

* If we have any doubt & Query about the requirement we need to contact Business Analyst.

functional Testing /field level

Requirement

1. Add User:-

1.1 Username :- It should accept 6-32 chars.

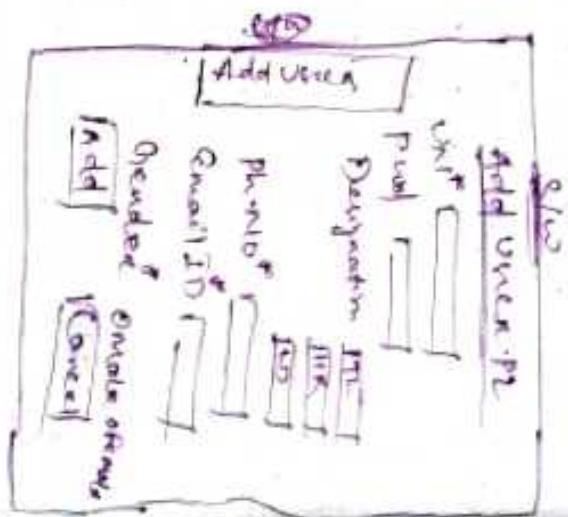
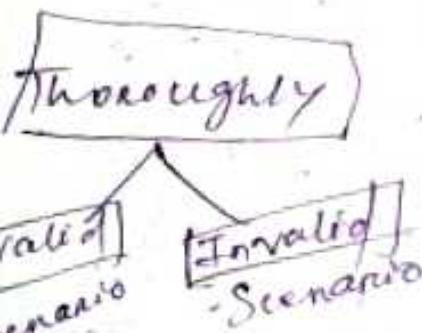
1.2 Password :- It should accept 4-8 chars

1.3 Designation :- " " one value.

1.4 phone Number :- It should accept 10 digit

1.5 Email ID :- " " is Valid Value.

1.6 Gender :- " " is only 2 value



Total ways of Documenting Scenario?

<u>Sl No</u>	<u>Component Name</u>	<u>Input</u>	<u>Output Result</u>
01	Phone no	9249358290 14648...99 abc	Accept Reject Reject

~~HW~~
Assignment - 1

(total 15)

Req (1) Name chars (5-20)

(2) Phone
(10 digit + Country code)

(3) Email ID
Userid @ mail ID
format _____@_____.com

While Each IS Given Requirement Should
be Numbered -

- * If the requirement is Numbered
then it becomes easy to understand.
- * If the req. is Numbered it becomes
easy to Prioritize.
- * It gives a clear clarity in understanding
the requirement.
- * If the number of req? Communication
between developer & Test Engineers

Req (A)

1. Loans

1.1 Personal Loans :-

1.1.1 Mortgage Loans :-

1.1.2 Vehicle Loans :-

Functionality of Testing

Ways

We have 3 diff ways of testing

(1) → Over Testing

(2) → Under Testing

(3) → Optimisation Testing

Ex

Req

1. Amount Transferred

① FAN → It should accept 10 digit A/c no

② TAN → " " " "

③ Amt → It should accept 100-5000

SW

FAN []
TAN []
Amt []

Transferred []

(U) Over Testing / Donkey Testing / Exhausting Testing

Testing the Application with same set of scenarios again & again we call it as Over Testing.

Disadvantages

Waste of time.

Ex FAN []

(10 digits) (✓)

9 digits (X)

8 digit (X) → Same set up scenario

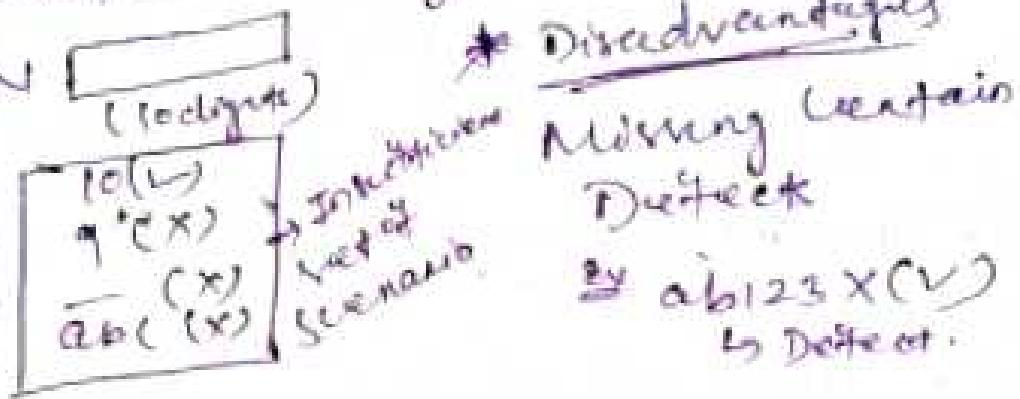
7 digit (X)

6 digits (X)

② Under Testing

Testing with the Application in
current set up scenario we call
it as Under testing.

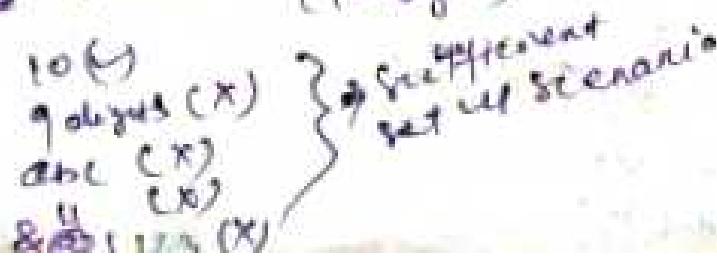
Ex PAN []
(10 digits)



③ Optimization Testing

Testing the Application with Suboptimal
set up scenario, we call it as
Optimization Testing.

Ex PAN []
(10 digits)



Assignment

S/N	Name	Input	Exp-Result
		Component Name	Result
01	BIBWIA		Accept
02	MITU		Reject
03	BISWAJIT123		Accept
04	BISU@5438		Accept
05	LUCKY@OPAL...(21)		Reject
06	@AB@@!		Reject
07	@@@		Reject
08	—		Accept
09	123 SONU&&		Accept
10	# 123 ABC		Accept
11	BISU123%%.9		Accept
12	ITIS @		Accept
13	LUCY @ %		Accept
14	\$\$12		Reject
15	AB CD* ?		Accept

Q) Types of Functionality Testing?

2 types: ① +ve Testing

② -ve Testing

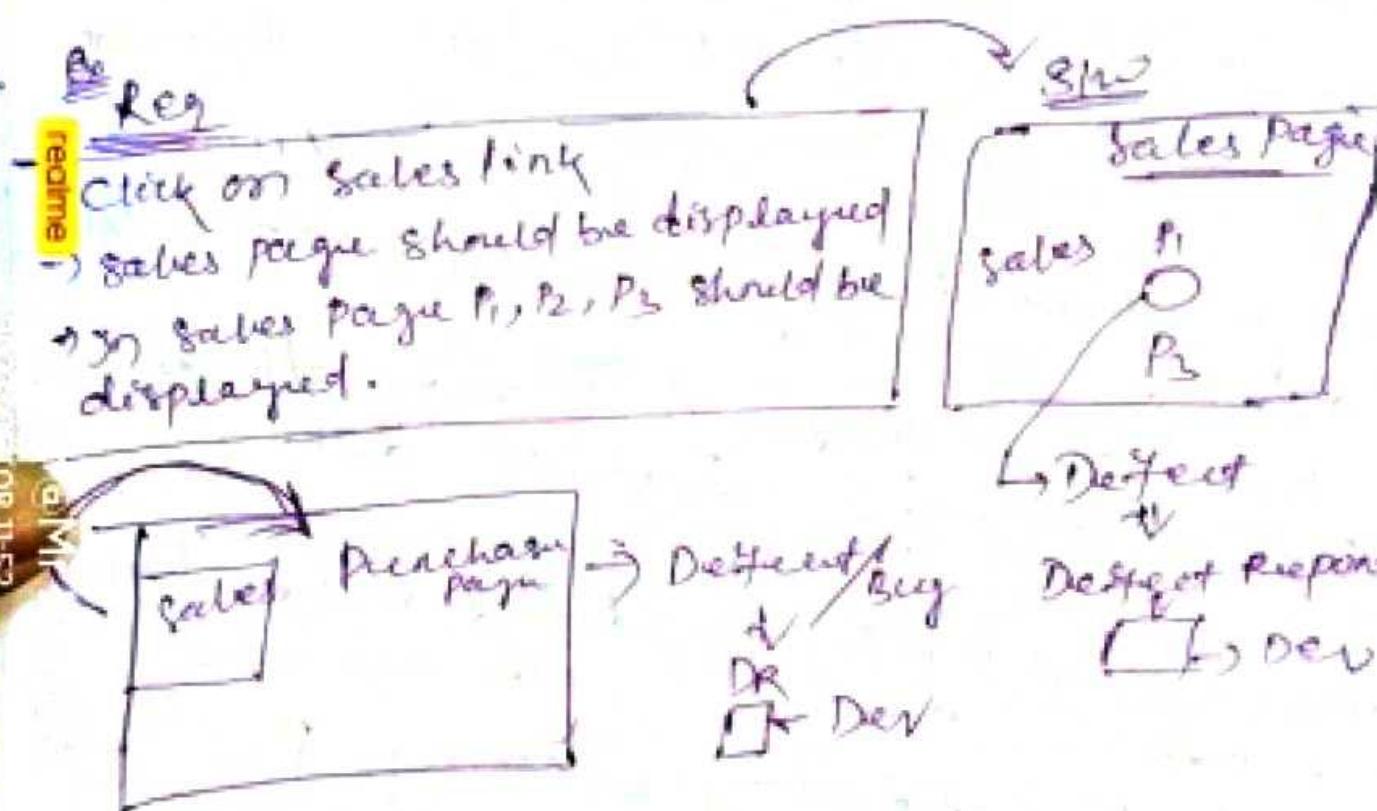
+ve Testing :- Testing Each & Every Component
By Entering the valid Data

-ve Testing :- Testing Each & Every Component
By Entering the Invalid Data

Defect Tracking

Defect \Rightarrow Something which is not working according to the Requirement Specification we call it as Defect.

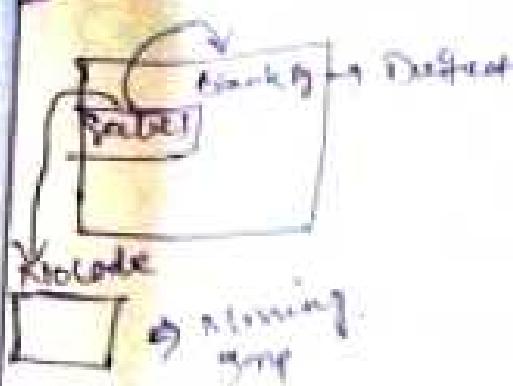
* Derivation from Requirement Specification
we call it as Defect.



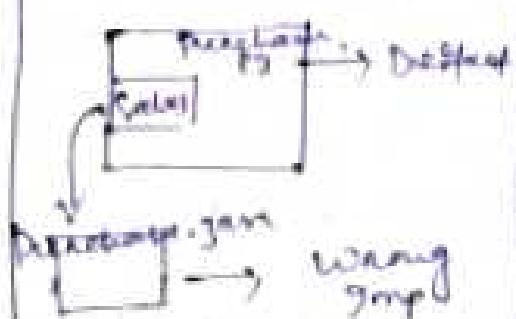
Q Why do we get Defect?

- We get Defect because of 3 reasons - (1) Missing implementation
(2) ^{Wrong} ~~Bad~~ implementation
(3) Extra implementation,

Missing Implementation



Wrong Implementation



Extra Implementation



QUESTION

Diffr. betw. Defect, Bug, Error & Spurious?

Defect

- * Something which is not working according to the requirement specification
- * We call it as Defect?

Bug

- * � formal Name given to the defect we call it as Bug.

Error

- * A mistake done by your programmer in a program
- * We call it as Error
- * We have 2 types of Error
 - (1) Compile Time Error
 - (2) Run Time Error

(1) Compile-Time Error

* A mistake in the Syntax of a Program we call as Compile-Time Error.

Ex Syntax if ($a > b$)

```
{  
    S.o.p ("a is greater");  
}  
else  
{  
    S.o.p ("b is greater");  
}
```

→ ~~Runtime Error~~ → Compile-Time Error.

(2) Run-Time Error

* A mistake in a Logic of Program we call as Run-Time Error.

Ex $a = [6]$

$a = [7 | 8 | 9 | 10 | 11 | 12] \rightarrow$ Logic

$\text{int}[a] = 7, 8, 9, 10, 11, 12, 13$

Fairness: One defect ~~can~~ might lead to Fairness or many defects might lead to Fairness.

recime

ST NO	Component Name	Action Description	To Do	Opp. Result
01	Reprint bottom	To check that when applicable find a user click on Report button. Find a job & now your camera page should be displayed.	Not applicable	Find a job now your camera page should be displayed.
02	Full Name	To check that when click on full name. Name should be displayed	Not applicable	Reprint button accept has been pressed now click on full name. Name should be displayed
03	Email ID	To check that when click on Email ID, write Email ID should be displayed.	Not applicable	Reprint button accept has been pressed now click on Email ID. Email ID should be displayed

JOB of a Development Engineer

- * The Development Eng. Analyse the DR. we will try to fixed the Defect in source code.
- * After fixing the Defect of the Development engineer will send the DR to ^{not} the Test engineer.
- * He will change the status from Assign to Fixed.

JOB of a Test Engineer

- * The Test engineer will analyse the DR and retest the Bug/Defect.
- * If the Defect is really Fixed the TE will change the status fixed to closed.
- * If the defect is not fixed the test engineer will send the DR back to the Development engineer & change the status from fixed to Reopen.

Defect Life Cycle

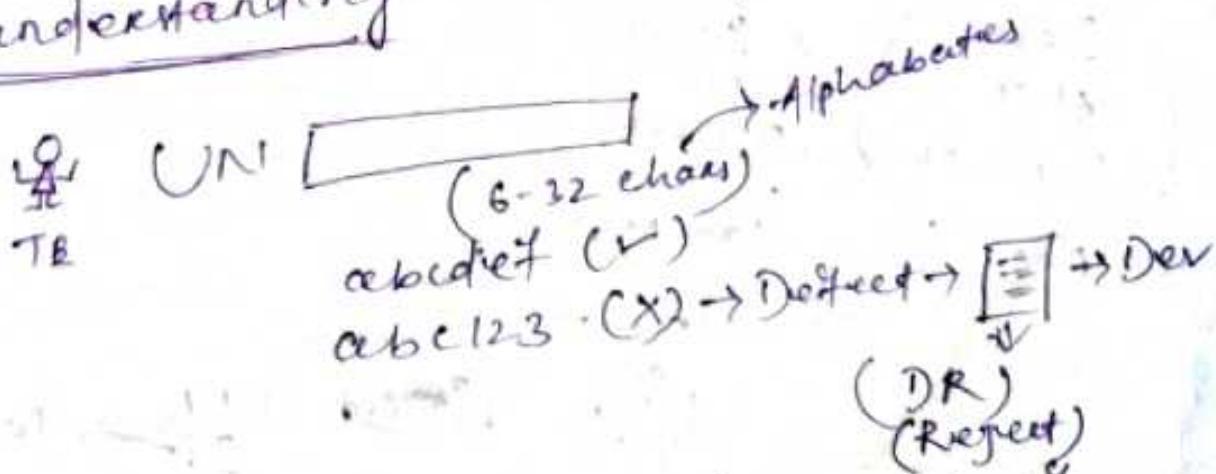
- 1 * Reject
 - 2 * cannot be fixed
 - 3 * RFE/CR (Request for Enhancement)
 - 4 * postponed
 - 5 * duplicate
 - 6 * Not Reproducible

① Reject (Send it to the a feature not rejected)
* Test engineer finds a defect prepared
on defect report and then sends it to the
Development Team.
* Now the Developers will send the status
of Defect Report as Reject.
Q) Why to be get Reject Status? ↗
Ans: It is not rejected because it is not required.

Q) Why to be get rejected

- * We get rejected because of Misunderstanding
- 1. Misunderstanding

1. Misunderstanding



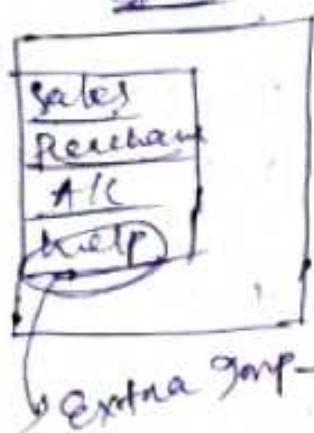
- * We get reject status because of extra-
implementation.

2. Extra Implementation

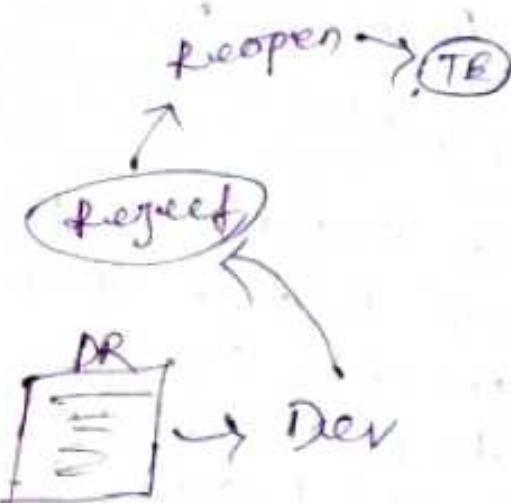
Note

↳ Whenever we get extra-requirements, we call it as defect and the developers will try to reject the defect report. So at that time we need to contact the Business Analyst.

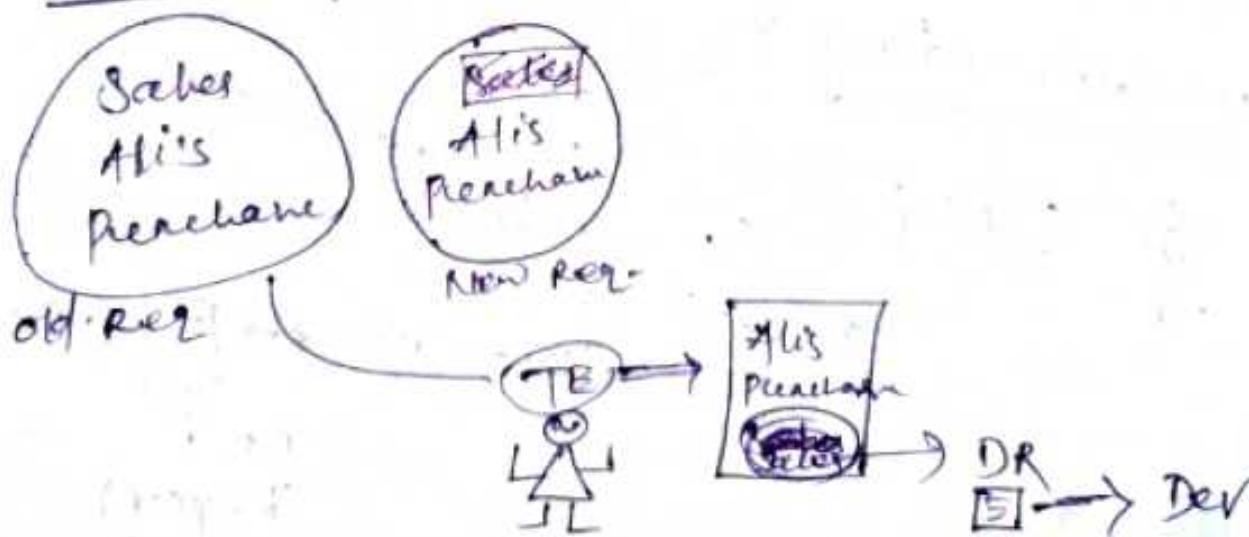
S/W



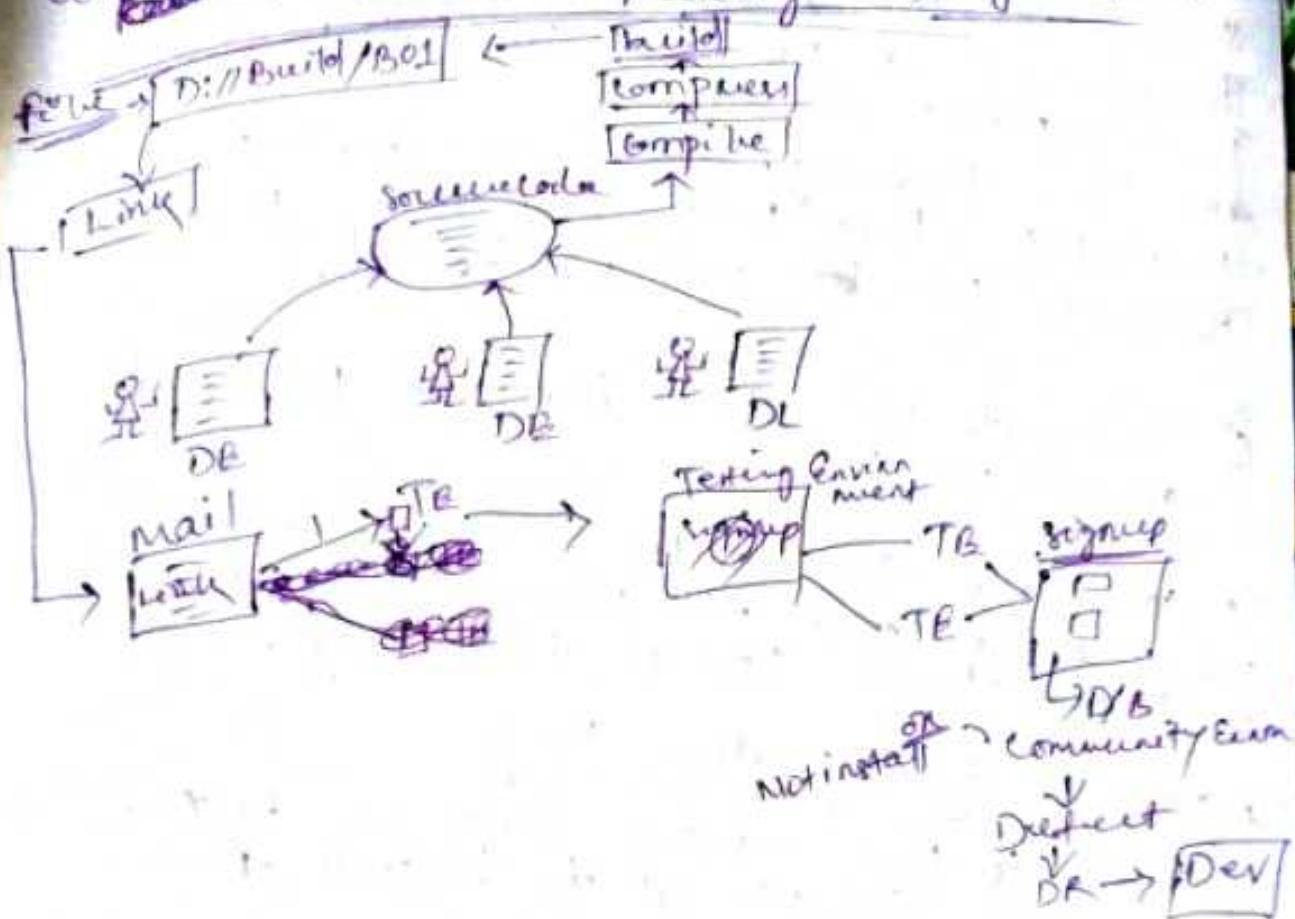
↳ Extra requirement → Defect



3. We get Reject-Header Because of Reverting Old-Requirement.



~~common~~ we get Request Status Because of installations / ~~wrong~~ Configuration



Development Environment

It is a place where we developed a software.

Testing Environment

It is a place where we Test a software.

* Can not be fixed

- ↳ Test Engineer finds a defect and then Prepared a DR and send it to the Development team.
- ↳ The Development team will accept the DR but they say that the defect can ~~not~~ not be fixed.

Q Why do we get Can not be fixed status :-

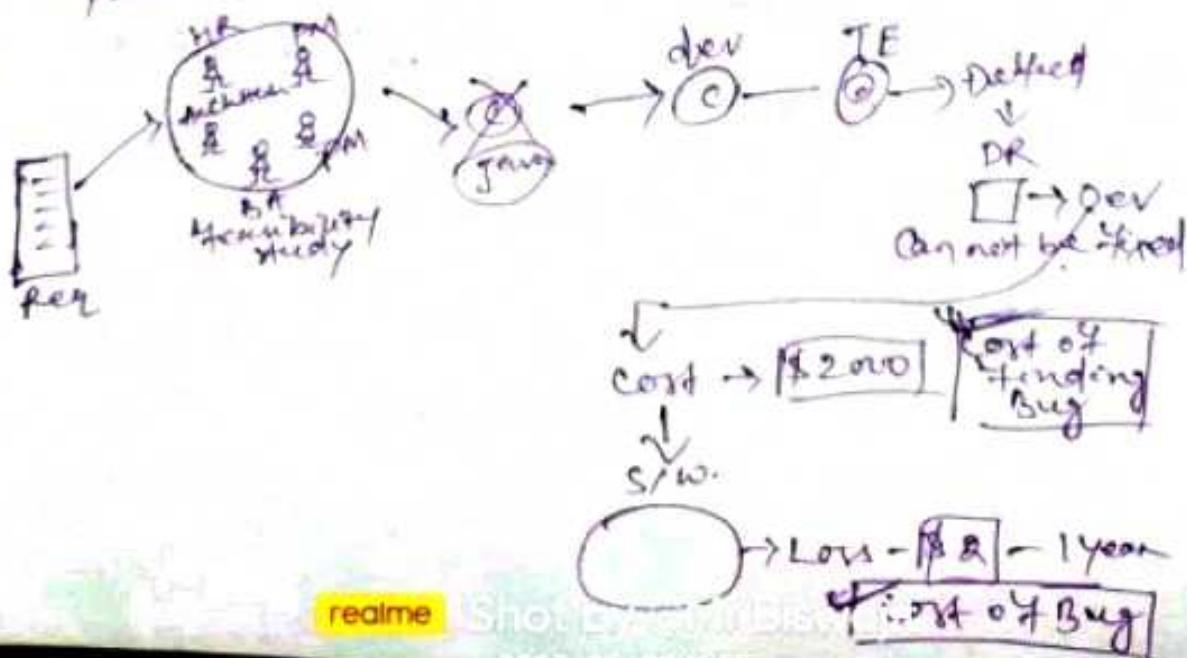
- (1) we get Can not be fixed status whenever cost of fixing the bugs is more than cost of the bug.

(2) Cost of fixing the Bug

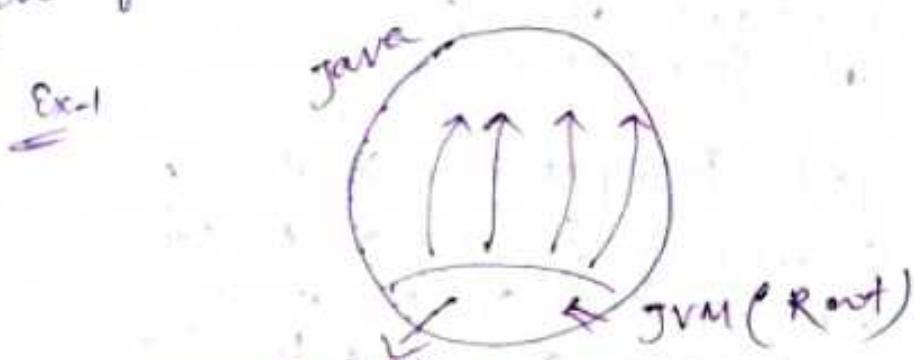
It is the amount charge by the software company to fix the defect.

(3) Cost of Bug :-

It is the loss incurred by the defect in the business.

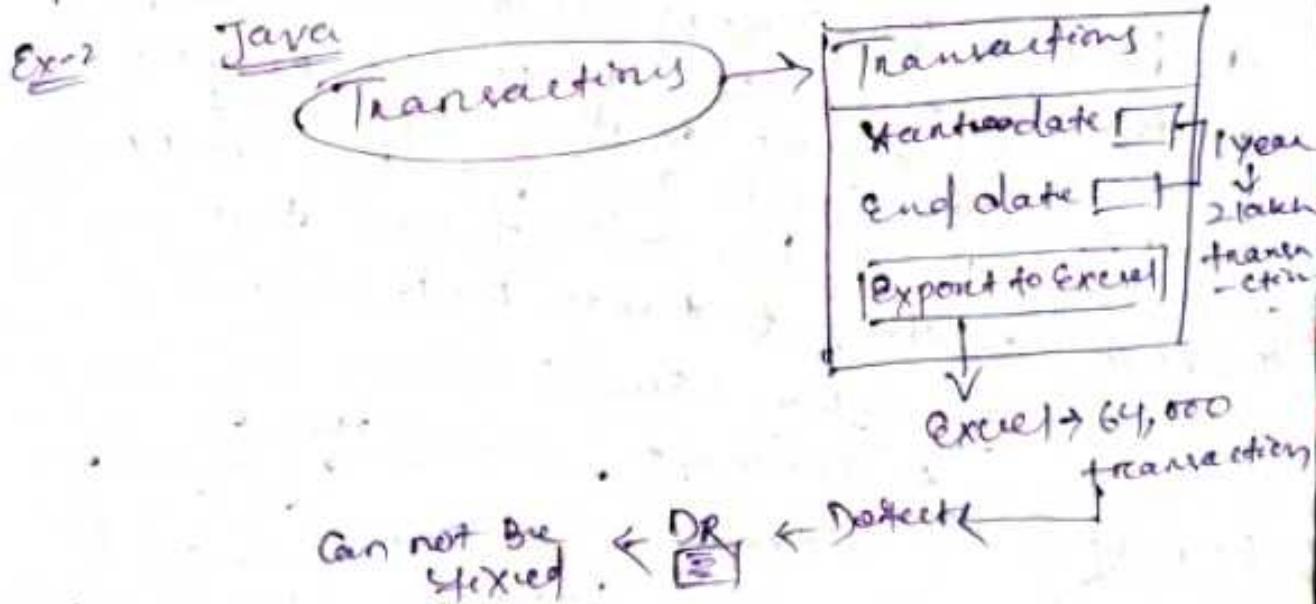


~~Causes~~ ~~break~~ ② By test Engineers finding ~~the~~ ^{a minor} Defect which is present in Rest the software and if it is not affecting customer ~~Break~~, in such a case Developer will change the aspects to can not be fixed.



\downarrow Dark DR \leftarrow Minor
Defect
 \downarrow can not be fixed

③ Because of Technology not supporting
can not be fixed



Postponed Status

What is Postponed Status?

- ↳ Developers are Accepting that it is a defect and they want some time to fix the Defect. In such a case Developers will change the status as postponed.

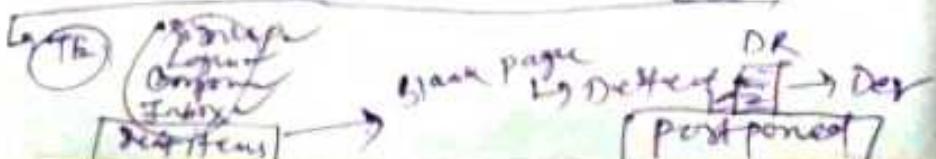
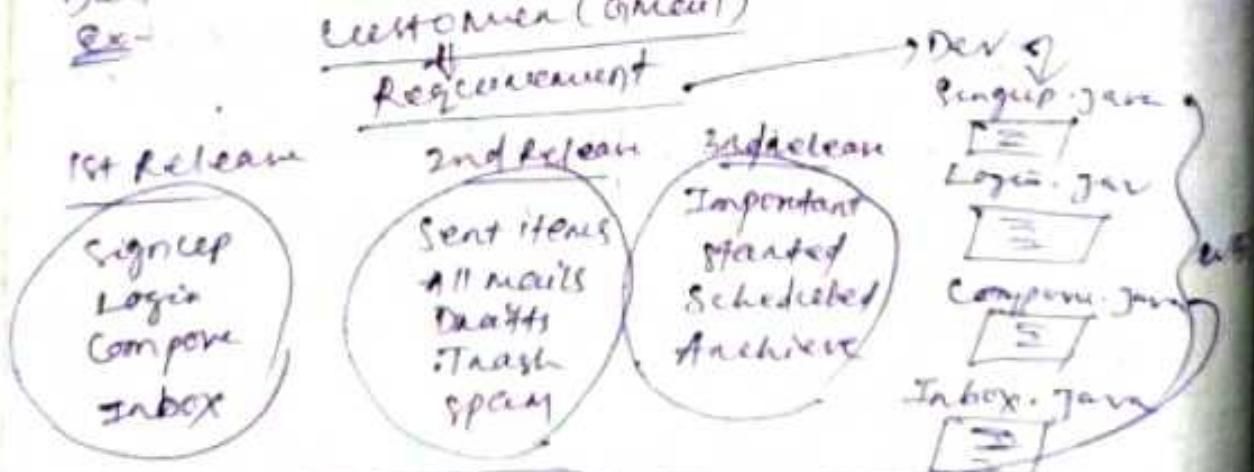
Q Why we get Postponed Status?

- ↳ If test engineer found a minor defect in the end of release and if Developers are not having sufficient time to fix the Defect in such a case Developers will say that we will fix the Defect in future ~~releas~~ release.

- ↳ If test Engineers find the Defect in the feature which is not required to the customer so the current release in such a case Developers will say that we will fix the feature in the future release.

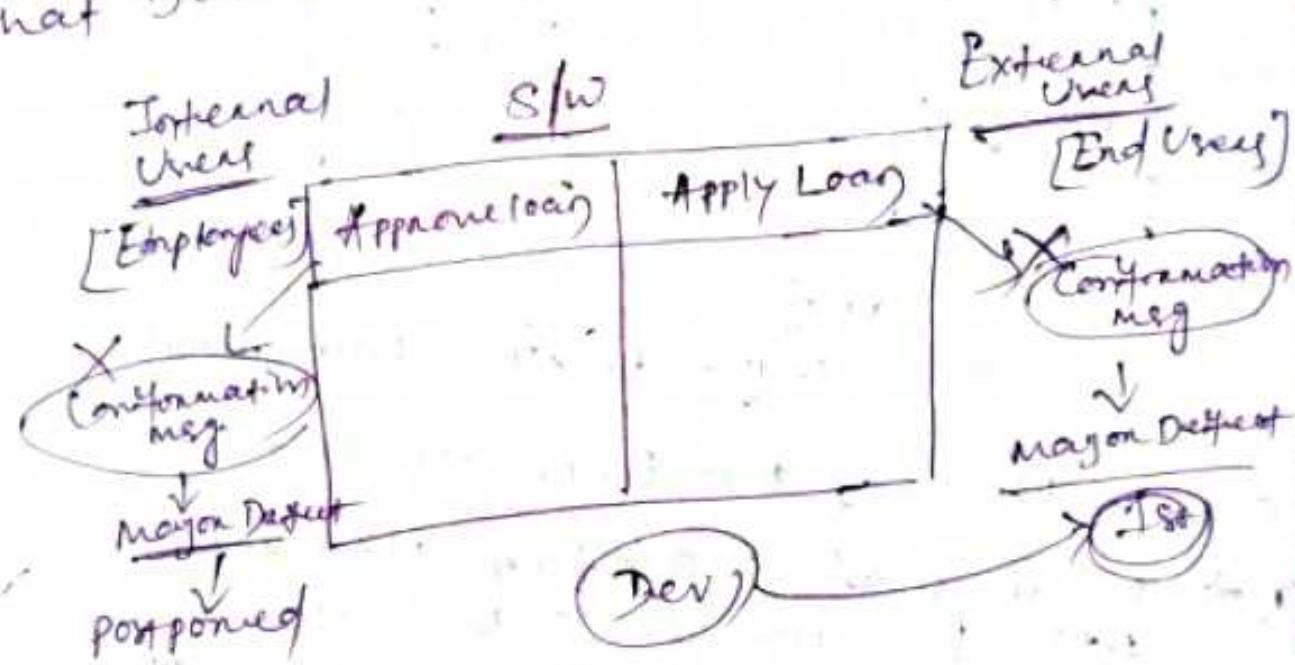
Defect

Ex- Customer (Gmail)



* If test engineer find defects and communicate that to Developers, Developers will say that customer is expecting to do lot of changes in the same fix activities. So this defect will be postponed until we get clarity from the customer.

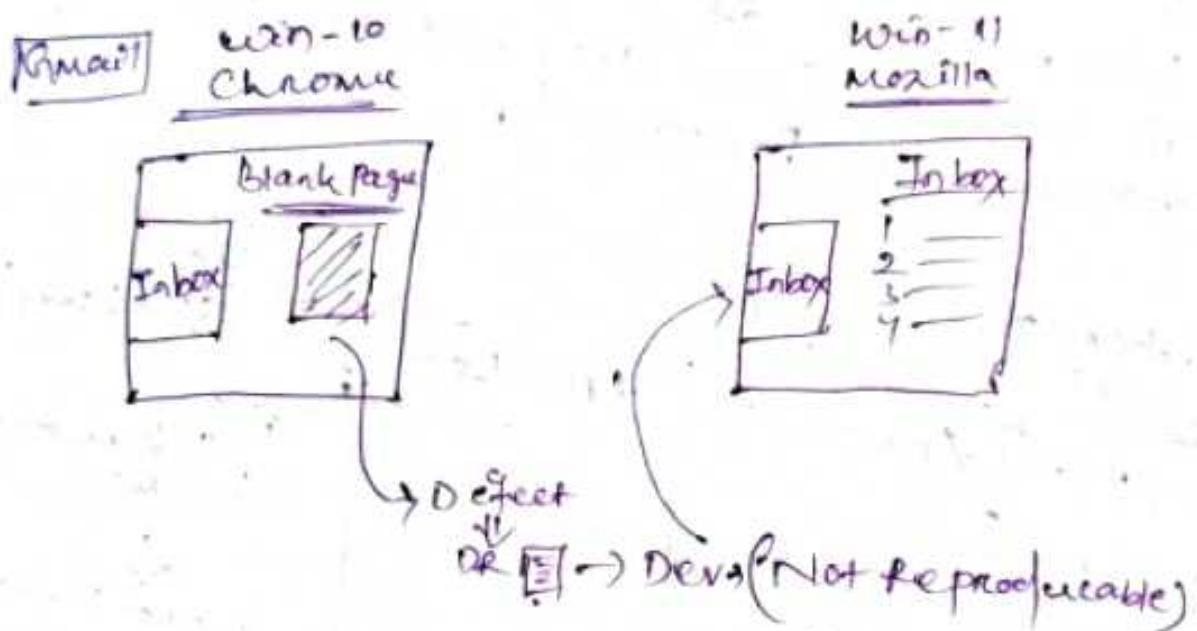
* If there is a defect which is exposed to Internal Users and if it is major/minor defect is such a case Developers will say that Defect is Postponed.



Not Reproducible

What is Not Reproducible Status?

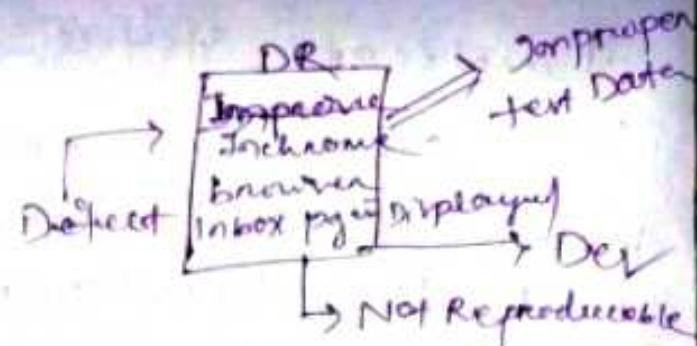
Test Engineers are able to see the defect but Developers are not able to see the same defect. In such a case Developers will change the status as not reproducible.



Q Why we get Not Reproducible Status?

- ① → Because of platform Mismatch
 - 1.1) Because of OS Mismatch
 - 1.2) Because of Browser Mismatch
 - 1.3) Because of Browser Version Mismatch
 - 1.4) Because of Browser Setting Mismatch
- ② Because of Improper Defected Report.

Because of Improper Test Data.



- ③ * We get not Reproducible Because
of Inconsistent Defect.

Q: How do you convince a developer when it is Not Reproducible states?
→ Being a test Engineer we will Prepared a proper defect report which Contains all the platform Details and Proper Test Data.

→ Being a Test Engineer we will take off a screen-shot of the Defect and attached it to the Defect Report so that he can see the prove to the Developer that it exists or not.

→ As a test Engineer, I will try to do a Screen Recording of the Defect & attached it to the Defect Report.

→ Being a Test Engineer if the Developer is not able to reproduce the defect they will try to take access of the Developer's **realme** **computer** with the help of **Mr Biswajit**.

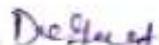
Application like - AnyDesk, Team viewer.

Deuplicate Stories

Test Engineers find a defect prepared a defect report and sends it to the Developer but the Developer says that other some DR is sent by some other test engineer then we get stories called as Deuplicate ~~stories~~.

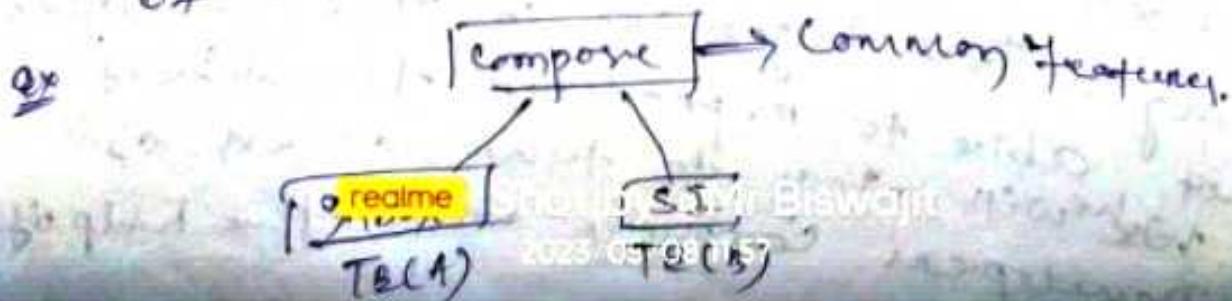
Q. Why we get Deuplicate Stories?

* ① We get Deuplicate Stories because some other Engineer will send one Defect Report to the Development team.

Ex: TB(A) → Compose → Login: 10:30 →  (11:00) - DR 

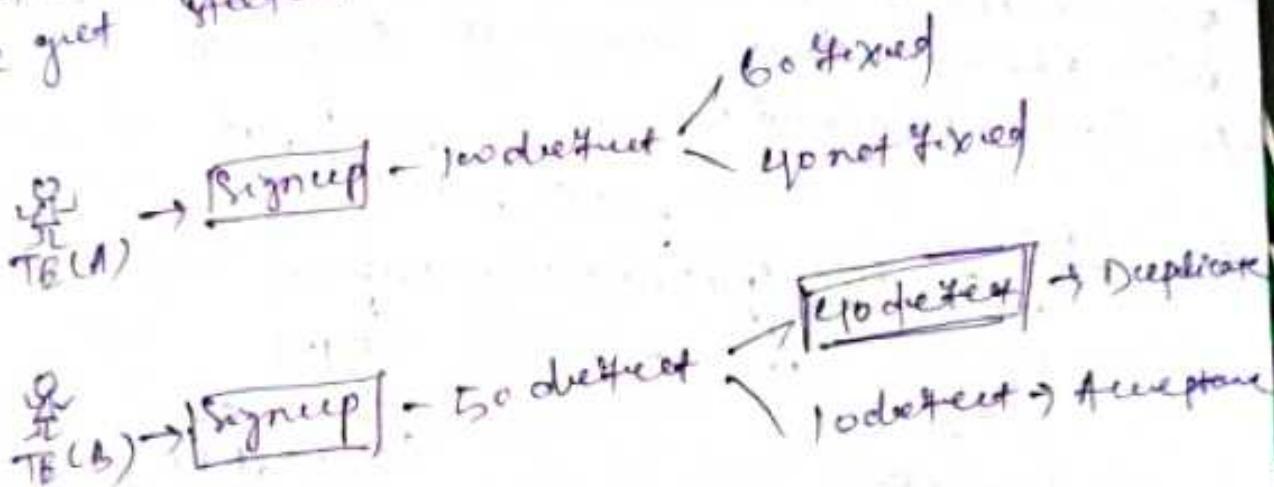
Ex: TB(A) → gnbbox → Login: 11:00 →  (11:30) - DR 

* ② We get Deuplicate Stories because of common feature.



* ③ We get Duplicate Status Because old test engineer has send the Defect report to the Development team and if a new engineer sends the same if a new engineer sends the same to the Developers then it is called as :- Duplicate.

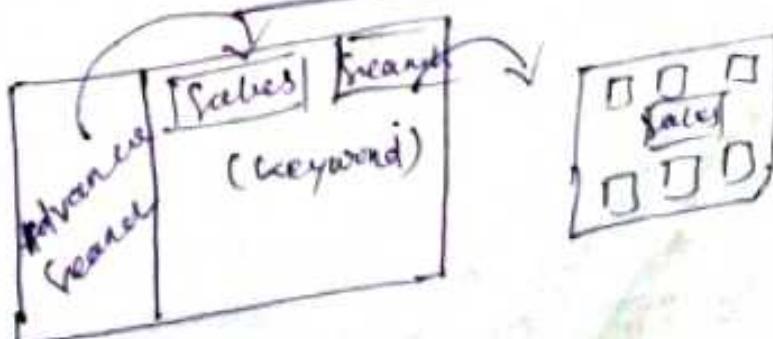
We get status



How do we avoid duplicate status?

- We can + avoid duplicate status with the help of Advance search.
- In Advance search we will enter the key words and check whether do we have a defect report with ~~not~~ that keywords.

DTT (Defect tracking tool)



RFB / CR

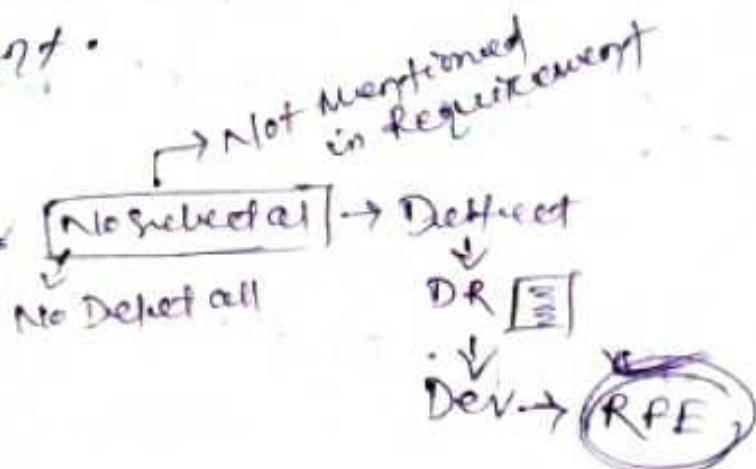
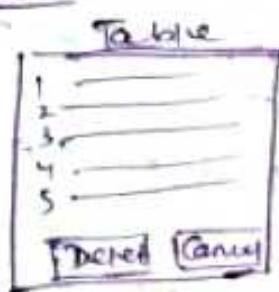
10/11/22

(Request for Enhancement)

Q) What is RFB Status?

* While testing the software finds any defects which is not part of the requirement is such a case Developers will change the status as Request for Enhancement.

Ex- Gmail



Q) Who can keep Request for Enhance status?

- * It can be given by -
 - Developer
 - Test Engineer
 - Manager
 - Business Analyst
 - Customer

DEFECT REPORT / DBFBCT REPORT TEMPLATE / Explain the Attributes of a Defect Report

DBRRMBS/SPR

1. DEFECT ID :-
2. BUILD NAME :- Build Gmail
3. RELEASE NAME :- Tiger
4. REQUIREMENT NO :- 302
5. MODULE NAME :- Compose
6. BUILD ID :- B03
7. STATUS :- New/Open/Assigned/Fixed/Rejected/Deuplicate
Can not be fixed / Postponed / Not Reproducible
/ RFB / Re-open / Closed
8. SEVERITY :- Blocker/Critical Defect/Major/Minor
9. PRIORITY :- High/Medium/Low
10. REPORTER :-
11. ASSIGN TO :- DBV
12. TEST CASE NAME :- Gmail Compose Scenarios
13. TEST DATA :- UN, PWD
14. TEST URL :- <https://Gmail.com>
15. DEFECT SUMMARY :- When user click on Compose,
Blank page is displayed.
16. BRIEF DESCRIPTION :-
OS :- Win-10
Browser :- Chrome
Browser Version :- 106
Steps to Reproduce Defect :-
 - ① Open the browser
 - Enter the URL & then click on Ok
 - ② login to gmail app
 - ③ Click on Compose link

17. EXPECTED RESULT: → When user clicks on Compose, Compose page should be displayed.

18. ACTUAL RESULT: → It displays a blank page.

19. Attach SCREENSHOT: →

_____ X _____ X _____

Q2 What is Defect Tracking tool?

It is a tool which is mainly used to store & track the defect in a centralized place.

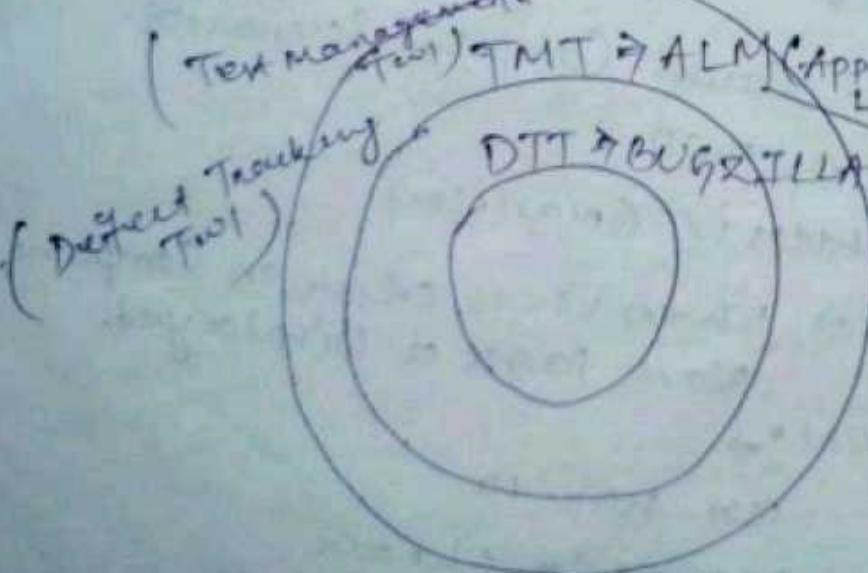
and communicate the defect to the developer in a organized way. (Project management tool)

PMT → JIRA → [Store Requirement
Write Test Case
Track Defect
Lifecycle management]

(Test management) TMT → ALM Application

→ Track Defect
Lifecycle management

→ Test Case
write



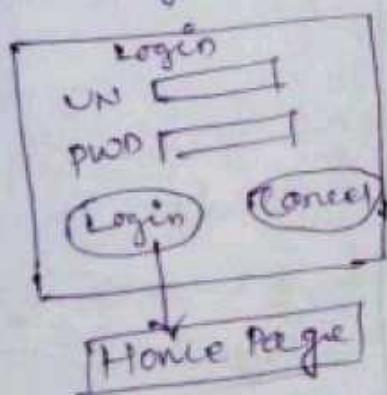
Name Some Defect - Tracking Tool

- 1) BUGZILLA → PQAe . D.T.T
- 2) JIRA → PMT
- 3) MANTIS
- 4) BUG PRO
- 5) BUG NBT
- 6) BUGZINI
- 7) RATIONAL CLEAR QUEST
- 8) CROC PLUS

~~step~~ & How does the Defect tracking tool works?
(Bugzilla)

Step-1:

Open the Browser , Enter the URL
<https://bugzilla.com> & then Click on OK .



* Test Engineer will Search for Duplicate Defect by Entering Defect-ID in Search text field. suppose if he don't know defect ID , he can search Duplicate Defect with the help of Advance search feature.

Step 2 to Step 3

Defect Report

→ Defect ID :-
Build Name :-
Release Name :-
Ref. No :-
-
-
-
-
-
-
[submit] [Cancel] [Reset]

↳ Unique Defect ID

write in Step 2

- * If there is no duplicate Defect test engineer
Create a new Defect Report.

Severity

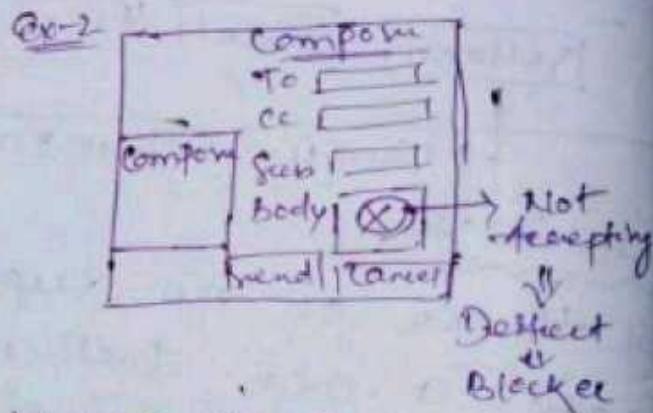
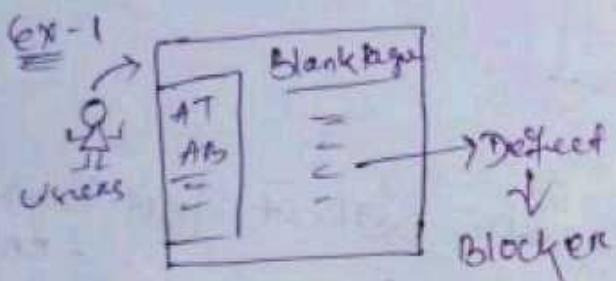
- * It is decided based on the impact of defect on the customer business.
- * There are 4 levels of Severity
 - 1) Blocker Defect / Show Stopper Defect
 - 2) Critical Defect
 - 3) Major Defect
 - 4) Minor Defect / Trivial Defect

12/11/22

Blocker Defect \Rightarrow

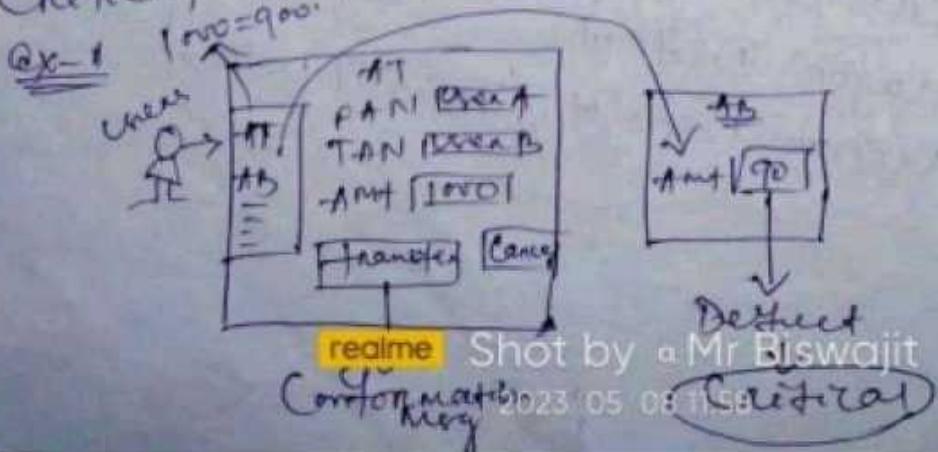
Assume that There is a Defect in a Application & It is 100%. Since that this defect will affect Customer Business Work Flow and also It is Blocking Test Engineer to test the features, such kind of Defects are called as Blocker Defect.

* This Defect will affect other



Critical Defect \Rightarrow

Assume that there is a Defect in the Application and It can 100% see that. This Defect will affect the Customer Business Work Flow But It is not Blocking Test-Engineer to Test the features. It means still we can continue the testing. This kind of Defects are called as Critical Defect.



Ex-2 Whenever User Recharges for 100/- we should get full talktime of Rs 100.

S/W

Mobile	9249358290
Amt	100
Recharge Plan	

Recharge is Submitted
for Rs 101 on Date

Defect → Critical

Major Defect :

Assume that there is a defect in the Application and you can not see that how this defect is going to affect the customer business work-flow. This kind of defects are known as Major-Defect.

Ex-1

S/W

Mobile	9249358290
Amt	100
Recharge Plan	

X Confirmation msg

↓
Defect

↓
Major

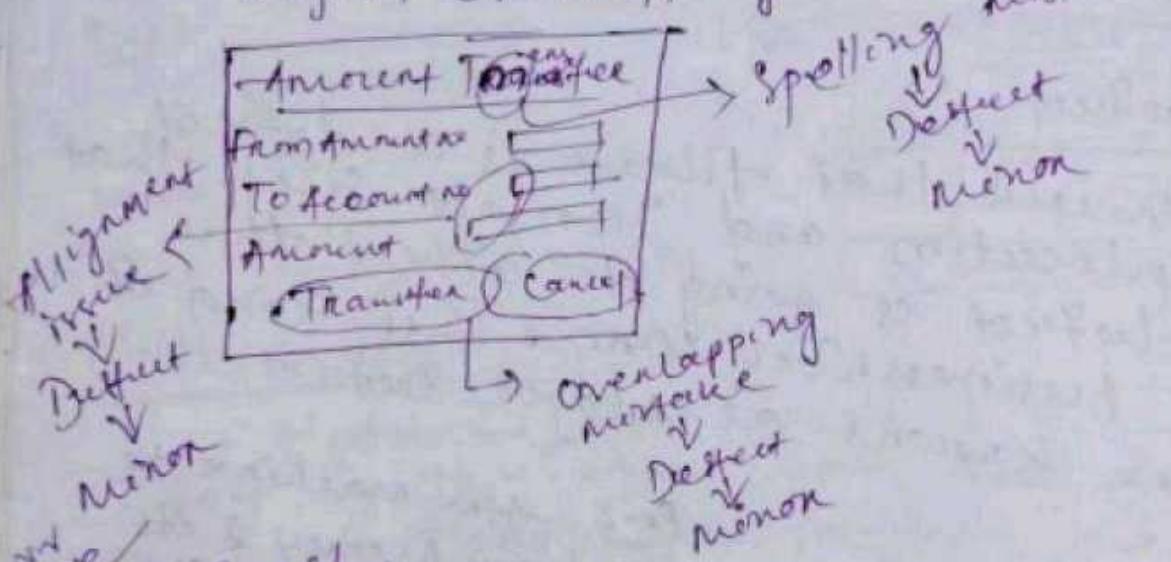
Ex-2 ATM Machine to withdraw Money & its not giving you benefits. it is major defect.

Ex: Facebook Signup page remember password feature is not working.

Minor Defect (Trivial Defect)

→ Assume that there is a defect in the application. It may seem that this defect will not affect the customer business. Work-flow this kind of defect are called as Minor Defect.

Ex-1 Spelling mistake, Alignment issue, Object Overlapping



Priority

14/11/22

→ It is the important to go to fix the defect.

→ How soon the defect must be fixed by the developer

→ There are 3 levels of Priority

① High Priority (P₁)

② Medium Priority (P₂)

③ Low Priority (P₃)

High Priority (P₁) :

if a defect is having Priority as P₁, then Developer should fix the defect immediately.

Medium Priority (P₂) :

if a defect is having Priority as P₂, then Developer can fix the defect within some test-cycles / within some builds / within release.

Low Priority (P₃) :

if a defect is having Priority as P₃ then Developer can fix the defect in next upcoming release or within some two to three releases.

Ex) In whatsapp user is not able to do video call ? severity (Blocker)
priority (P₁)

Q&A) In Amazon In the payments page there is a spelling mistake at the bottom.

↳ Severity (Minor Defect)

Priority (P₃)

realme

Shot by Mr Biswajit

2023 05 08 11:58

↳ low severity
low priority

④ In OLA Users is able to book or ride and finished the ride by entering wrong OTP?
↳ Severity (Critical Defect)
Priority (P1)

⑤ In what app Meeter Functionality is not working?
↳ Severity (Major Defect)
Priority (P2)

⑥ In youtube Users is not able to play the video?
↳ Severity (Major)
Priority (P1)

⑦ In youtube if user is subscribed then it is showing 1500 subscribed?
↳ Severity (Critical Defect)
Priority (P1)

⑧ In facebook Login Page, Remember Password functionality is not working?
↳ Severity (Major)
Priority (P2) (P1)

Note: If test engineer finds any defect which is directly visible to the User they that **realme** Shot by Mr Biswajit will take highest Priority (P1)

⑤ In Flipkart while doing payment user is not able to save the Credit Card details. Severity (Major): Priority (P2)

Combination

1. High Severity [Blocker/critical] → High Priority
2. Low Severity [Minor] → Low Priority
3. High Severity → Low Priority
4. Low Severity → High Priority

High Severity → Low Priority.

Ex- ① Gmail → Help → Blank Pg → Deferred
S! → Blocker
P! → P3

② WhatsApp → *not able to install soft time* → Deferred
S! → Blocker
P: P3

③ Make Mytrip.com

Chrome
Mozilla
IE → 1st error
Opera
Safari

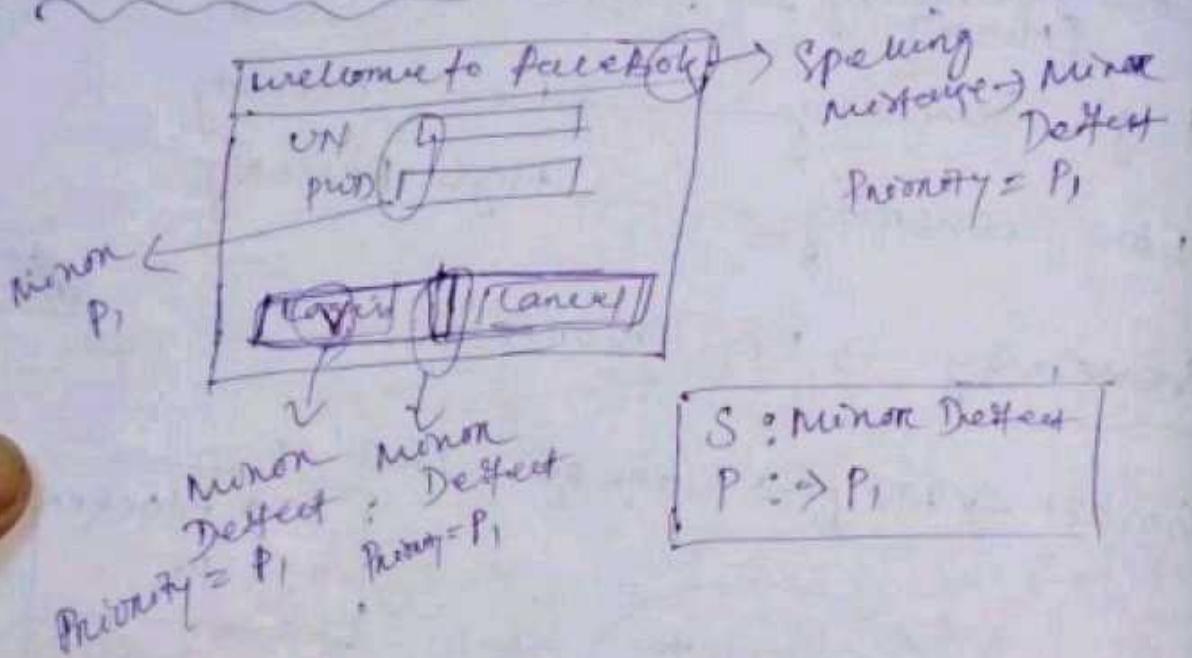
IE
From: _____
To: _____
Search: _____

Blank Pg → Deferred / for IE
S! → Blocker
P: P3
Arrived Departure cont

Note

If Test Engineer finds any Blocker or Critical Defects in the application which is cured by less number of cases then that defect is always take lowest priority (P_3).

Low severity \rightarrow High Priority



— x — ^{END} Interview Question — x — END — x —

- ① Who will be give Severity & Priority?
Ans \hookrightarrow Test engineer.

group NOTE

Developers will always fix the defect 1st which is having highest priority other than highest severity.

Defect 1
S : Blocker
P : P_2

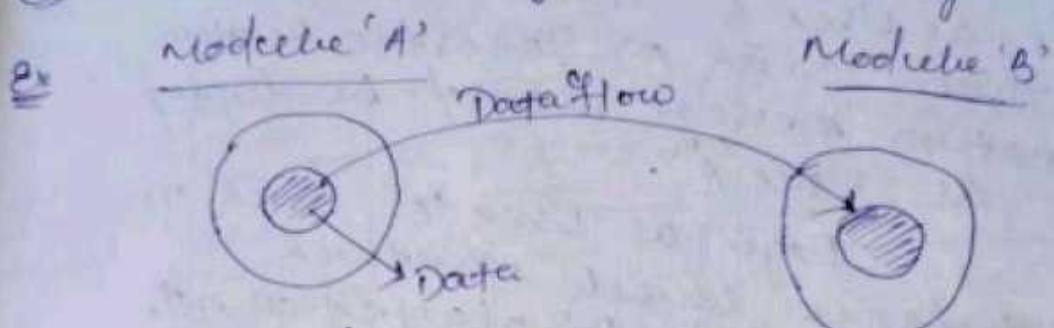
Defect 2
S : Minor
P : P_1

fixed
1st in Dev

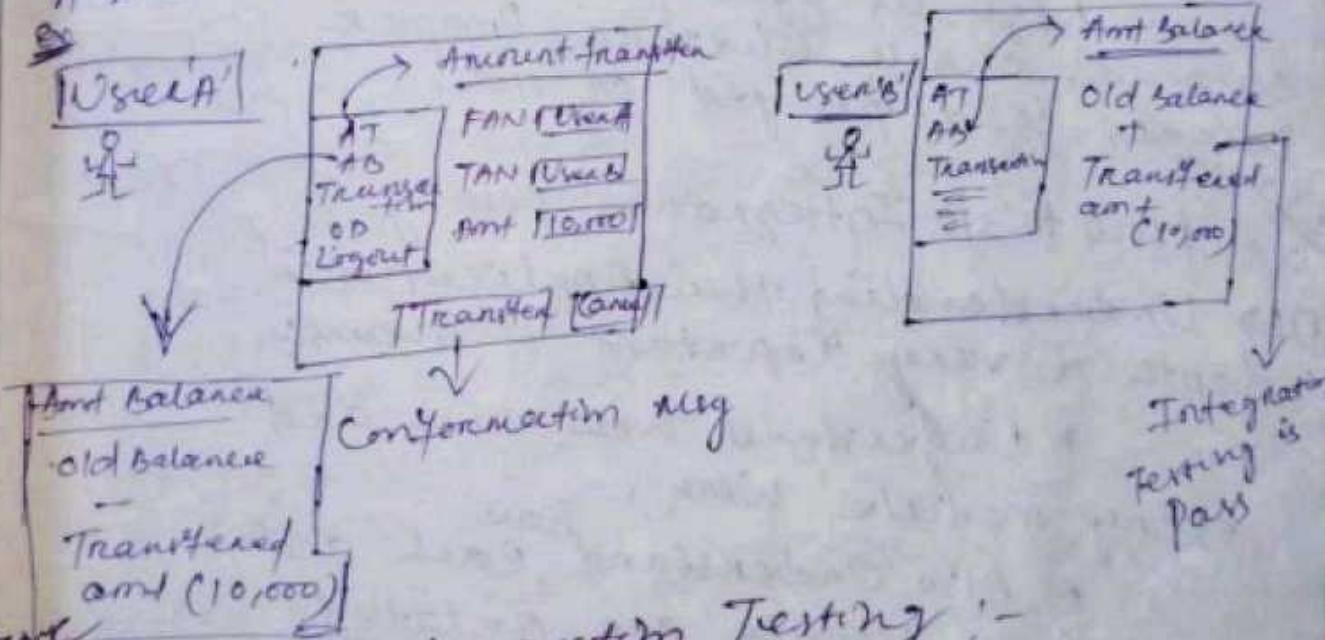
realme Shot by Mr Biswajit

16/11/22

- ② Integration Testing \Rightarrow (Interface Testing)
- * Testing the Dataflow between the Modules is called as Integration Testing.



- * If module 'A' is able to send the data and if module 'B' is able to receive the data. Then Integration Testing betⁿ the Modules A & B is Passed.



Ways of Integration Testing :-

3 types

- ① 2 way IT : \Rightarrow There is dataflow betⁿ Module A to module-B and module-B to Module A i.e. 2 way IT.
- ② 1 way IT : \Rightarrow There is dataflow betⁿ Module A to Module B or Module B to Module A i.e. 1 way IT.
- ③ 0 way IT : \Rightarrow There is no data flow betⁿ Module A to Module B or Module B to Module A i.e. 0 way IT.

Logout as User 'A'.
Login as User 'B' click on
Amount Balance check whether
Proper Balance is displayed or not.

Scenario-2

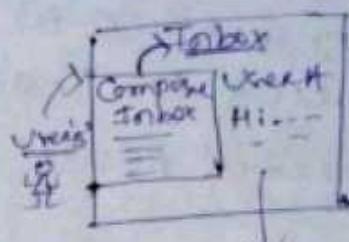
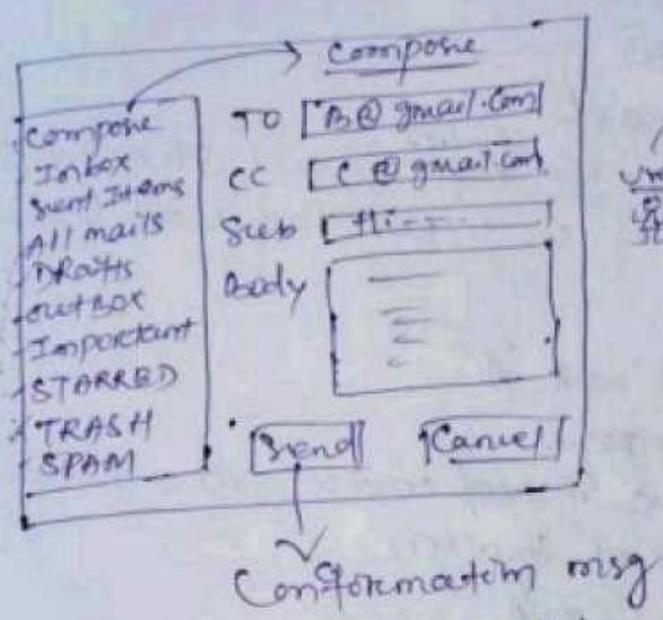
Login as User 'A' click on Amount
balance, check whether Proper
Balance is displayed or not.

- ~~How to do Integration Testing?~~
- ① ↳ Understanding the Application in depth is very important it means:-
 - * Understand how each and every module work.
 - * Also understand how each and every module are related.
 - ② ↳ Identify all possible scenarios.
 - ③ ↳ Prioritize the scenarios
 - ④ ↳ Document the scenarios according the priority.

- ⑤ ↳ Execute the Scenario
- ⑥ ↳ While executing if you find any defects communicate it to Developers.

 Ex

↳
User 'A'



Integration
Testing & Pass

Integration Scenario - 1

1. a) Login as User A
- b) Click on Compose & fill all the field with valid data
- c) Click on send button
- d) Click on logout as User A
- e) Login as User B
- f) Click on inbox

Expected Result :> Sent mail should be displayed in inbox page of user B's account.

 10 scenario

TEST CASE :-

<u>Step No</u>	<u>Description</u>	<u>Tested</u>	<u>Expected Result</u>
1.	Open the Browser & Enter the URL & Click on Go	www.gmail.com	Login Page should be displayed
2.	Login as User & work Valid UN & PWD	UN = Vireen PWD = # @123	Home page should be displayed
3.	Click on Inbox link Select a mail & Delete it.	NA	Inbox page should be displayed
4.	Click on Trash link	NA	Selected mail should be deleted.
5.	Click on Logout link	NO	Deleted mail should be displayed in Trash
			>User A should be logged out from the app successfully

V.V
Q3 Qmp

What is the diff b/w test scenario & Test Case?

Test Scenario

- * It is a high level document of all the ~~Business~~ Business Work Flow according to the requirement.
- * We can conceive test scenario by looking to Requirement.
- * By looking into test scenario we can't test any Application until we have very good product knowledge.
- * Test Scenario tells what to test.

Test Case

- * It is a detail Document of the scenario which helps test the Application.
- * We can conceive Test Case by looking to scenario & Requirements.
- * By looking into Test Case, we can Test any Applications no matter whether we have product knowledge or not.
- * Test Case tells How to test.

Ex-3

Admin

IT

Add User	
Add user	
Edit user	
List user	
Search user	
Delete user	
Logout	realme

Shot by Mr Biswajit

2023 05 08 11:59



Types of Integration Testing

There are 2 types of Integration

Testing : are :-

- ① Top-Down IIT
 - ② Bottom up IIT
- (1) Incremental Integration Testing
- (2) Non-Incremental Integration Testing

1) Incremental I Testing

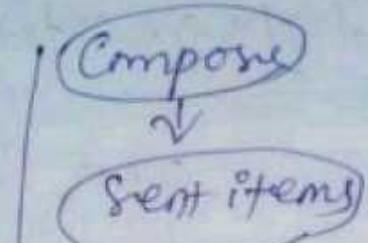
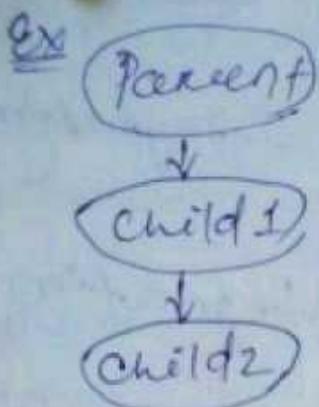
Incrementally adding the module and testing the data-flow b/w the Modules is called as Incremental Integration Testing.

ex

- Ⓐ sign up
- Ⓑ Login
- Ⓒ compose
- Ⓓ ~~comment~~ items

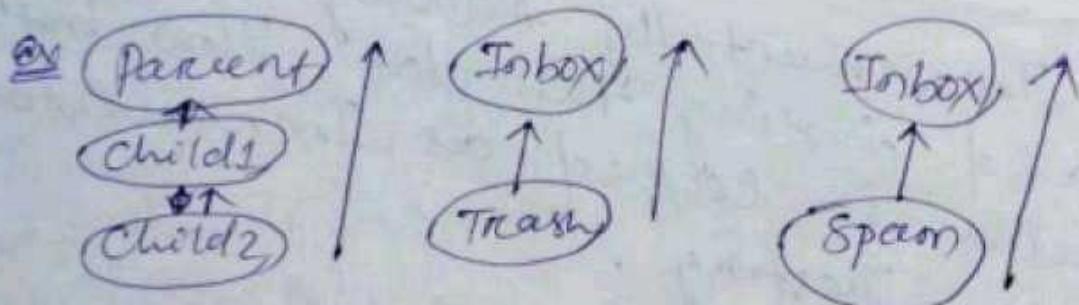
(i) Top - Down - IIT ↗

Incrementally adding the module and testing the Data-Flow between the Modules but make sure that the module which we are adding is child of previous module.



ii) Bottom-up FIT \Rightarrow

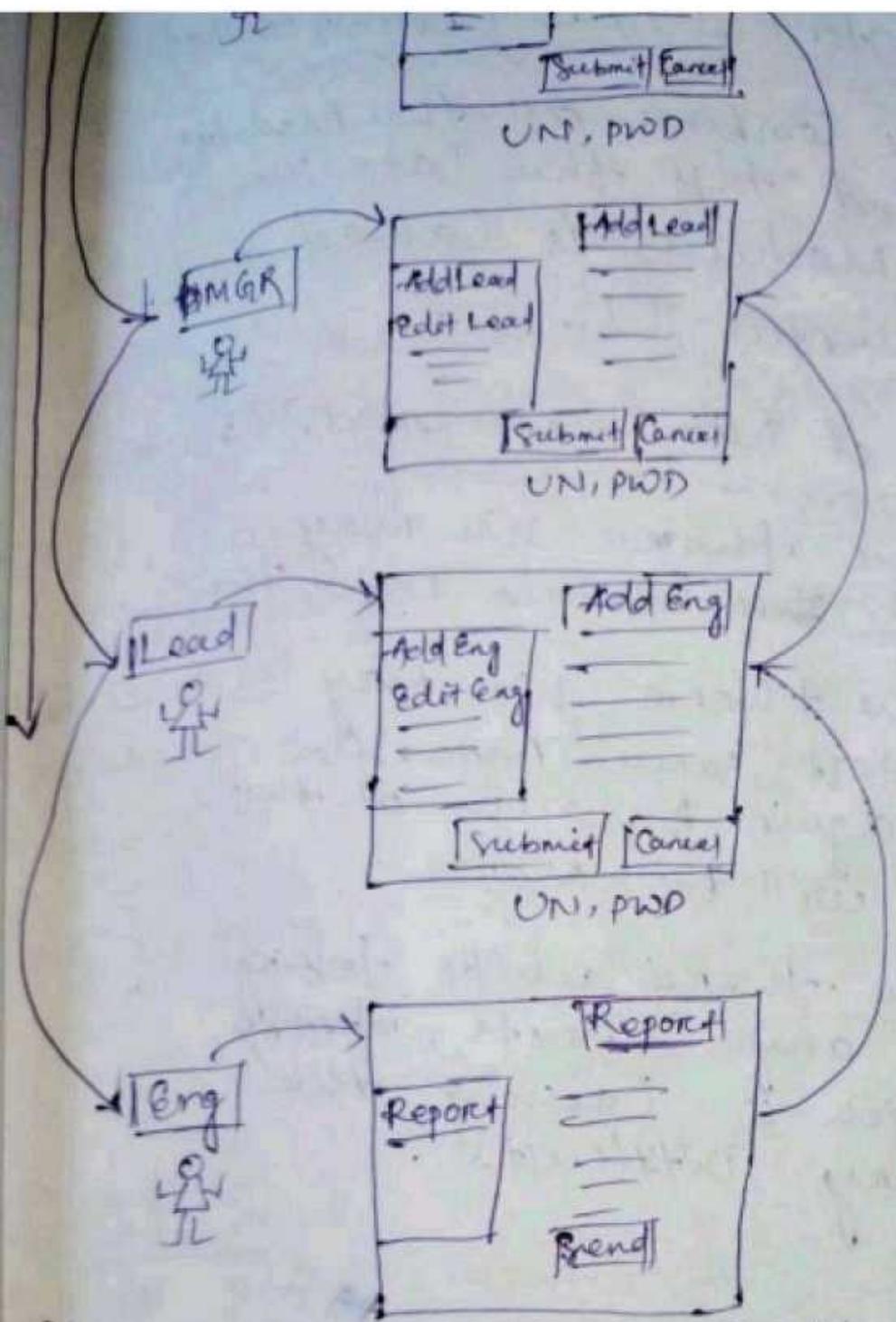
Incrementally adding the module and testing the Data - flow between the modules But make sure that the module which we are adding parent of previous module.



Sandwich Testing

It is the combination of Top-Down IT & Bottom-up IT Testing.





Bottom-up Approach

Note Q) why do we go for Non-Incremental IT ?
Ans
> When we don't know which is Parent & child Module or when it is difficult to identify which is Parent and child module or when the Data flow is very complex, we don't want to waste time in identifying which is parent & child module.

→ It is better to go for Non-Incremental Testing.

② Non-Incremental IT / BIG-BANG Method

* Here we combine all the Modules in one shot and test the Data-Flow between the Modules is called as - Non-Incremental IT.

* Drawbacks of Non-Incremental IT?

→ Chances are there we may miss to test some certain Data-Flow.

→ Chances are there we may repeat to test same Data-Flow again and again. Because of this time taken will be more.

→ Chances are there while testing if you find any Defects, identifying the root which is causing the Defect is very difficult.

(End)

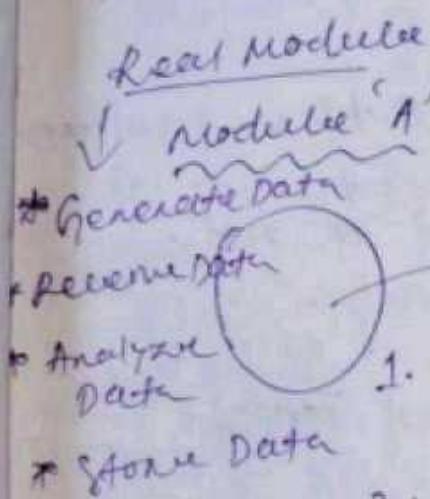
Interview Question

19/11/12

Q Assume that there are 2 modules A & B. Module A is Ready and module B is not ready. In such a case can you do Integration Testing.

Q What is STUB and Driver?

A Yes, we can do Integration Testing by introducing Stub & Driver.



Dummy Module

Module B

STUB

* Generate Data
* Revenue Data

Driver

1. Set up the environment

2. Does the transaction
3. Analyze the Data & Send the Report.

STUB

* It is a Dummy Module which acts like a Real Module.

It can only generate Data & Revenue the Data.

Driver

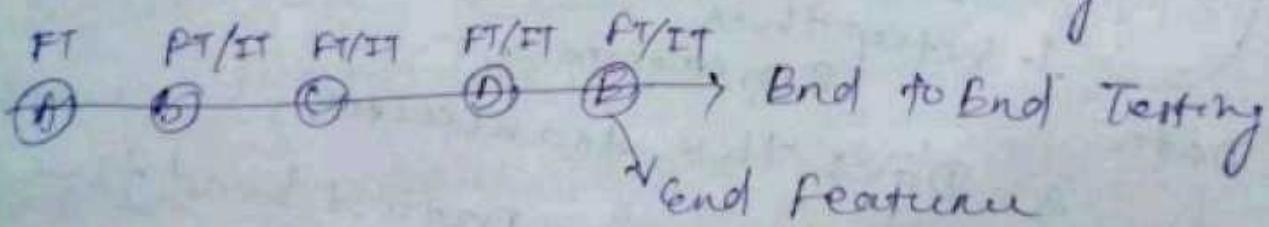
It will set up the environment. does the transaction betⁿ the Real module & the Dummy Module. and Analyze the Data then sends the Report.

SYSTEM TESTING

It is an end to end testing where test environment is similar to Production Environment.

End to End testing :-

~~and to end~~ Navigating throw all the feature and testing whether End Features is working as per the Expected or not. is called as End to End Testing.



Example

Requirement :-

- if any customer applies OD for 1st After 1 month it should charge 2% RI as Activation fees.

- if some customer applies OD for 2nd after 1 month it should cha

~~Definition~~ Considering all the Real-time End-to-End Business Scenario's & testing whether software is ~~able~~ capable ~~to~~ handling it or not is called End-to-End Testing.

Ex-2 Discrepancy

1. New Customer → > 50,000
→ 2% RI
→ 250 Rs AF

2. 2nd time same customer → Any amount
→ 2% RI
→ NO AF

3. 3rd time same customer → Any amount
→ 2% RI
→ Refund AF

Development Server / Development Environment

- It is a set up which is used for developing the Software.
- It contains Software, Hardware, Network & Server.

Re

Test Server / Test Environment

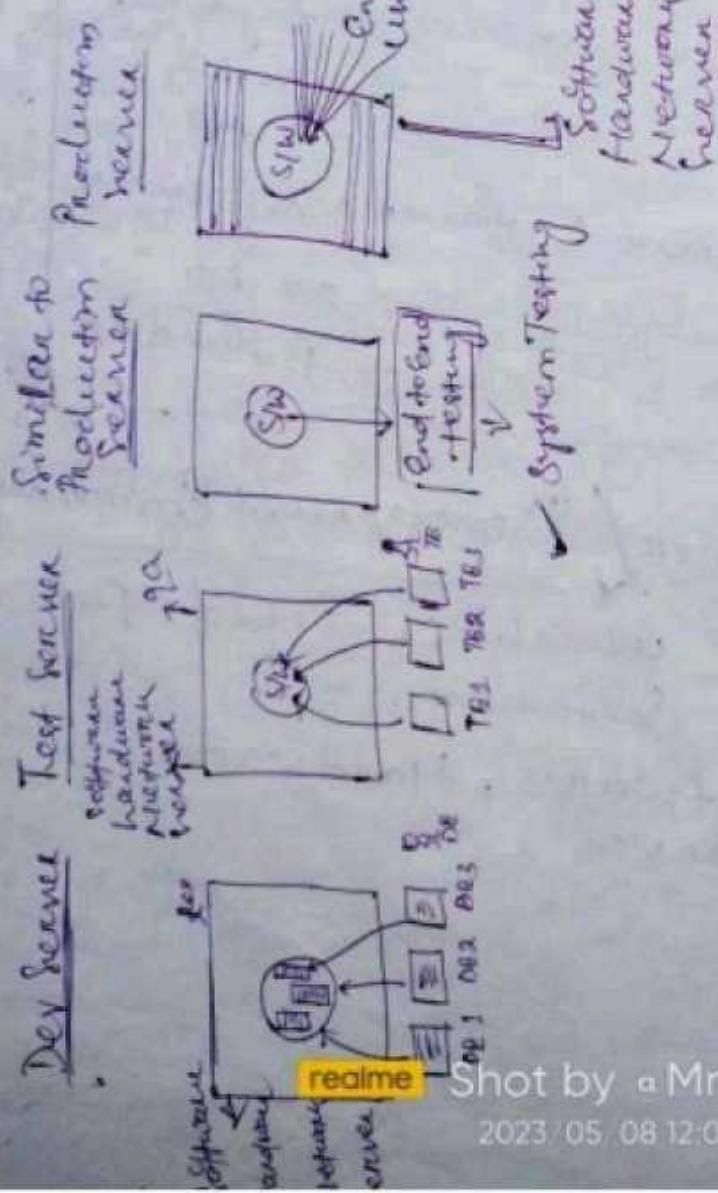
* It is a test env which is used for testing the software.

(*) It contains Software, Hardware, Network & Server.

Production Server / Production Environment

* It is a test env which is used to run the software for the Business.

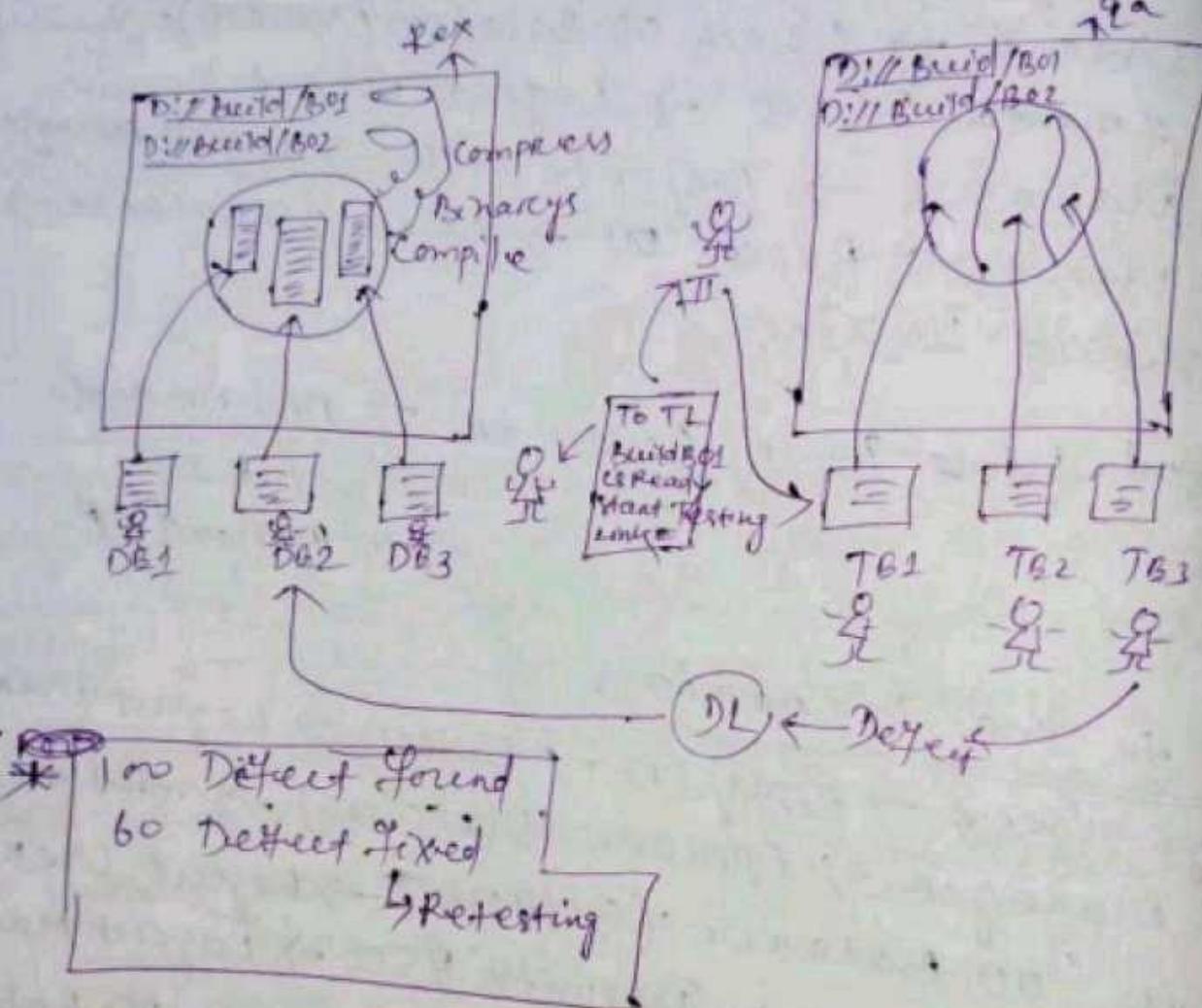
* It Contains Software, Hardware, Network & Server.



- Q How will Developers Test & Leech the code in Real Time?
 OR Explain Build & Release Process?

Test Steamer

Dev Server

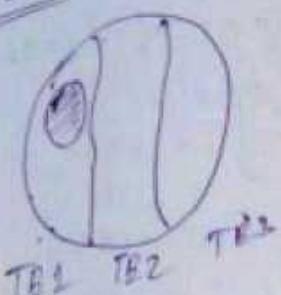


Retesting \Rightarrow (Conformance Testing)

When Developers are giving New Build as a test Engineer We will check whether Defects are fixed Properly or not. This is called as Retesting. (Retesting is also called as Conformance Testing).

Note
Whenever we get new Build, we should
Re-test only the Defect which are fixed.

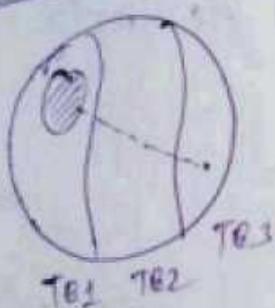
Case-1



TB1 should Re-test the Defect

TB1 should test the impacted region

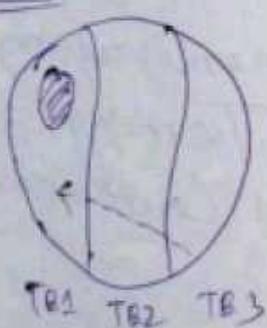
Case-2



TB3 should Re-test the Defect

TB1 should test the impacted Region

Case-3



TB3 should Re-test the Defect

TB1 should test the Impact region

Note

* Owner of the Defect should retest the Defect
* Owner of the region should test the impact region.

Q What is Build ?

→ When we compile all the program we get binarys then we compress all the binarys. we get a file that file we call it as Build.

→ We get build in a diffⁿ format like

1 - JAR [Java Archive]

2 - WAR [Web Application Archive]

3 - TAR [Tape Archive]

4 - ~~apk~~ .apk for Android

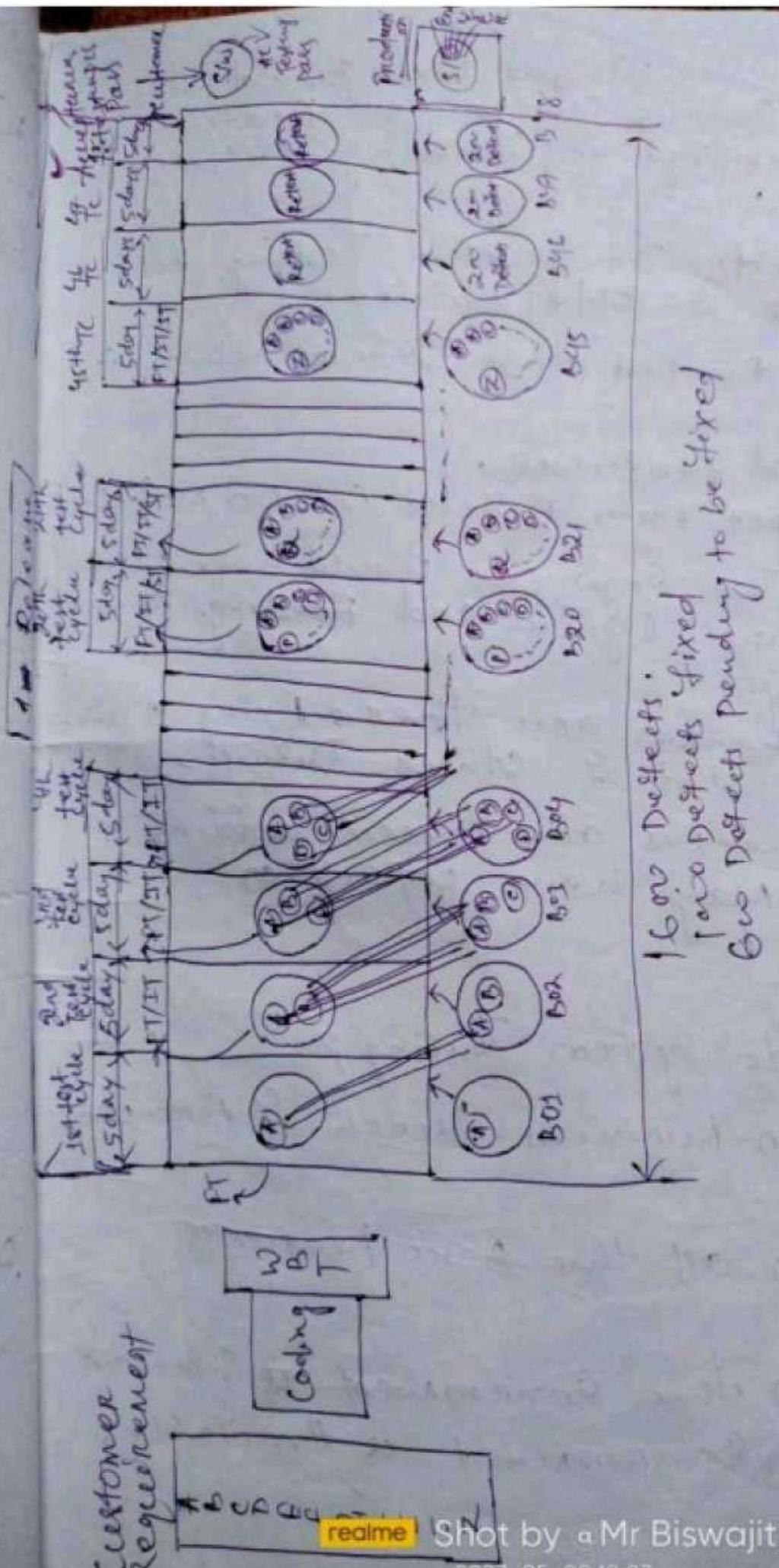
5 - .ipa for IOS

6 - .exe

Continuous Integration (CI) 23/11/22

↳ The continuous synchronization b/w Dev-Environment & Testing Environment we call it as Continuous Integration.

↳ We achieve Continuous Integration with the help of Jenkins. tools



1600 Defects.
1000 Defects fixed
600 Defects pending to be fixed

② Whenever we get new features,
Should start Testing with new Features.
Because Probability of finding Defects
will be More.

- ③ We have to do Integration Testing before
new features & old features.
- ④ We should Re-test the Defects which
fixed.

⑤ Test the old features.
& why to be get new Defects in Old Mod.

- An
- * Test Engineer would have fixed it
in previous cycle and finding it
in current cycle.
 - * Chances are there fixing it
may introduce lot of other Defects.
 - * Chances are there adding
new features may have impact on
old features.

Q-2 When we do System Testing?

- An:
- * When Minimum Bench of People
are Ready.
 - * When all the Basic Features
are Stable.
 - * When the Environment is similar
to Production Environment is Available.

Note

- * Even though all of the features which is requested by the customer is already still we continue testing.
- * Developers will keep up fixing the pending defects and test engineer will be retesting the defects which is fixed and also test all the remaining features.

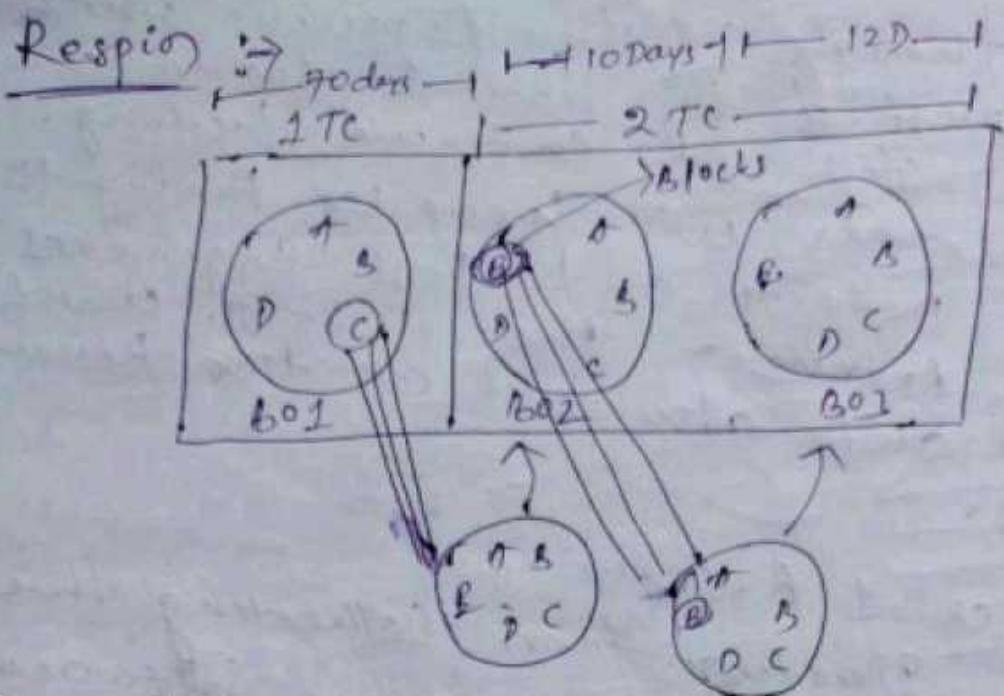
Q) what is 1 release ?

Ans

- * Starting from collecting the requirement, developing the software, Testing of for so many Test-Cycle until we Deploy the software to the Production, is called as 1 release.
- * Scrum or Company is following Agile Model, then it is called as Sprint.

Q) what is Test-Cycle ?

- * It is the effort or time spend to start and finished the testing of a particular build.
- * 1 test cycle will be for 3 days or 5 days or 7 days or 10 days or 15 days.
- * Test cycle varies from Company to Company & Project to Project.
- * Test cycle depends on → size of the application → complexity of the application → No. of Test engineer



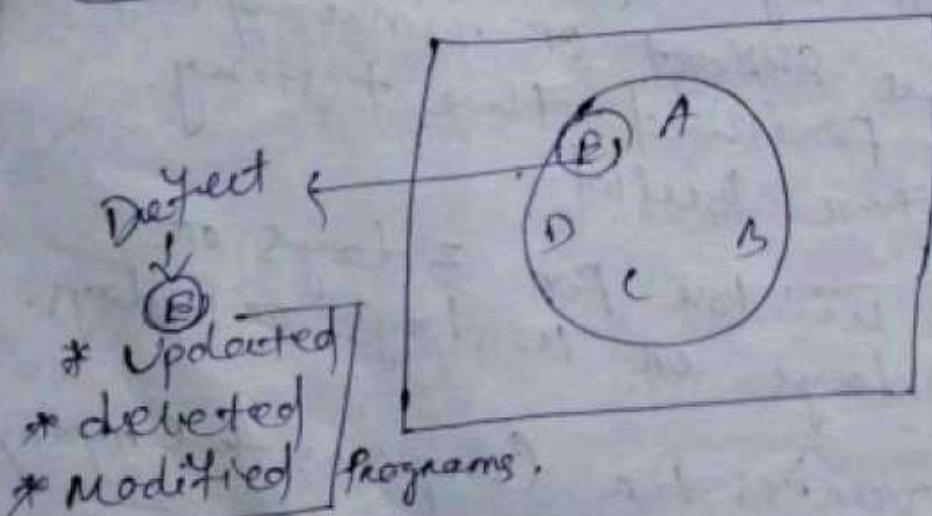
Respin

- * Getting More than 1 build in a single Test cycle we call it as Respin.

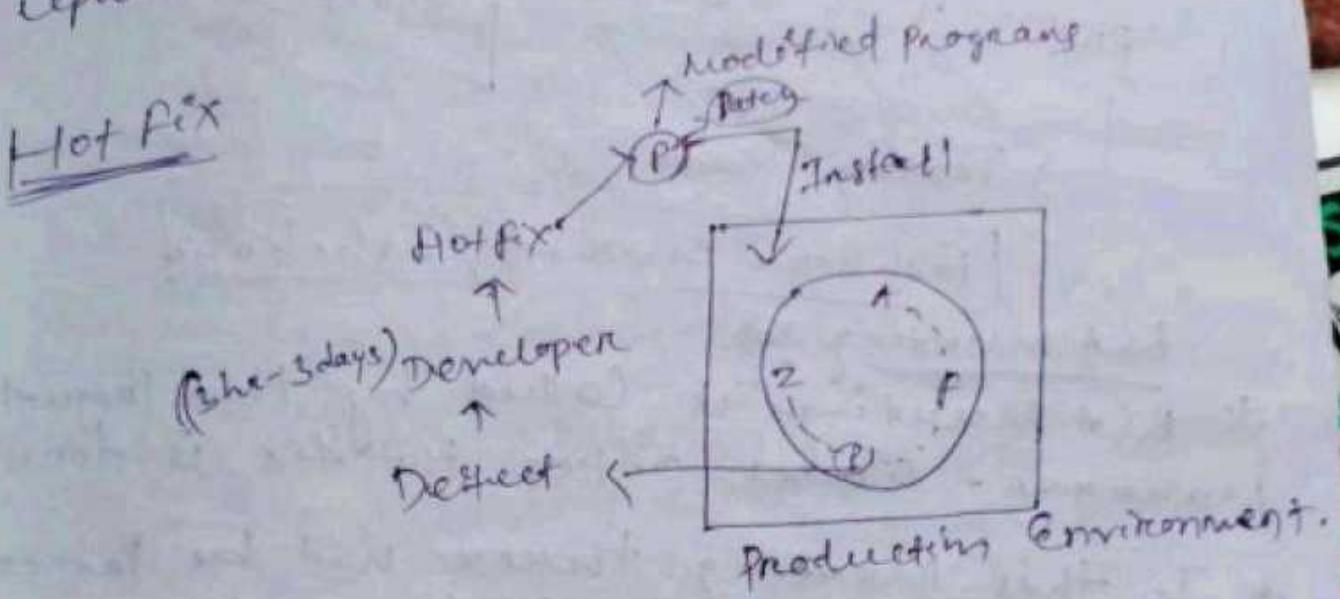
Disadvantages of Respin :-

- * Wastage of Time.

Patch :-



Patch It is a small Software which Contains updated, Deleted, & Modified Programs.



Hot Fix :

* Fixing the Defect occurred in Production Environment we call it as Hot Fix.

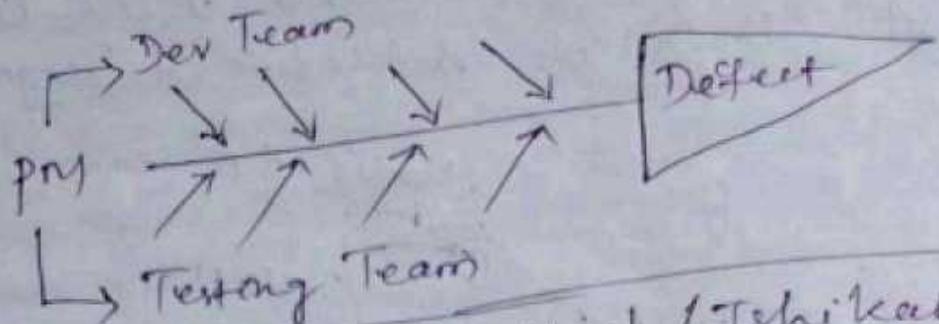
* It is usually Done b/w 3hr to 3 Days.

* Hot Fix is Done using Patch.

* After Hot Fixing the Project Manager will call a Meeting called as RCA Meeting. [Root Cause Analysis Meeting]

RCA Meeting [Root Cause Analysis] also called ~~Root~~ Fish Bone Method / Ishikawa Method.

Draw \Rightarrow



| Fish Bone Method / Ishikawa

RCA Meeting \Rightarrow

- * RCA Meeting is called by the Project Manager, Worx, after Hotfix is done.
- * In this Meeting there will be Project Manager, Development team, & Testing team.
- * In this meeting, we try to find the root cause of the defect.

① Ok

~~Test Plan~~

- > It is a document which contains all the future testing activity.
- > It has stages like ! - -
 - (1) Objective
 - (2) Scope .
 - (3) Approach
 - (4) Testing Methodology
 - (5) Schedule
 - (6) Effort Estimation

- ① Risk
- ② Contingency plan/ Back-up plan
Mitigation plan
- ③ Entry and Exit Criteria
- ④ Test stop Criteria
- ⑤ Test Environment
- ⑥ Test Automation
- ⑦ Defect Tracking
- ⑧ Deliberable
- ⑨ Roles & Responsibility
- ⑩ Test Suites

① Objective :

In this stage we will decide by exactly are we doing testing. What our aim is and what are we trying to achieve.

② Scope :

* In this stage we will decide which features are to be tested and which features are not to be tested.
 * Features to be tested examples
 Ex- Login, Signup, inbox, sent items, spam, etc.

* Features Not to be tested :-

Ex - Privacy Policy, Team Condition etc.

③ Approach ↗

> In this stage we will decide of types of activity & approach we will be doing along with task to be done in Project during Testing phase.

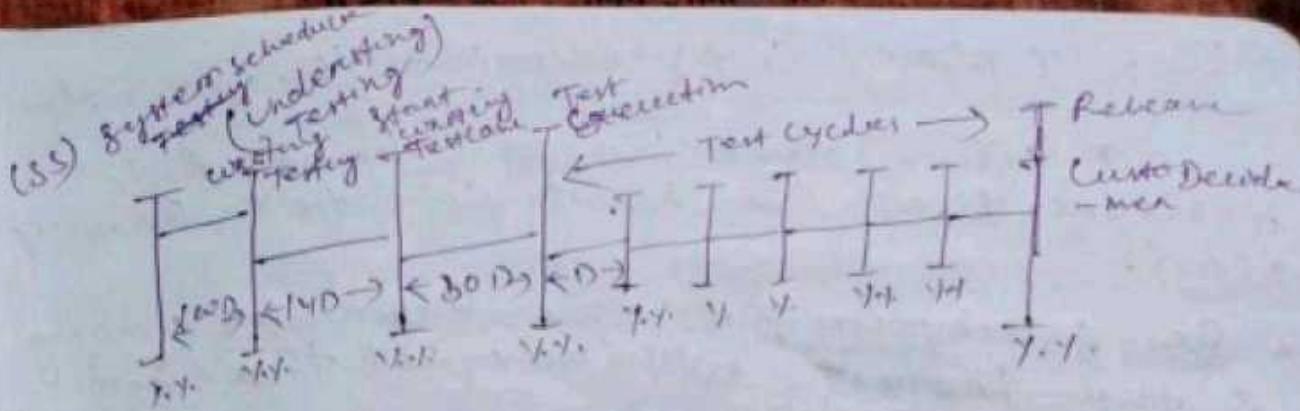
④ Testing Methodology ↗

	Instagram	Spider
Smoke	✓	✓
Functional	✓	✓
Integration	✓	✓
System T	✓	✓
Adhoc T	✓	✓
Performance	✓	✗
Compatibility	✓	✗
Cutover/Deployment	✓	✗

* In this stage we will decide type of testing which we will be doing.

⑤ Schedule ↗

In this stage we will set the timeline for all the activities which we will be doing during Testing.



⑥ Effort Estimation \Rightarrow

- \rightarrow In this stage we will decide the effort it needs to be put in that project.
- \rightarrow In this stage we will estimate the time effort to test the whole project.
- ~~\rightarrow In this stage we estimate the financial effort required to test this project.~~
- \rightarrow In this stage we will estimate the human effort to be put in this project.

⑦ Assumption \Rightarrow

In this stage whenever we are assuming something, it means we have to guess some of the stages & activity.

⑧ Risk \Rightarrow

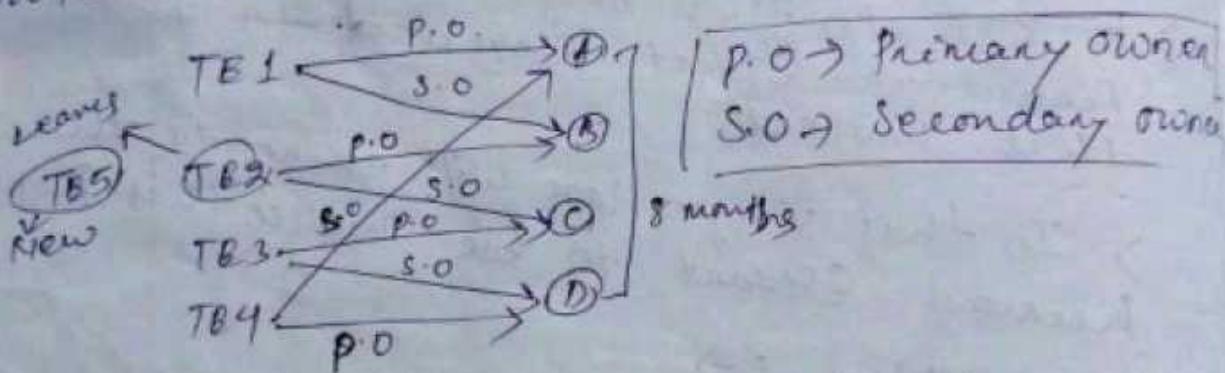
Whenever we are assuming things, it means that we are taking risk.

⑨ Back-up plan / Contingency plan / Mitigation plan

- * When ever we are facing Risk it means that we must create a Back-up plan.
- * Ex- Assumption : All the Engineers working in the Project will stay till the End of the Project.
- Risk → Test Engineer leaving the Project in Middle.

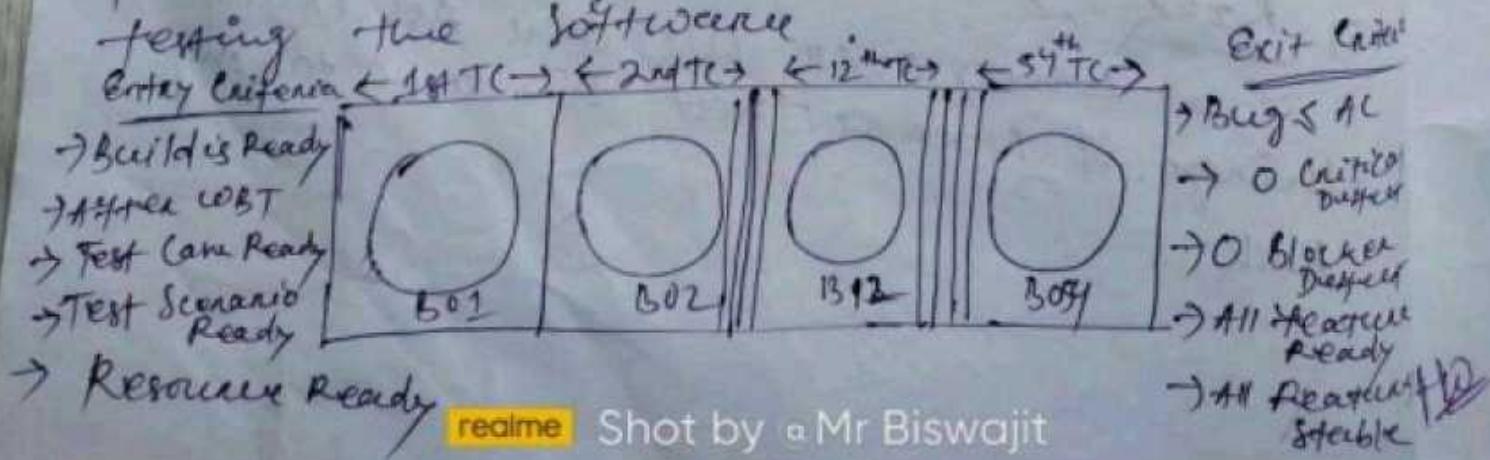
Back-up plan :-

Creating Primary owners & Secondary owners of the Features.



⑩ Entry & Exit Criteria :-

In this Stage we will decide the Conditions upon which we will start testing the Software. & we will decide the conditions in which we will stop testing the Software.



① Test-stop Criteria →

In this stage we will decide when exactly we will stop a particular testing.

Q When do we stop a particular Testing?

Ans → When the quality of the software is either too good or too bad.
→ Two good quality means:-

* When bunch of features are ready

* There are no Critical, no Blocker Defects.

* All the end to end Scenarios

* working properly.

→ Quality of the software is too bad means

* It has Critical or Blocker Defects

* It has too many major & minor bug.

* The build has not pass smoke

-Testing.

* We stop Testing when the schedule is going out of line

* We stop Testing when the budget is going over the limit.

⑧ Test Environment → or Test Bed

* In this stage we will decide all the hardware, software, networks & servers which are to be included in the test-environment.

* Here we will understand & analyze the production environment by the customer & decide whether to add any other platforms, or remove any platform, Software, Hardware, etc.

⑨ Test Automation →

* In this stage we will decide the features which can be automated and the features which can not be automated. We will decide which testing tool we are going to use while testing which automation framework are we going to use while testing.

⑩ Defect Tracking →

* In this stage we will decide the procedure of tracking a defect.

* Here we will set the severity of all the features :-

(i) Blocker Defect

(ii) Critical Defect

(iii) Major Defect

(iv) Minor Defect

* In this stage we will decide the level of priority in this project :-

(i) High Priority

(ii) Medium Priority

(iii) Low Priority

* In this stage we will decide what Defect tracking tool can be going to use in this Project.

⑮ Deliverable :-

* After completing the Testing, here we will identify all the Documents which needs to be Delivered to the customer along with the Software. The documents which needs to be delivered include :-

- (i) Test plan ;
- (ii) Test cases
- (iii) Traceability Matrix
- (iv) Test execution report
- (v) Release Note
- (vi) Graphs & matrices.

⑯ Release Note :-

* It is a document which contains instruction on how to release the software in to production environment.

* Release Note Contains :-

(i) List of all the Defects that are found .

(ii) List of all the Defects that are fixed

(iii) List of all the Features that are Modified or Deleted.

(iv) List of platform on which the Product is Test Cases

(v) List of **realme** Shot by Mr Biswajit Product is Test Cases

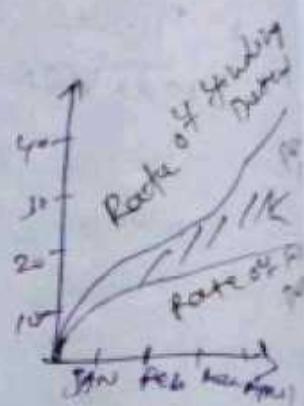
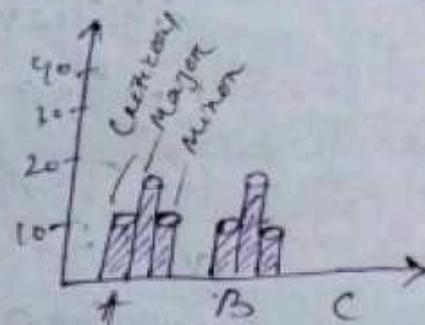
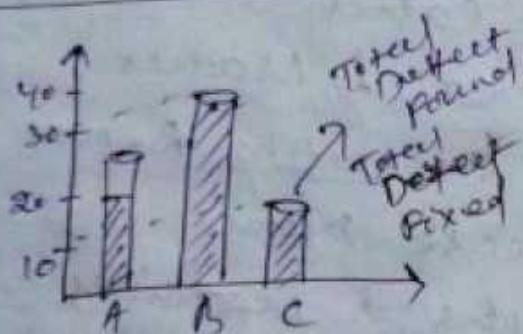
2023/05/08/2023

(vi) Procedure to Install the Product in the Production Environment.

(vii) Graphs & matrices :-

Here we will collect all the Prevalence Data and other properties data and prepare the graphs and matrices to be submitted along with the product.

Disease Distribution Graphs:-



Matrices :- (Disease Distribution Matrices)

Module-A

Module Name	Categorical Defect		Major Defect		Minor Defect	
	found fixed	Not Fixed	found fixed	Not Fixed	found fixed	Not Fixed
A	20	3	11	10	20	19
B	30	20	30	12	22	21
C	20	10	31	30	25	20

Test Effectiveness Matrix

Module-B

Test Engineer	Categories		Major Defects		Minor Defects	
	Defects Found	Fixed	Reported	Fixed	Found	Fixed
Tom	20	3	11	10	20	19
Jenny	30	20	30	12	22	21
Bhavay	20	10	31	30	25	20

⑯ Roles & Responsibility :-)

- * In this stage we will Decide the Roll of all the test engineer in the project during Testing.
- * In this stage we will Decide who will become team leader. The Roles & responsibility of a team leader.
- * Who will become Test Manager?
 - The Roles & responsibility of Test Manager
- * Who will become Automation Test Engineer?
 - features & responsibility of Automation Test Engineer
- * Who will become Manual Test Engineers?
 - Features & responsibility of each Manual Test Engineer

⑯ Template

- * In this stage we will decide the outline & the format of the document which we will be using during Testing of this project.
- * Ex:- format of the Test-Case & Defect Report etc.

Acceptance Testing / Real Testing 17/12/22

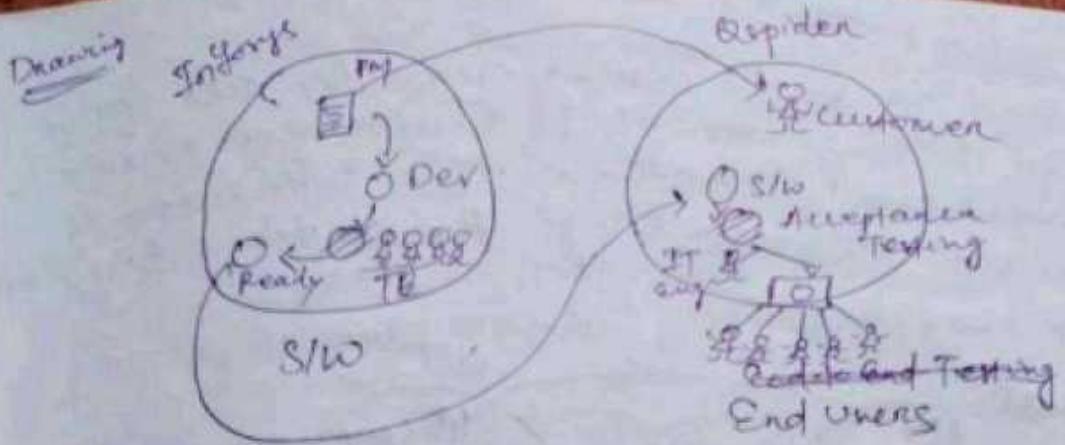
* It is an end to end testing done by IT Engineers to check whether the software is capable of handling Real Time Business Scenario.

* There are 4 approaches of Acceptance Testing :-

- (1) Approach - 1 / Alpha Testing
- (2) Approach - 2 / Beta Testing / User-Acceptance Testing
- (3) Approach - 3
- (4) Approach - 4

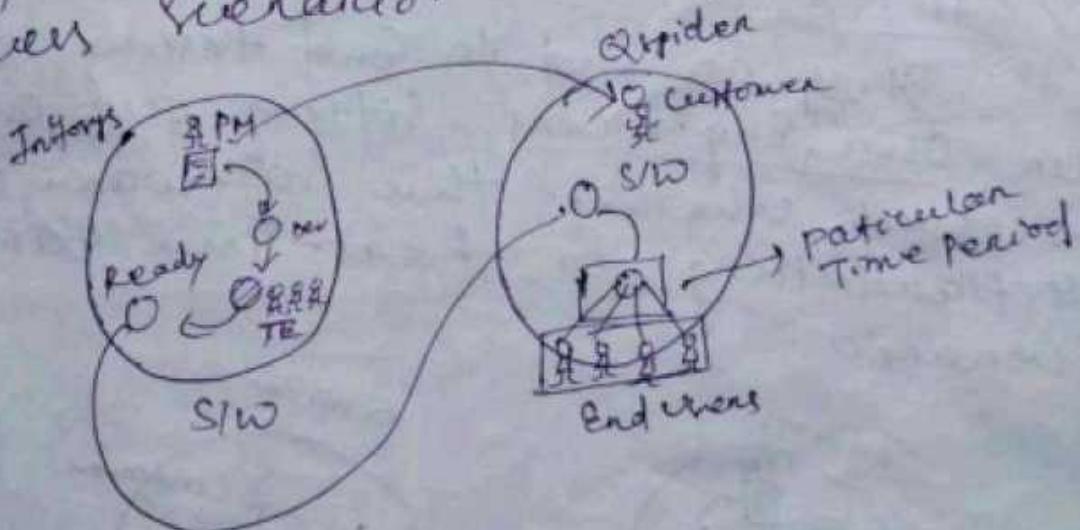
① Approach - 1 / Alpha Testing ⇒

* It is End to End Testing Done by IT Engineers, sitting at the customer place and checking whether the software is capable of handling Real-Time Business Scenario.



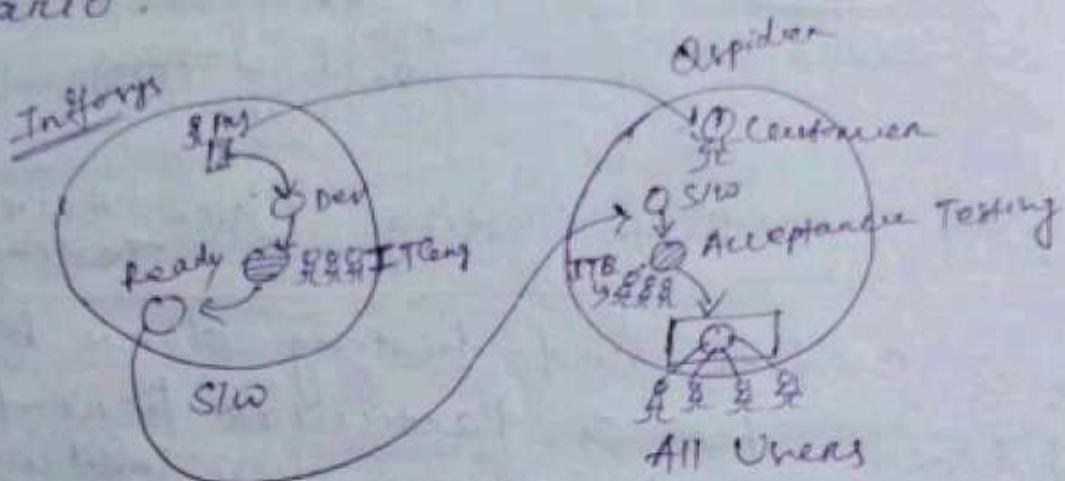
Approach-2 / Beta Testing / User-Acceptance Testing

* It is end to end testing done by End Users for a particular period of time to check whether the software is capable of handling real time business scenarios.



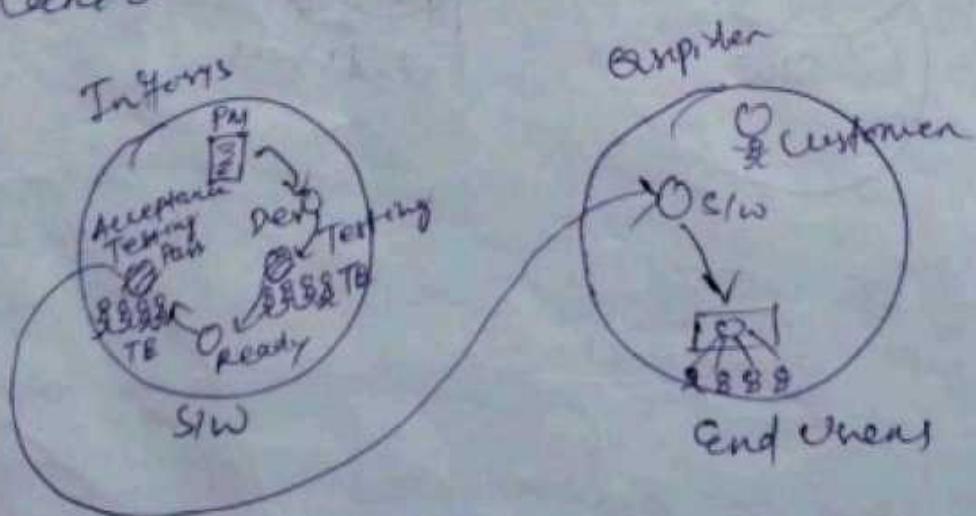
③ Approach - 3 \Rightarrow

It is End to End Testing done by our own own ^{Own}
Engineer sitting at the Customer place and checking whether the software is
 Capable of Handling Real-Time Business Scenarios.



④ Approach - 4 \Rightarrow

It is an End to End testing done by our own ^{Test} Engineers sitting over own place to check whether the software is Capable of Handling Real-Time Business Scenarios.



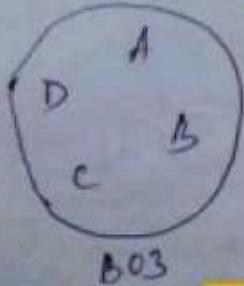
Q) Why do we Acceptance Testing?

- * Because of less time there are chances that the software company might release some bug in production environment to avoid that the customer will doing Acceptance Testing.
- * To check whether the software has any blocker or Critical Defect.
- * To check whether all the features given in the requirement are present in the software.

SMOKE TESTING

SKIM Testing / Build Validation, Testing, or Dry Run
sanity Testing / Health Check Testing
Confidence Testing

- * Testing the basic and critical features before doing thorough Testing in a software we call it as Smoke Testing.
- * While doing Smoke Testing we will only do the Testing.
- * Smoke Testing should be done for each & every build.

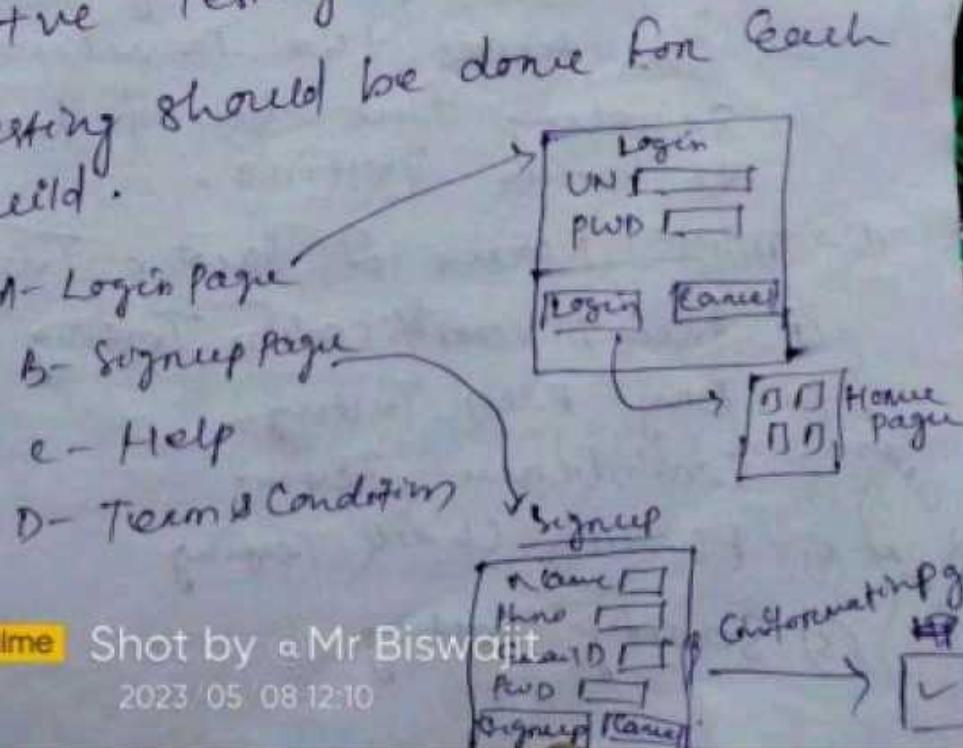


- A - Login Page
- B - Signup Page
- C - Help
- D - Terms & Conditions

realme

Shot by Mr Biswajit

2023/05/08 12:10



Q) Why do we do Smoke Testing?

- * We do smoke testing to check whether it has any Blocker, Critical Defects.
- * To check whether the Build which we have received is Testable or not.
- * To check whether the Build is installed properly or not.
- * To check whether we have received any Broken Build.

Q) When do we do Smoke Testing?

- When we get the Build in Testing Environment.
- The Customer before Starting Acceptance Testing will do Smoke Testing.
- The Customer after Releasing the Software in to Production Environment will do Smoke Testing.
- Sometimes the Developers before generating the Build for Testing will do Smoke Testing.

Q) Diff' Name of Smoke Testing?

- ① Build verification Testing
- ② Dry- Run Testing
- ③ Confidence Testing
- ④ Health Check Testing
- ⑤ Sanity Testing

Q2 v.v. Qmp Diff b/w Smoke & Sanity Testing?

SMOKE TESTING

- * We test the Basic & Critical Functionalities Before doing Thorough Testing.
- * In Smoke Testing we only do +ve Testing.
- * Smoke Testing is Wide & Shallow Testing.
- * In Smoke Testing we document & execute Test Cases.
- * Software Developer or Tester perform smoke Testing.
- * Smoke Testing is a subset of Acceptance Testing.
- * Smoke Testing is to verify stability.
- * Smoke Testing is like General Health check up.

SANITY Testing

- * Like In Sanity Testing we test the Basic & Critical Functionalities Before doing Thorough Testing.
- * In Sanity Testing we only do +ve Testing.
- * Sanity Testing is Narrow & Deep testing.
- * In Sanity Testing we do not refer to any formal Document.
- * Tester alone perform sanity Testing.
- * Sanity Testing is a subset of Regression Testing.
- * Sanity Testing is to verify Rationality by reason.
- * Sanity Testing is like specialized health check up.

Performance Testing / Spikeline Testing

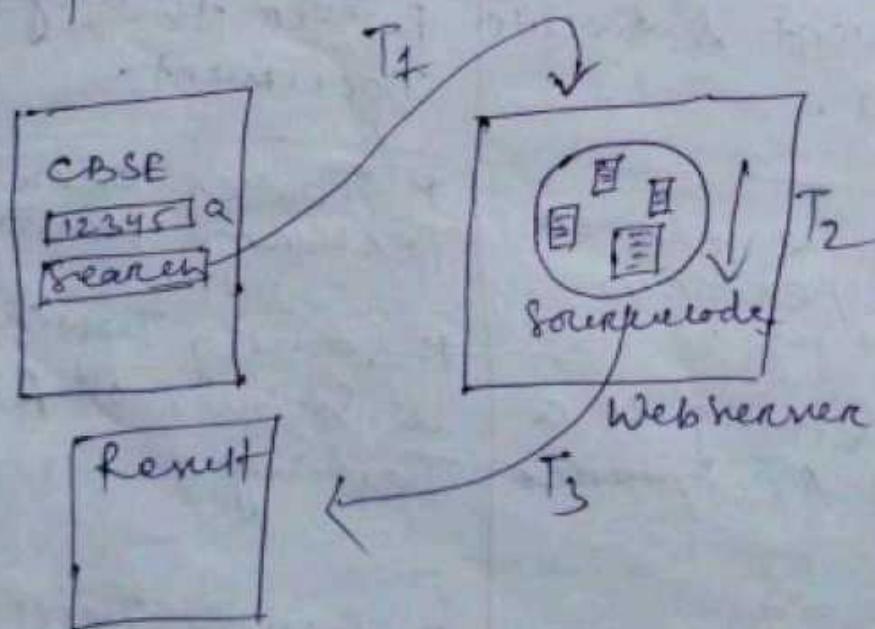
or (Non-Functional Testing) 2/12/22

Testing the Stability and Response Time of an Application by applying Load. We call it as Performance Testing.

Load :> The number of Users using the Application.

Stability :> The Ability to withstand Load.

Response Time :> The time taken for the request to reach Servers, Execute the program and generate response. We call it as Response Time.



Response Time

$$RT = T_1 + T_2 + T_3$$

Q) How to do Performance Testing?

→ Performance Testing is done by Performance Testing tools.

Ex- J-METER

→ NeoLoad

→ Load Runner

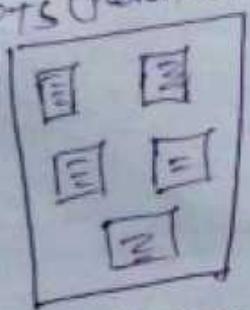
Q) How to do performance Testing with performance Testing Tool?

Requirement

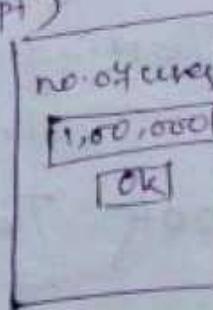
→ 1,00000 users response

Time is 2 seconds

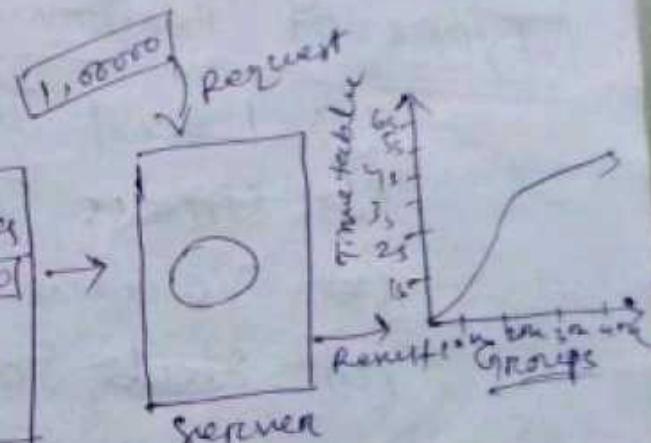
PTS (Performance Test Script)



J-Meter



I-Meter tool



Explanation of the diagram

* Enter the Performance Test - script into the J-Meter tool.

* It will ask the no. of users to be generated, enter 1,00,000 users and click on OK.

* 1.lakh users request will be generated and send to the servers.

* The **realme** Shot by Mr Biswajitaph
in the form of a

- * The test engineer will Analysis the graphs and check whether it has pass or fail.
- * If it has failed the test engineer will send the software back to the developer.
- * The developer will make all the necessary changes in the software and send it back for testing.
- * The test engineer will repeat at the step until the test case has passed.

Type of Performance Testing :-

- (1) Load Testing
- (2) Stress Testing
- (3) Volume Testing
- (4) ~~Shock~~ Soak Testing

① Load Testing :-

- * Testing the stability and response time of an application by applying load less than or equal to the demand of no. of users.

② Stress Testing :-

- * Testing the stability and response time of an application by applying load greater than realme the demand no. of users.

③ Volume Testing

Testing the stability and response time of an Application by applying Huge Volume of Data.

④ Soak Testing ↗

Testing the stability and response time of an Application by applying Load for a continuous period of time -

Q: Why do we do Performance Testing?
or (for what kind of Application we do Performance Testing)?

- A: * for Application which brings in huge Revenue.
- * for Application which has Huge Customer Base.

Q: When do we do Performance Testing?

- Ans → We do Performance Testing when the software is functionally stable.
- We do performance Testing for some Application from SDLC Test only.

Performance Testing : When TE finds a defect then communicate to the developer who will do some changes / modifications in code in order to improve the performance & called as Performance Tuning.

Advantage of Smoke Testing

- ① In Candy Merge we will find Bank & Customer Resources.
- ② Due to open having agreement to fix the defect.
- ③ Test cycle will not be ~~delayed~~ Delay.
- ④ Retraining will not be required.

Brain Storming

Session
start?

- Any other system study
- ↳ Once after finding all the scenario we will conduct Brain storming Session.
 - ↳ In this meeting Complete testing team will be present.
 - ↳ we conduct this meeting to improve the efficiency of Test cases.

- Adhoc Testing / monkey testing / Random Testing
- * Testing the Application Randomly without referring to any formal Documents
 - * While doing Adhoc Testing we only do -ve Testing.

Q: Why do we do Adhoc Testing?

- As the End users will not be ~~using~~ using any the Application in any formal way there are chances that end user might find some bugs. To avoid if have bugs we do Adhoc Testing.
- By testing the software with formal document only the no. of bugs which comes we find will be very less that is why we do Adhoc Testing.
- To Increase the no. of bugs
- To Improve the test coverage
- To improve the software is working according to the implicit requirements.

Implicit Requirement:

This are the requirement which are not given in the requirement specification but are understood.

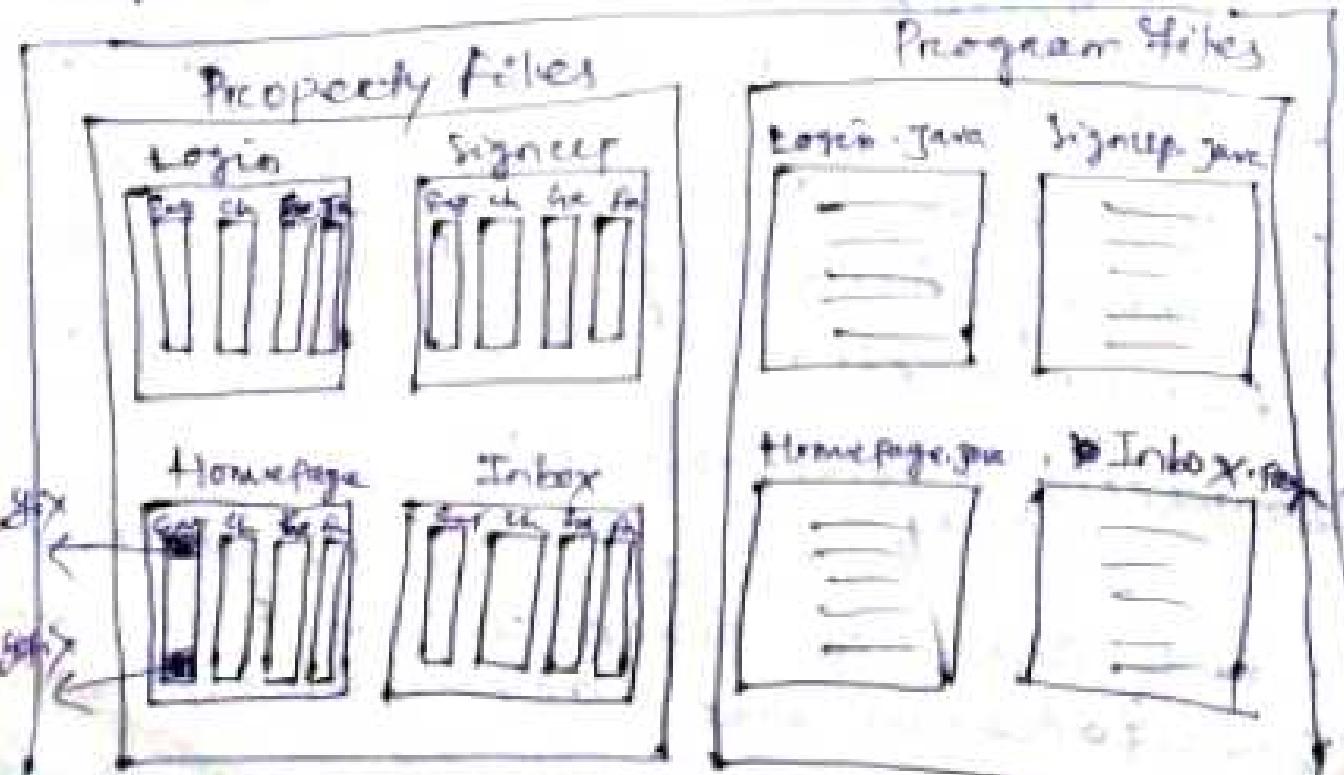
Q) When do we do Adhoc Testing?

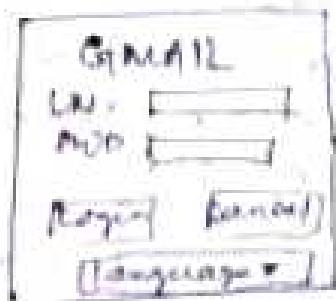
- Ans → We do Adhoc Testing Before giving the software to the customer.
- While doing smoke testing we never do Adhoc Testing.
 - When doing ~~smoke~~ testing (ST, IT, ST)
if we find any Adhoc Scenario, we ~~use~~ ^{use} Adhoc testing and continue in Adhoc Testing.
 - If we are getting too many Adhoc scenarios then document those scenarios and do it whenever the test engineer come free.

3/12/22

Globalization Testing

* Testing the application which is developed for multiple languages and formats:





* There are 2 types of Globalization.

Testing ?

- (1) Internationalization Testing (I18N)
- (2) Localization Testing (L10N)

Explanation of the Diagram →

* The user select a language from the browser and searcher for the application to be tested.

* Web server gets the request and the respective program from the programming file is executed.

* While sending the response it selects the property file according to the response to be generated and sends the response.

* The test engineer identifies

* The test engineer will analyze the property file which is selected with the help of varied prefix & suffix.

* The test engineer will decide whether prefix & suffix and decide whether the testing is passed or failed.

Note → The first Engineer will Assign
Particular prefix, or suffix before
executing.

(1) Internationalization Testing (I18N):

* Testing the Application which is
Developed for international languages
and formats.

(2) Localization Testing (L10N):

* Testing the Application which is
Developed for local languages and
formats we call it as Localization
Testing. \rightarrow GUI → (Graphical User Interface)
(End)

Usability Testing \rightarrow GUI / Cosmetictesting

Testing the User friendliness
of an application is called as

Usability Testing:

Q Why do we do Usability Testing?

Ans → To check the Look and feel
of the Application.

→ I will check whether all the
important features are accessible
within 3 clicks.

→ The Application should be easy
to understand.

→ All the important features are
frequently used features should
be easily accessible.

Q) When do we do Usability Testing?
* When the application is functionally stable.
* For some applications we do usability testing from SDLC Stage only.

Q) For what kind of applications will we do Usability Testing?
* For applications which has wide range of users.
* For applications which brings with huge amount of revenue.
* For applications which does not give any training for their end user.

Accessibility Testing

→ Testing the software which is developed for the physical challenge people.

OR
→ Testing the Usability of an application developed for physically challenged people.

Ex: Colour blindness - Handicapped person etc.

* Yellow Box Testing :-

* Testing the warning messages of an application is called as Yellow Box Testing.

* It is a subset of Usability Testing.

STLC

SDLC

- Requirement Collection
- Feasibility Study
- Design
- Coding
- Testing
- Installation
- Maintenance

STLC

- System Study
- Write Test Plan
- Write Test Case
- Traceability Matrix
- Test Execution
- Defect Tracking
- Test Execution Report
- Retrospect Meeting

STLC: → It is a Step by Step Procedure to test the Software.

Stage-1

(1) System Study →

* In this Stage We read the requirement and understand the Customer Requirement.

(2) Write Test plan →

* In this Stage we document all our Customer Testing Activities.
* We decide (i) How many Test engineer do we need.

(i) What Features to test

(ii) What is the Schedule for Testing.

(iv) What are other roles of each engineer working in the Project etc.

③ Write Test Case :

→ In this stage we prepare the Test Case.

* It is a document which contains all possible scenario.

* Here we have stages of writing the test case.

1st stage :> (1) Identifying the scenario
(2) Documenting the Test Cases.

(3) Reviewing the Test Cases.

(4) Fixing the review comments.

(5) Varietying Approval

(6) Test plan Approval

(7) Storing it in Repository.

④ Traceability Matrix

→ In this stage we prepared

the Traceability documents.

→ It is a document which ensure

that each and every requirement has got at least 1 test case.

Requirement No	Detailed Requirement	Test Case Name
1.0	Login Page	—
1.0.1	Login Button	—
1.0.2	Cancel Button	—
1.1	Signup Page	—
1.1.1	Name	—
1.1.2	PL No.	—

* Types of Traceability Matrix :-

- (1) Forward Traceability Matrix
- (2) Backward Traceability Matrix
- (3) BiDirectional Traceability Matrix

(1) Forward TM → (1st to Last)

Starting from the top of the requirement and Tracing it till the bottom we call it Forward Traceability Matrix. (1st to last)

(2) Backward TM →

Starting from the last requirement and Tracing it till the requirement we call it Backward TM.

③ Bidirectional TM \Rightarrow Cross-references / References
Starting from any feature in between,
and Tracing in both the direction,
is called Bidirectional TM.

Note: This document is also called
as Cross - Reference Matrix (Cross
Traceability Matrix).

⑤ Test Execution \rightarrow

* In this stage we will start Executing
the test cases.

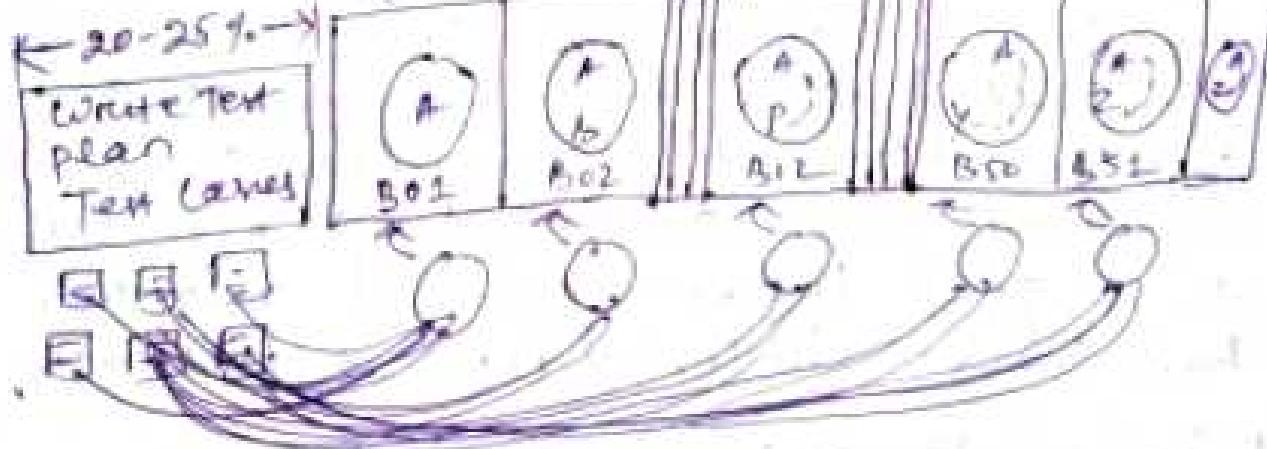
* This is where the test Engineers
will spend most of the time.

* The maximum Defects are found
in this stage only.

\hookrightarrow this stage only.

* This is where the test Engineers
are most productive of the Organization.

75-80% of time



but how difficult it is to do
from the beginning to the end.
So what approach do we want

⑥ Defect Tracking

- * In this stage we will discuss the process of how the Defects will be tracked.
- * What Defect tracking tool are going to be used.

⑦ Test Execution Report :

- * It is a document which contains details analysis on how many test cases are executed, how many Work Items are raised, what are how many work failed, what are the Defects we have incurred.
- * How many Defects are fixed, how many Defects are not fixed, etc.

⑧ Retrospect Meeting \Rightarrow (Project closure or post-mortem)

- * It is a Meeting conducted by the Project Manager.
- * Once after the Project is completed the test Engineers and the Development Engineers meet together and discussed about all the good Process followed and the bad process followed in that Release.
- * Before starting the next Release they will refer to all this good and bad practices referred in Retrospect document from the previous Release.

* By see this document, they will try to make sure that the good process are repeated and bad process are stopped.

TEST CASES

* Test Case is a document which contains all possible scenario.

Test Scenario

* It is a guideline description of a possibility.

Test Case

* This document contains step by step procedure to test the software.

* What are the drawbacks of writing the test cases?

- * If we do not write test cases we have depend on other test engineer.

* The quality of testing will depend upon the mind of the test engineer.

* The quality of the testing will depend from person to person.

* There will be no consistency in test execution.

* The information about the user's task is not to be added to the document and it is necessary if

Q) When we do write Test Cases?

* When the Developers are developing the software, the test engineer will write the test cases.

* When the Developers are modifying the software the test engineers will modify the test cases.

* When the Developers are updating the software, the test engineer will update the test cases.

Q) Why we do write test cases?

1) To have better consistency.

2) To improve the test coverage.

3) To make sure that we are depending on Process ~~and~~ rather than a Person.

4) To avoid Training Every new Engineer ~~of~~ on the Product or the Requirements.

5) Test Case ~~is~~ acts as a prove document for the Customer Developing team and the testing team.

6) Test Case acts as a base document for all the Automation Test Scripts.

7) If we have documented the test cases then you don't have to remember the scenarios.

8) Get others + Organi
9) If time +

Test
(1) → Case
(2) → Equ
(3) → Scen
(4) → Del
(5) → Ste

[2]
Wh
various
and b
reas
redme

Decision

→ Get user
Combined

→ Ex A & B
the user
State +

→ In State
Condition
under

of this a
Condition
testing
Input
System

- 8) If the test cases are documented then the testing will be done in an organized way.
- 9) If we write test cases then the time taken to execute will be less.

Test Case Design Technique :-

- (1) → Error Guessing
- (2) → Equivalence Partitioning
- (3) → Boundary Value Analysis
- (4) → Decision Table Based Technique
- (5) → State transition

(3) Boundary Value Analysis :-

Whenever the Input is Range of values derive scenarios for $a_1, a, a+1$, and $b, b+1, b-1$.

Range of Values Age $\square (5 - 55)$

$$\begin{array}{ll} a = 5 & b = 55 \\ a_1 = 6 & b+1 = 56 \\ a-1 = 4 & b-1 = 54 \end{array} \left. \begin{array}{l} \text{Boundary} \\ \text{Values} \end{array} \right\} \text{Defect}$$

Decision Table Based Technique :-

- It used for functions which respond to a combination of inputs.
- Ex A submit button should be enabled if the user has entered all required fields.
- State transition :-
- In state transition technique changes in input conditions change the state of the application under test (Act). The tester can perform this action by entering various input conditions in a sequence. In this technique testing team provide valid & invalid input test values for evaluating the system behavior.

Format of a Test Case :-

ITRMST -

HEADER

1) Test Case Name :-

2) Requirement No :-

3) Module No :-

4) Severity :-

5) Test Data :-

6) Pre Condition :-

7) Test Case Type :-

8) Brief Description :-

BODY

Seq/ Step No	Description	Input	Expected Result	Actual Result	Status	Comments

FOOTER

Approved By :-

Signature :-

Approved Date :-

Reviewed by :-

⑦

The Test Case Divided into 3

- Parts :-
- (1) Header
 - (2) Body
 - (3) Footer

Header :-

1) Test Case Name

Format :- 'Project Name - Module Name - Scenario'

Ex. → 'FB - Login - All Functionality Scenario'
→ 'Instagram - UNAID Scenario'

2) Requirement No. :-

Hence We will be committing about which Requirement case we covering this Stage Case.

3) Test Data :-

It is a Data which we need to execute this test case.

Ex. :- www.facebook.com,

UN = ABC
PS = 123

4) Pre Condition :-

This are the Activity needs to be completed before executing this test case.

Ex. Set Activation fees and Interest rate
(for system Testing ON Features)

⑤ Test Case Type :- (PTC/ITC/STC)

Hence we write PTC/ITC/STC:

⑧ Brief Description :-

Here we will write briefly about what we will do in the test case.

FOOTER

Approved By :- Your Test Lead will Approve all the test case.

Approved Date :- The Date when it was Approved by the Test Lead.

Author :- The Name of the Test Engineer who wrote the Test Case.

Reviewed by :- Names of the Person who Review the Test Case.

TEST CASE DESIGN TECHNIQUE

(I) Error Guessing :-

We guess all of the Possible Defects or Error Based on Our Experience, Knowledge, and Instinct.

Ex Phone ↗

→ 9876543210 (10)

→ 987654321 (9) X → 123@#\$(x)

→ 98765432154 (11) X → ABC#@123#(y)

→ ABC(x)

→ @#\$!(x)

→ ABC123(x)

→ ABC@#\$(x)

② Equivalence Partitioning : 8/12/22

(i) PRISMATI :

Whenever there is a range of values, then we must find 1 valid value & 2 Invalid values.

Ex Name
(3-30)

① ② ③

(ii) Set of values :

Whenever there is set of values, as input find 1 valid value and 2 Invalid values.

Valid - 10

Invalid - 45

Invalid - 5

Product
shoes - 10 Pant - 30
shirt - 20 number - 40

(iii) Boolean Value :

Whenever the input is Boolean value then Identify Scenarios for all the Possible values.

Eng	✓
GEE	
French	
Chinese	
Spanish	

④ Practise →

When the input is Range of Values, with Deviations consider the Range.

We find all the possible values and Identify the scenario.

Ex-

Insurance

18-30 → 20,000/Year (18-30)

31-49 → 30,000/ya 17 18-30 31-49 10

22

45

⑤ Boundary Value Analysis →

Whenever we have Range of Values like this we will identify all possible scenario by taking $(A, A+1, A-1), (B, B+1, B-1)$

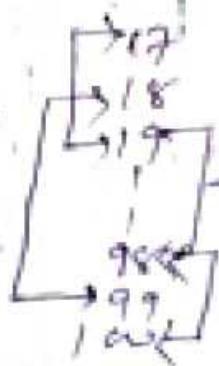
Ex- (3-30)



⑥ Test Case Optimisation →

Whenever we have Range of values then we can skip equal partitioning and go for Boundary Value Analysis.

Ex - Age []
(18-99)



if it works for boundary values then it will work for the range in between.

Functionality Test Cases :-

Q How to write functionality Test Cases
Ans * Always start with writing Any Navigational steps.

* Takes 1st Component :-

→ Identify all Error guessing scenarios.

→ Identify all Equivalence Partitioning scenarios.

→ Identify all the Boundary value scenarios.

* Takes 2nd Components :-

→ Identify all Error guessing scenarios.

→ Identify all Equivalence Partitioning scenarios.

→ Identify all the Boundary value scenarios.

Sign up

Name	[Text Box]
Phone	[Text Box] (3-30)
Email	[Text Box] (10 max)
Gender	<input type="radio"/> Male <input type="radio"/> Female <input type="radio"/> Other
Submit	[Text Box] Cancel

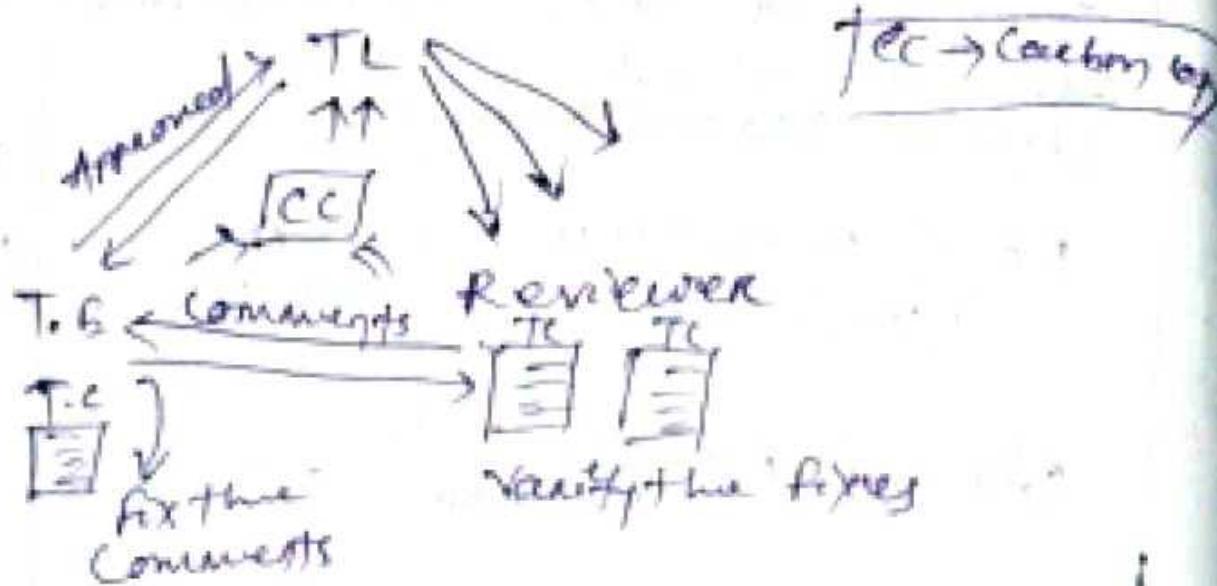
Integration Test Cases

- Q: * How to write Integration Test Cases?
- ↳ if you have covered something in Functionality then Don't cover it in Integration.
 - ↳ if you prioritizes your steps then the total no. of steps will be less.
 - ↳ Always imagine the Application while writing the Test Cases.
 - ↳ Always write your Test Cases in Simple English.

<u>S/N</u>	<u>Description</u>	<u>User</u>	<u>Expected result</u>
1	Open the browser Enter the URL, Click on Search button	user1@192.168.1.100	Login page should be displayed.
2	Login with user1 & user1 password	UN - ABC PWD - 123	Home page should be displayed.
3	Click on Account Transaction Page		Account Transaction page should be displayed.
4	Enter from A/c no to A/c no and Click on Transaction button	Acc - A TAN - B Amt - 1000	Confirmation page should be displayed.
5	Click on A/c Bal button		A/c Bal page should be displayed and balance should be -1000.
6	Click on Transaction button		Transaction page should be displayed and details should be displayed.
7	Login with user1 & A/c		Login page should be displayed.
8	Login with user1 & user name & password	UN - XYZ PWD - 245	Home page should be displayed.
9	Click on A/c Balance button		A/c Balance page should be displayed and Old Balance should be Old Balance + 1000.
10	Click on Transaction button		Transaction page should be displayed & details should be displayed.

Test Case Review Process \Rightarrow

12/12/22



TC \rightarrow Action by

Q: On what basis do you assign someone to review the Test Case?

- ↳ When another Engineer is working on a similar module.
- ↳ When the Other Engineer ~~has~~ has worked on the same module in previous project.
- ↳ When the Other Engineer has good knowledge of the domain.
- ↳ When the Test Engineer is working on the project from the beginning and ~~he~~ he has clearly understood the requirement.

Review Ethics

- ↳ Always Review the Test Case and not the Author.
- ↳ Only find the Mistakes in the Test Case and not the Screenshot.
- ↳ If there is any mistake in the Test Case after the review process both the Author and the reviewer will be responsible.

Q Why we do Review the Test Cases?

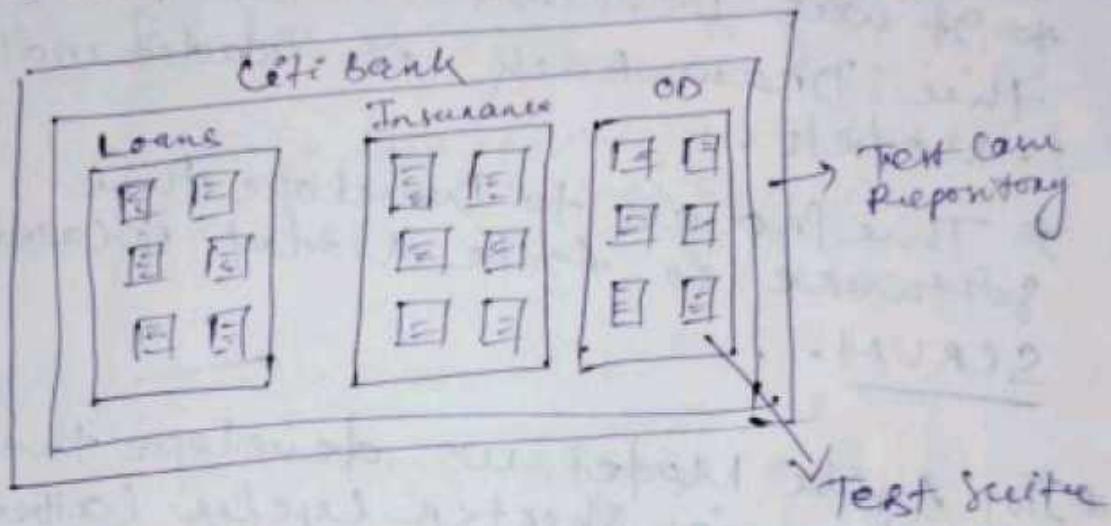
- ↳ To check whether Author has not written any wrong scenario.
- ↳ Missing scenarios.
- ↳ Extra Scenario / Repeated Scenario.
- ↳ Whether the Test Case are Simple to Understand.
- ↳ Whether the Test Data and Pre-condition in the header and footer are written properly.
- ↳ Whether the Test Case is Prepared in the format decided in Test plan.

Test Case Repository →

It's a place where all the Approved test cases are stored.

Test Suite →

Group of similar Test-Cases is called "Test suite".



Where do we write Test Cases?

→ Ms-Word, Ms-Excel, Test Case management Tool.

Agile Model

SCRUM MASTER

→ The Head of the Scrum Team is called as the Scrum Master.

→ His prime role is To facilitate complete Meeting and Coordination between all the Stake holders it means:-

(i) Greeting Architect Support to developer.

(ii) Greeting Product owner support to test engineer.

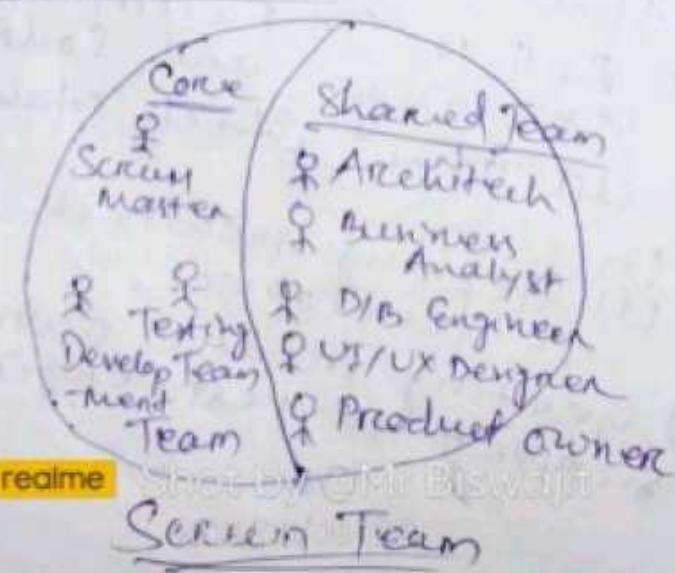
realme Shot by Mr Biswajit

Agile Model

12/12/22

- * It is a step by step procedure to Develop a Software.
- * It is an Incremental and Iterative Process.
- * It was Developed to Overcome the Draw-Back of Traditional Models.
- * The Process to Develop the Software in Agile Model is called SCRUM.
- * In Agile Model we develop the software in shorter cycle called as Sprint.
- * All the Engineers working in the Process are a part of SCRUM Team.

SCRUM TEAM ⇒



- * The Scrum Team is formed by Scrum Master.
- * Scrum Team consists of Core Team & Shared Team.
- * Core Team consists of Scrum Master, Development Engineers, & Test Engineers.
- * Shared Team consists of Architect, Business Analyst, Database Engineer, UI/UX Designer, (User Interface / User Experience) Product owner.

Product Owner ?

- > He is the one who acts as a customer.
- > Product owner can be from the customer sides or from the software company sides.
- > He is the one who sets the Acceptance Criteria.

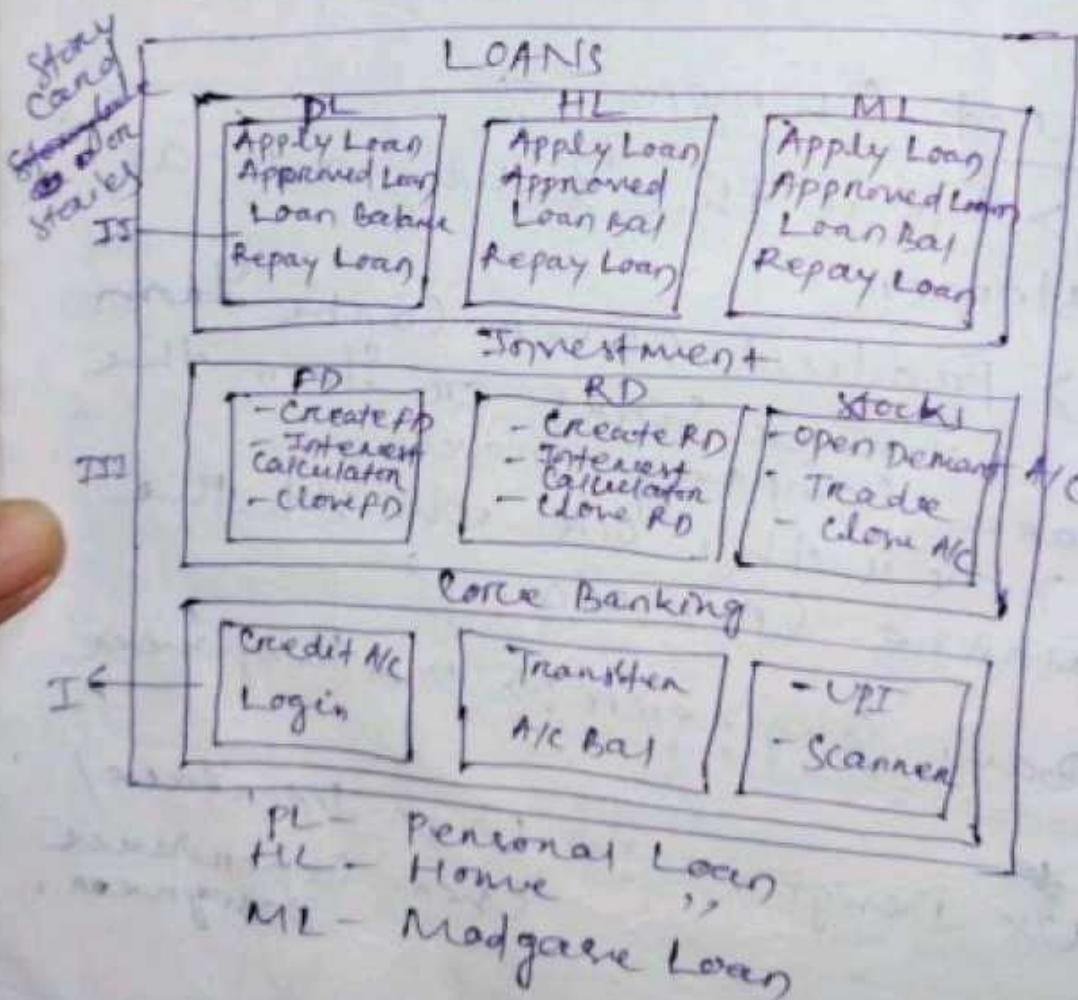
Data Base Engineer ? D/B Engineer

- * UI/UX Designer :> User Interface / User Experience Designer.

Product Back Log

- * It is the prioritized list of stories which are to be completed within that Project.
- * It is managed by Product owner.
- * The product owner, Scrum Master, Business Analyst, Architect, are involved while making the product Back log.

Product Back Log



* In Product Backlog the stories generally need not be detailed.

Sprint Backlog :

* It is Prioritized List of stories and the associated tasks which are to be completed within that sprint.

Sprint Backlog :

* It is list of stories which has to be developed for 1 sprint.

* It is created by Scrum Master.

* When Product Backlog is the base to create Sprint Backlog.

Product Backlog

* It is a prioritized List of Stories.

* This Product Backlog will prioritize by Scrum Team.

* By looking into product Backlog we will create Sprint Backlog.

Scrum Ceremony (4 types)

* Sprint planning meeting, Daily stand-up meeting, Sprint Review Meeting, Sprint Retrospective meeting.

scr

① Sprint planning Meeting

- * Here the Entire screen Team Seats together and pull Stories from the Product backlog.
- * The Scrum Master Assigns each Engineer (Development & Test Eng.) Stories.
- * The Engineers will Decide the task which are to be completed to build the Stories.
- * Each Engineer will Decide estimate time to complete a Story. It is called as Story point.
- * Generally the Sprint planning meeting will Last for 8 hours (2h per week).

② Daily stand up Meeting / RoleCall Meeting / Daily Scrum

- * In this Meeting the whole Scrum Team Meets.
- * This Meeting is arranged by Scrum Master.
- * Here Every Engineer speaks about:-
 - What task did they complete yesterday.
 - What problems they faced yesterday.

→ What tasks are they planning to do today.

→ what Problem/ impediment are they expecting today.

~~imp~~ ✓ The Scrum Master will try to solve some problem in the meeting itself. If it takes too much time, he will write in the impediment backlog.

* In this Meeting Every Engineer will speak point to point basis only.

* This meeting generally last for 10 to 15 minutes.

* This meeting is conducted in the beginning of the day.

~~STORY Board / white Board:~~

Engineer	Stories Pending	Stories In progress	Stories Completed
Tom	2 3 19	15	11
Jerry	3 4	2 5	1
Mr. Bean	4	1 2 5	3
Ballayea	5	9	11 2 3

* It Contains the List of Tasks which are to be :-

- Pending
- In process
- Completed

* It's generally hanged on a wall where every body is moving.

* It is done to build sense of responsibility in the engineer.

Sprint Retrospective Meeting :

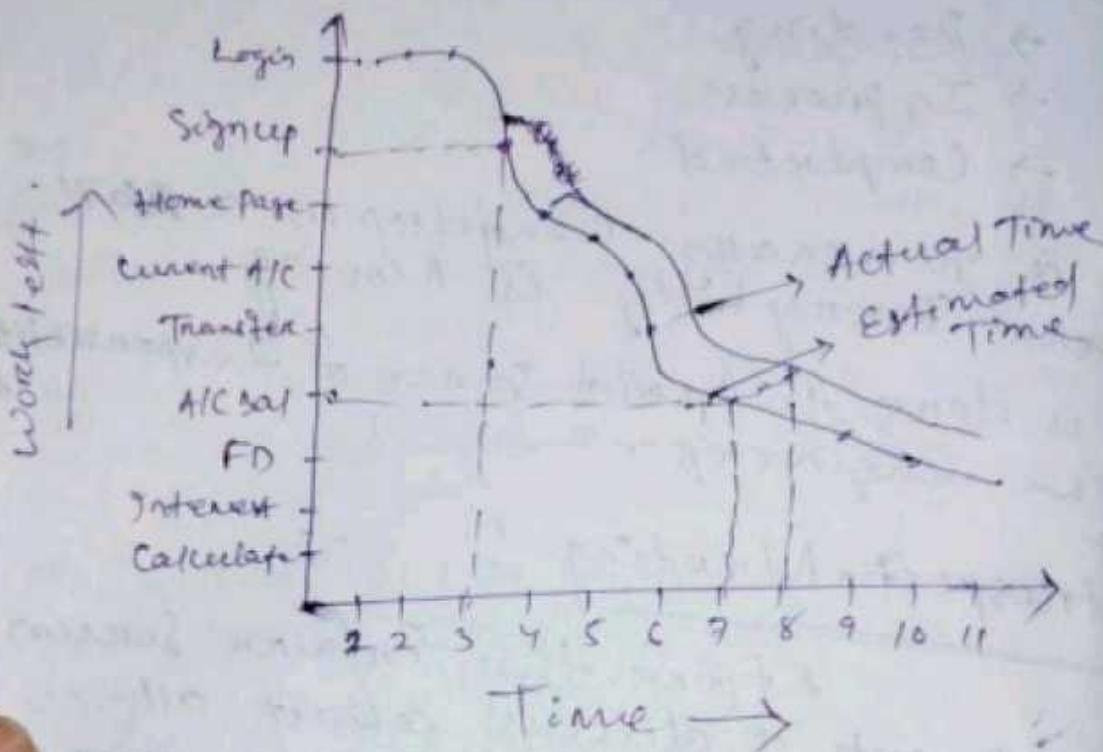
* Here the Entire Team Meets & discuss about all the achievement and mistakes there were performed in the sprint.
* We document these activity in Retrospect document.

* Before starting of the next sprint during sprint planning Meeting we refer to this document all make sure that all the good activity were repeated and all the mistakes were not repeated.

Impediment Backlog

→ It is a document which contains all the problems.
→ It is created by Scrum Master.

Burn Down Chart



* It is the graphical representation of Work left vs Time.

Story Point ⇒

→ It is the Estimated time taken to complete one task or a story.

CHICKEN ⇒

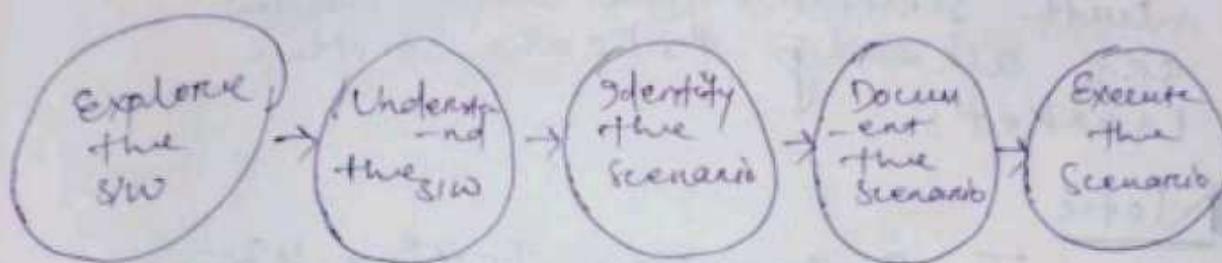
He is the person who will observe the whole process and not will be doing any task.

— X — X (end)

Exploratory Testing

11/12/22

- * Exploring the Application, Understanding the Application, Identity the Scenario, documenting the Identity scenario and then Executing the documented Test Cases without referring to any formal document is called as Exploratory Testing.



When do we do Exploratory Testing?

- * When there is no requirement specification given.
- * When we don't have enough time to read & understand the requirement specification.
- * When we don't understand the requirement it self.

What are the Draw-Back of Exploratory Testing?

- * We might miss understand some bug as a feature.

- * We might miss understand some ~~bug~~ features as bug.

- * We might not be

realme

Shot by @Mr Biswajit
the Me

2023 09 08 12:22

Q How do we overcome the Drawbacks of Exploratory Testing?

- * By getting clarity about the requirement or Application from the Business Analyst, Development Team or Customer itself.
- * By Domain Knowledge
- * By Comparing the Application with Similar Application which are already released in the Market.

Note:

In Exploratory Testing We do functionality, Integration & System Testing.

Regression Testing

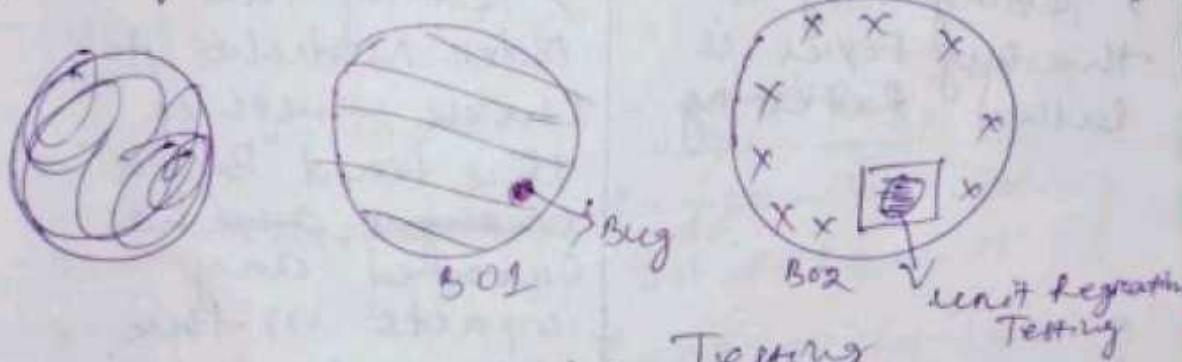
- ⇒ Testing the Unchanged features of an Application to check whether the Bug Fixes (^{unchanges}) have Created any impact or not.
- ⇒ Retesting the Old module to check whether the Bug Fixes (^{unchanges}) has Created any impact or not

* There are 3 types of Regression Testing :-

- (1) Unit Regression Testing
- (2) Regional Regression Testing
- (3) Full Regression Testing

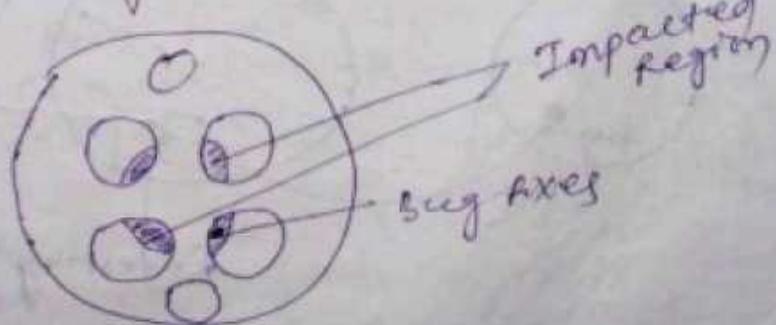
① Unit Regression Testing

Testing only the Bug Fixes or changes is called as Unit Regression Testing.



② Regional Regression Testing

Testing the Bug Fixes and only the impacted Area is called as Regional Regression Testing.



③ full Regression Testing \Rightarrow

Testing the Bug fixed or changes along with all the Remaining Modules or Features, is called as full Regression Testing.

Q) Diff b/w Retesting & Regression Testing?

Retesting

> Testing only ~~does~~ the Bug fixes is called Retesting.

Regression Testing

> Testing the Older Modules to check whether the fixed Bug or changes area has created any impacts on the other Modules.



and
the
atencies.
Testing.

Note

Repetation of PT/IT/ST on older Modules is called Regression Testing.

Progression Testing :-

- * Testing the new Module or changes is called Progression Testing.
- * Testing the other older features Because We have added a new feature is called Regression Testing.

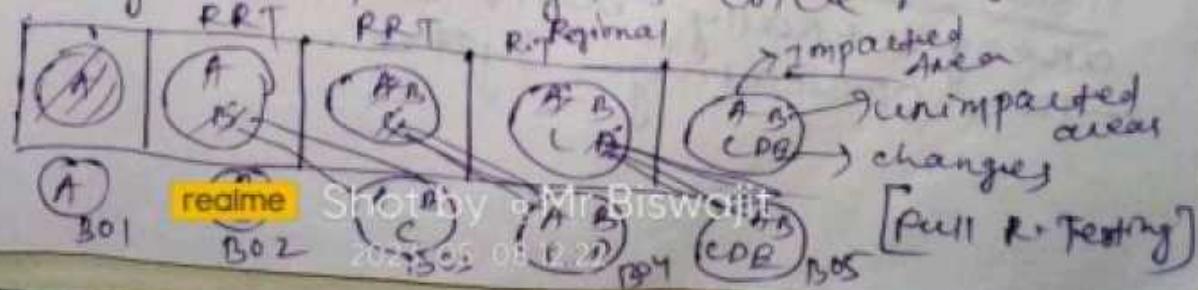
When do we do Regression Testing?

- We start doing Regression Testing in 2nd build of 1st release.
- We start doing Regression Testing of 1st build in 2nd release.

Q

Why/when to be go for Full Regression Testing?

- ① We go for Full Regression Testing whenever changes are done in the Core Features.
- ② We go for Full Regression testing every few test cycles once.

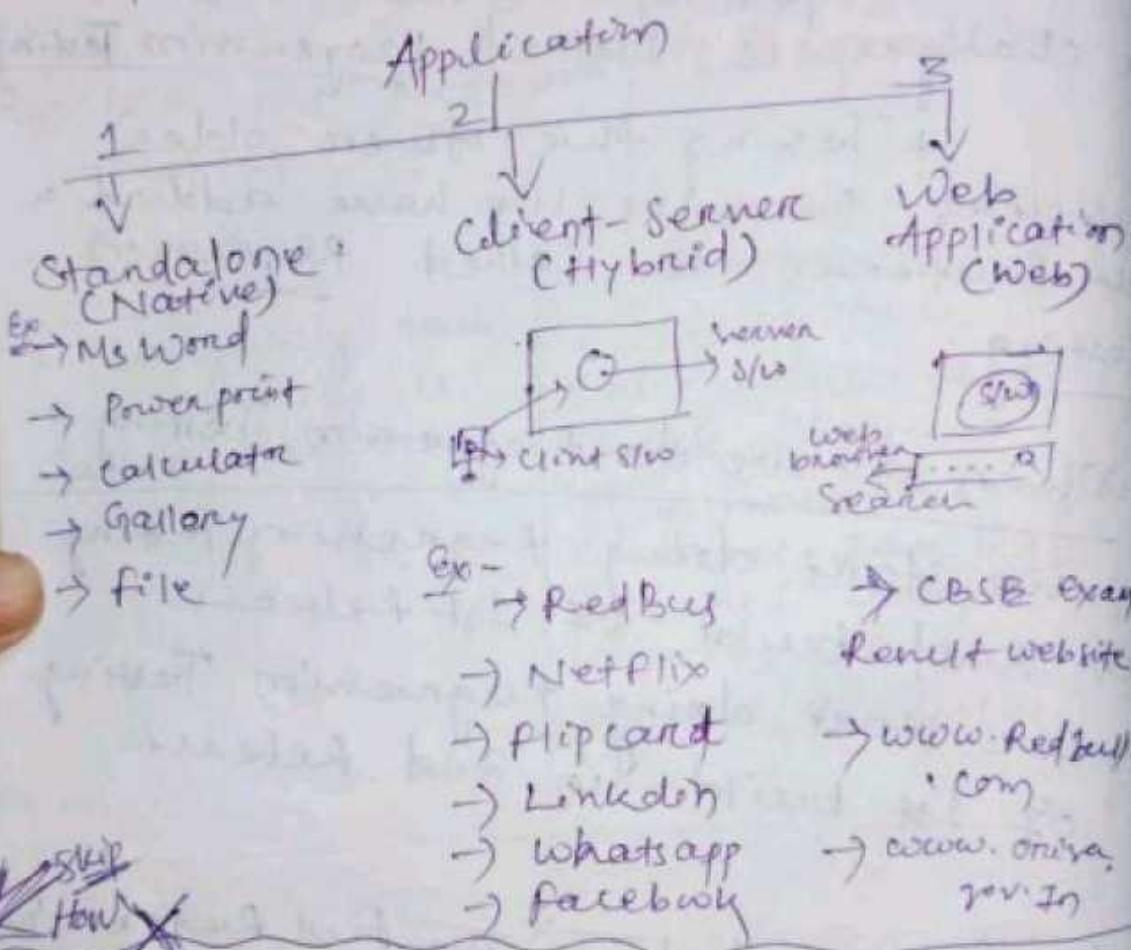


Shoutby @Mr.Biswajit

[Full R+T Testing]

Compatibility Testing \Rightarrow Crossed Test
Testing the Application
in different Hardwares, Software
and Operating systems, is called
as compatibility testing.

Types There are 3 types of Application.



Ex- Gallery, My files, Notepad,
MS-Office, power point etc.

② Client - Server Application

→ These Applications have the original software in the server called as server software. and can be access directly to the server or the user can download software something called as client software in to their device also.

→ This are also called as Hybrid Application.

→ Ex- RedBus, Facebook, Instagram, WhatsApp, Netflix etc

③ Web- Applications ↗

→ This Application are the type of application which can only be accessed through Web Browser.

Ex:- Exam Results websites
www.pepsi.com etc.

 When we do compatibility testing?

- ① When the software is functionally stable in the base platform then we think about testing the software in different platforms.

win10
base platform

Google Analytics

win10 = 55% → Base Platform
win11 = 16%
win 8 = 50%
chrome = 5%
Android = 15%.

How to do Compatibility Testing?

It depends on the type of application.
There are 3 types of application.

- ① Stand Alone Application
- ② Client Server Application
- ③ Web Application

① Stand Alone Application

Any application which is used only by one user at a time and also internet server is not required. This kind of application is known as stand alone application.

Ex MS paint, Notepad, Calendar, Calculator.

② Client Server Application

- ① There are two kinds of software one is client software and the other is server software.

- ② User can access server software with the help of client software and internet and server is required here. Ex Client Server,  Shot by Mr Biswajit  Instagram.

Note: We do Compatibility Testing only for Client Software, not for Server Software.

Web Application

- ① Any Application which is used with the help of Browser is called as web Application.
- ② It is a kind of Client Server Application where in Browser acts like a Client Software.
Ex: Facebook.com, inst.com, yahoo.com
- Q: How will they allocate work in real time while doing compatibility testing?

Approach-1

TB1		win10 win11 Linux
TB2		win10 win11 Linux
TB3		win10 win11 Linux

Approach-2

TE1		edge win10
TE2		edge win11
TE3		edge Linux

For 1 Test Engineer to understand the complete software practically not possible so we prefer Approach-1 to do Compatibility Testing. Because in this Approach each & every Test Engineer will get a chance to compare his bunch of Features in diff platforms. So that he can find more no of defect.

- Q: what types of defect we can find while doing compatibility Testing?

A: ① Alignment issue

② Object Overlapping

③ Scattered Content

④ Scroll Bar issue

⑤ Change in look & feel of the application certain projects may not be displayed in certain Browsers.

Globalization Testing

① Localization Testing

- * Testing the Application which is developed in Local Language and format.
- * Here we check whether currency format, dateformat, pin code format, and colour of the image format is displayed as per the country standards or not.

② Internationalization Testing

Testing the software which is developed for multiple language is called Internationalization testing.

- * Here we check whether content is displayed in right language or not.
- (i) Content is displayed in right place or not.
- (ii) features are broken or not with the language changed.

How to do I18N Testing (for Japanese language)

- ① Go to the Japanese property file.
- ② Add prefix `$&` Suffix to the contents of the property file.
- ③ Save the property file. Open the application, select the language (Japanese). Corresponding page will be displayed.
- ④ I will check for prefix if prefix is correct means content is displaying in right language.
- ⑤ I will check for the suffix if suffix is correct means right content is displaying in right place.

Note: Pseudo Translation :-

As a test engineer for our understanding purpose we will add prefix and suffix to the contents of a property file in order to test the software in multiple language. This concept is called as Pseudo translation.

what type of defects we can find while doing I18N testing?

- ① ~~change~~ Changes are their content may not be displayed in right language.
- ② Changes are their right content may not be displayed in right place.
- ③ Alignment specification problem covers the language changes.
- ④ Tool tip defect.

Tool Tip

- * Tool tip means when user points ~~over~~ cursor over the image, rectangle box will be displayed which explain above the image.
- * Tool tip should be displayed with the language which user has selected if not then it is called tool tip defect.

Regression Testing

① How do we find impacted Areas

① We find impacted Areas Based on Product Knowledge.

→ As a test engineer we know how each and every module is inter-related and on that knowledge the test engineer can find impacted Areas.

② We can find impacted Areas By preparing Impact Matrix.

Changes	Sign up	Login	Compose	Inbox	Profile	Logout
Attachments	X	X	-	-	X	X
pwd	-	-	X	X	-	-

③ We can find Impacted Areas by conducting Impact Analysis Meeting.

→ (i) Test Engineer will attend daily standup meetings with the Scrum Master.

(ii) During this time the entire Scrum team will discuss about changes as well as impacted areas.

Agile Methodologies

~~Agile~~

Q) why Agile is important

- ① Agile is important Because we make profit in lesser time compare to other Models
- ② Agile makes people responsible

Q) When do we use Agile Methodology?

- 1) When the customer is not clear on requirement.
- 2) When the client expects quick delivery.
- 3) When the customer doesn't give all the requirement at a time.

Story Board / White Board

* It is a Board which contains Engineer name, pending tasks, Ongoing task, completed task.

Eg Name	Pending	Ongoing	Completed
A	3	2	1
B	2	3	1
C	1	3	2

Burn - Down Chart

↳ It is Graphical Representation of work left out versus time on the project (concrete graph).

Burn - up chart :-)

It represents how much work has been completed in a proj.

Scrum Master

* This Complete Meeting is driven by Scrum Master.

* His prime role is to ~~facilitate~~ facilitate & co-ordinate the Complete Meeting & co-ordinate between all stake holders (relevant per

Ex ① Getting Architects Support to the developers

② Getting Product Owner's Support to testers to understand the Requirements

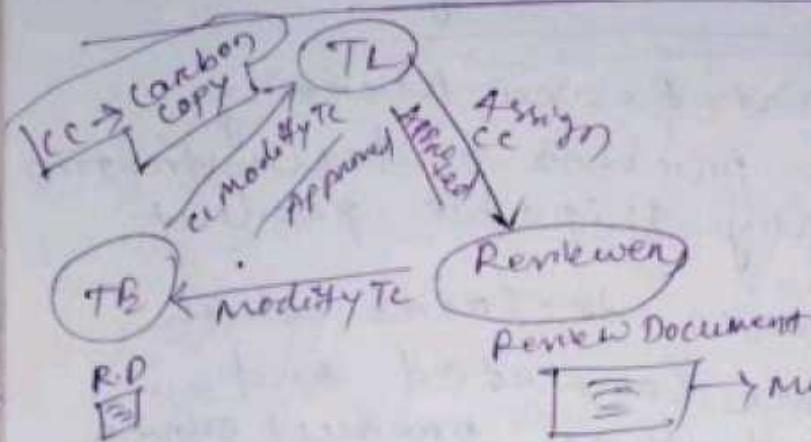
Test Script

It is an automated Programs which is executed After Manual Testing

Manual Testing

↳ Manual Testing is the Process Manually Testing Software for finding defects or Bug.

Q) Test Case Review Process or Peer Review



Mistakes

- ① Repeated Scenario
- ② Wrong "
- ③ Missing "

- ① The Author will write all the test cases and then send it to the test Lead.
- ② The test Lead will Assign all the test cases to the reviewer.
- ③ The reviewer will Review all the test cases, while reviewing he might identify certain mistakes like - missing scenario, wrong scenario, repeated scenario.
- ④ All the mistakes will be documented by the reviewer this document we called it as Review document.
- ⑤ The Author will go through the complete Review document and then they will modify the test cases.
- ⑥ The same Modified test cases will be given to the reviewer as well as CC to Test Lead.
- ⑦ The reviewer will check whether the modified test cases are correct if all the test cases are perfect then reviewer will send an Approval mail to the Test Lead.
- ⑧ Test Lead will send Approved mail to the Author.

~~if the realme phone is approved then only Author will express his opinion on the build.~~

Scrum Ceremony (Agile Model)

④ Sprint Review Meeting

- * In the Sprint Review Meeting Scrum team present a demonstration of a potentially shippable product.
- * Product owner declares which features are completed and not completed. Then product owner adds the additional features to the product backlog based on the stakeholder's feedback.

Daily Stand Up Meeting / Scrum Meet / Toll Call Meeting

- > It is a meeting which is done by the entire Scrum team.
- > It is a meeting which is conducted every day for 15 minutes.
- > In this meeting we discuss about:-
 - what are the tasks that is being completed yesterday.
 - what are the problems that is being faced by the engineer yet.
 - what are the tasks that is being planned.
 - what are the problems that are going to be faced today.

Defect Triage

If there is nothing but the classification or grouping the Defect & giving severity to the Defect we call it as Defect Triage.

→ We go for Defect Triage whenever we have too many Defects to be fixed but time is very less.

Defect masking / Latentized Defect

One defect ~~hides~~ ^{Hide} Another Defect is called Defect masking/latentized Defect.

Defect seeding

~~Important~~ Intentionally adding Defect is a software to check the efficiency of test engineer by Development Lead ~~to~~ the Instruction of Test Lead.

Defect Age :-

Defect Age is time taken to finding the Defect to fixing the Defect.

for " " we will identify Test Scenario
for a pen .

- ↳ To check that the Pen is properly writing in the paper .
- ↳ To check that the Pen cap is properly integrated to the Pen .
- ↳ To check that weight, color etc of the pen .
- ↳ To check that the text written by the pen is erasable or not .

Q) How will you test a chair ?

Q) How will you test a lift ?

Q) Why test ?

Functional Testing

* Here we test the working of the software .

* Here we test the customer requirement .

* Functionality T.
Integration T.
System T etc.

Static Testing

* We will do static testing in the early stage of the SW .

* While doing static testing we do not execute the

Non Functional Testing

* Here we Test the Behavior of the SW .

* Here we test the Customer Experience .

* to Performance
Usability T etc .

Dynamic Testing

* We will do dynamic testing in latter stages of the SW .

* While doing dynamic testing we will execute the code .

* It will be defined .
* In this we will check source code doing test on other box testing .

Quality C

* finding and rectifying the errors .

* In QC we identify the problem .
* We have to make the process ready .

* QC = QA .

Q) why

① To check correct platform .

② Develop one place in SAMU in the system get due to avoid Testing .

- * It will Prevent Defect.
- * In this testing we will check the source code by doing Path testing or other white box testing.

- * It will find and fix the defect.
- * By this testing we will performance Function & Non-Functional Testing.

Quality Control (QC)

- * finding the Defect and making sure that the Defect are fixed is QC.
- * In QC it is Product Oriented.
- * It starts only after the product is in testing and ends when product is ready.
- * QC is Part of QA.

Quality Assurance (QA)

- * It is the process which helps maintain the quality of the product.
- * In QA , It is User Oriented.
- * It starts Before the product is ready and continues till the product is delivered to the customer.

Q) why we do Compatibility Testing?

- ① To check whether the SW is working Consistently in all the platform.
- ② Developer Developing the SW in one platform and Test Testing the SW in same platform. When we give SW in the End user they will start using in multiple platforms. So they might get defect which leads loss to customer so avoid this loss we do Compatibility Testing.

of Acceptance Criteria

It is nothing but the Maximum no. of Defect present in the SW.

Q) When we release a SW in Production Environment?

- > When All the Features are Stable.
- > When the test Environment is similar to the Production Environment.
- > When there is no blocker / Critical Defect.
- > When all types of testing is conducted by the SW.

Manual Testing Questions

- Q) what is the Diff b/w SDLC & STLC?
- ↳ SDLC deals with Development / coding of the software while STLC deals with Verification and Validation of the software.
- Q) Traceability Matrix?
- ↳ It is a document which shows the relationship between test cases and requirement.
- Q) What is Black Box Testing?
- ↳ It is used to test the SW without knowing the internal structure of code or program.
- Q) At what stage of the SDLC does testing begin?
- ↳ In SDLC testing starts from requirement stage but here we are testing requirement ideas not coding.
- Q) Define Defect Life cycle / bug life cycle?
- ↳ It is the specific set of states that a bug goes through from discovery to fix.
- Q) In what format will a get builds?
- ↳ In compressed format like:- zip file, Tar file etc.

Alpha Testing

- * The alpha Testing is conducted at developer's place.
- * It is performed in virtual Environment.
- * It is performed before S/W is released to End User.
- * The Developers keeps records of all the Errors & Problems.
- * It involves both Black box and white Box Testing.

Beta Testing

- * The Beta Test Conducted at user's place.
- * It is performed in Real Environment.
- * It is performed after releasing the S/W to End Users.
- * The End users are the Problems & Errors later reports to Developers.
- * It involved only Black box Testing.

Q) GUI Testing :> Graphical User Interface is the process of Testing S/W . User interface against the provided requirements.

Q) What is Test Strategy?

↳ It is a high level document and usually developed by project manager. In this Document we Approach how we go about testing the Product and Achieve the goals.

Q) High Priority Low Severe Severity?

↳ In reyntra help feature help button not working.

Q) Low Priority High Severity?

↳ Recharge a ph no as 100 then Recharge is successful but we do not get any confirmation message.

What is Scrum?

A Scrum is a process for implementing Agile methodology.

If you find bug then what is your next step?

I will first ensure that whether it is a really bug or not and this bug is already found by other test engineer or not. Then I will log into defect tracking tool.

① If there is no company logo in web page then it is Major Defect and Low priority.

② If you have 100 test cases in the 1st cycle 25 failed and 75 passed how many test cases you will pick for the 2nd cycle?

Ans I will pick 100 test cases because to ensure that there should not be any side effect of fixing the bugs.

Q Tell me how you will perform stress testing of a Bike?

① Try to run it at its highest speed when in 4th gear.

② Try to run it at its highest speed when bike in 1st gear.

③ Try to run it at highest speed when tyre pressure is very low.

④ Try to run when there is no engine oil.

Q) What should be the goal of SW Test Engineer?
Ans The goal of SW test Engineer should be to find the maximum bug; And fix them as early as possible and make sure that they get fixed.

Q) what do you understand by HTTP & HTTPS?

HTTP → Hyper text Transfer Protocol
Ans It is unsecured and Operates at application Layer.

HTTPS → Hyper text Transfer Protocol Secure.

Ans It is secured and Operates at Transport Layer.

URL → Uniform Resource Locator
www.google.com wide Web
Latest Android version → Android - 12

Cookies → It is a file Created by website you visit. They Make your online experience easier by Saving Browsing information.

Cache → Cache is a thing that is hidden or stored somewhere that from Computer Memory.

Bug Leakage

When end users And met the bug and this bug was not catched by QA team.

~~Test~~ →
should
be done
once

bug fix

bug fix → When we know the bug is present so when we fix it after knowingly we will not be cause the priority will be low because the priority will be low and severity that bug is low.

Defect Triage / Bug Triage

TTP 28

Protocol
is not
done

Analyze the Bug
Assign priority for it
Assign severity for it
Assign the bug to bug owner.

Test Harness

Test Harness is the collection of stubs and drivers designed to automate test execution.
→ at test script points → Test execution
and test script repository.

Test Coverage
→ coverage all the possible test scenarios of a coverage cell the possible has been measured how much testing has been done on a particular piece of software.

How to achieve test coverage?
→ achieving good requirement traceability by establishing good requirement traceability.

Recover Testing ?

→ How quickly the system can recover after the SW crash or hardware failure is called Recover Testing.

Design based UI and UX :-

UX Design :→ User Experience design → user

UI Design :→ User Interface Design → controls

Q what is the Diff b/w Project Based Testing & Product Based Testing?

Project Based is nothing but client requirement . product Based is nothing but market requirement .

* if there is no sufficient time for Test
as you have to complete the testing then
what will we do ?

- ① Smoke testing
- ② Usability Testing
- ③ walk through with Product

Test Data

Test Data is the Data is used by test engineer to execute the test case .
It can prepared by manually and also using tools .

Q What are the Advantages and Drawbacks of Manual Testing ?

Disadvantage / Drawbacks :-

- > It requires more time than Automation testing .
- > It can create susceptible human errors .
- > It is time consuming to maintain test cases .
- > Test Engineers need to know the product well .

Advantage

- > Uses human intelligence to find errors .
- > It helps to maintain a testable software .
- > Find errors outside of the code .
- > Provides accurate and user experience testing this is helpful for user .

> Test Engineer can focus more on Complex features and functions.

Defect clustering

It states that a small number of modules contain most of the defect detected. It means 80% of defect are caused by 20% of module.

Test closure:

Test closure is a document that gives a summary of all the test conducted during the SW development life cycle. These documents include objective, time taken, total cost, test coverage and any defect found.

Test Scenarios

Lift

- ① Verify the dimension of the lift (F/T)
- ② Verify the ^{types of} door specification as per the specification.
- ③ Verify the types of metal used as interior & exterior.
- ④ Verify the capacity of load in terms of total weight. (P.T)
- ⑤ Verify the lift moves to the particular floor click on the number (S.T)
- ⑥ To verify the lift door will be properly closed or not (I.T)
- ⑦ Verify there is available emergency buttons
- ⑧ Verify light are available in the lift.
- ⑨ To verify that the test weather is the lift button as per the country standard or not (G/T)
- ⑩ To verify that look of the lift ~~exterior~~ interior & exterior design as per the specification or not. (U/T)
- ⑪ To verify that lift moved on the particular floor click on the number (S.T)
- ⑫ To verify that randomly selected or number other lift is moved to other particular floor (Adhoc/T)

Verification

- > Verifying or testing CBS, SDS, HLD and LLD check whether it is according to the requirement or not is called verification
- > It is done by Test Engineer
- > It is also called as Static Testing
- > It is done before development
- > Here we check ~~as~~ whether we are building right product / right SW / right system.

Validation

- > Testing the functionality of an application by executing test cases is called as validation
- > It is done by TB
- > It is also called as Dynamic Testing
- > ~~also~~ It is also called done after development
- > Here we check ~~as~~ whether ~~we~~ are building right product / right SW / right system.

Sprint

* Sprint is a set of features within product backlog as per the goal & it is part of the end product.

* Each sprint is not a complete project.

Adhoc Testing

* first, We learn about the product and then start the testing.

* Documentation is not the main criteria

* It is the perfection of testing.

* Adhoc testing helps to find innovative ideas from the research.

Retrace

* Retrace is the complete features set for the particular version of the product.

* It is long term part of the end product so each retrace can be an endpoint to other project.

Exploratory Testing

* We learn and test the product simultaneously.

* Documentation is the main criteria.

* It is the learning of the application.

* It helps to develop the application.

Principles of Software Testing :-

- 1) Detect clustering
- 2) Pesticide Paradox
- 3) Context Based testing
- 4) Early Testing
- 5) No-exhaustive Testing
- 6) bug-free software not possible
- 7) Testing to find Defect

Pesticide Paradox

It is a condition which we are unable to find new defects and hence we need to change our approach towards testing the application.

Ex Moving from Functional testing to non-functional testing or changing the test scenarios to test More edge scenarios.

Context Based Testing

The Context based testing is a model on the context of the project rather than going by testing methodology.

Ex It recommends test Engineer to check their testing technique , Test Deliverables , Test Documentation and test objectives by looking into the ^{context} details of a specific situation. → ^{on} Dynamic Testing

Q) Levels of SW Testing ?

- Unit Testing
- Integration Testing
- System Testing
- Acceptance Testing

Q) Why you go for Functional Testing first?
→ Suppose you are testing with multiple users. for example 1000 users but there is no point of testing code 1000 users if that particular feature is not working for one single user.

Q) Various static testing Techniques?

- 1) Informal reviews
- 2) Technical Reviews
- 3) Walkthrough
- 4) Inspection
- 5) Static Code Review

Q) Various Dynamic Testing Techniques?

- Test Case Design
- Test Environment Setup
- Test Case Execution
- Test Analysis

Q) What are some quality strategies of Agile Models?

- 1) Iteration
- 2) Re-factoring
- 3) Dynamic Code Analysis
- 4) Short Feedback Loops
- 5) Reviews and Inspection
- 6) Standards and guidelines
- 7) Milestone Review

Q) Disadvantages of Agile Model

- i) New requirement may be conflict with the existing requirement.
- ii) changes are there with feature correction and change the project will cross the expected time.
- iii) There may be difficult to estimate the final cost of the project due to constant interaction.
- (iv) A Detailed requirement is absent.

Q) Scrum Ban / Kanban

Scrum Ban is a Scrum and Kanban based model for SW Development. The model is used in the project that need continuous maintenance, various program errors, or some sudden changes.

Q) What are the major principle of Agile Testing?

- Customer satisfaction
- Face to Face communication
- Sustainable Development
- Continuous Feedback
- Quick respond to changes.
- Successive improvement
- well-Organized
- focus on efficiency
- Error free clear mode
- Collective work

Q) What do you understand by the term Agile Testing?

- i) The agile Testing is the SW Testing process which is fully Based on the principle of Agile software Development.
- ii) It is the Iterative approach - where the user story becomes the output of the collaboration betw. the product owner and development team.

Q) Agile framework names:

- Scrum, Kanban, Feature, driver development Test, driver development

- Q) Can we use waterfall over Scrum?
→ Yes, sometimes we use when the customer req. is simple, small, well defined and does not change until the project complete.
- Q) What are the skills of good Agile Tester?
→ He must be familiar with Principles and concept of Agile.
→ He must be excellent communication skills to communicate with the team & Client.
→ He can set the priority of a task according to customer requirement.
→ He should able to understand the customer requirement properly.
→ He should understand the project risk due to changing demand.

- Q) What are the objectives of SW Testing?
→ SW testing helps in preventing defects.
→ SW testing helps in finding defects.
→ SW " builds confidence.
→ " " reduces risks.
→ " " determines if the SW fit for end user or not.

- Q) What is Orange Testing?
→ Orange Testing is done when the SW is ready for release with specified requirement. It is done at the client place.

- Q) Scalability Testing?
→ It is a type of non-functional testing. It is to determine how the application under test scales with increasing workload.

Q) Security Testing :-

→ Security Testing is a process to determine whether the system protect data and maintain functionality as intended.

Q) Defect Cascading :-

→ It is a process in which one defect triggers another one and due to this it becomes difficult to find root cause of the defect.

Q) Test metric

give a measure of total number of tests created and executed.

it includes :-)
total Test
test passed
test fail
test Deferred.