

Regression Exploratory Testing

Understand the application or software identify all possible scenarios, document scenarios and test the application by referring the document is called as Exploratory testing.

OR

Explore the application, understand how each and every feature work, based on your understanding, test the application. This is called as Exploratory testing.

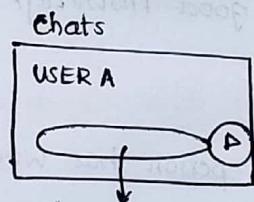
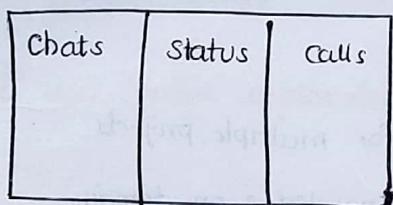
TE will do exploratory testing when there is no requirement or when req are missing or when TE are unaware about the requirement.

Why Requirement will be missing.

1. Chances are there, if the proj is very old project for old features, req might be missing.

How to do exploratory Testing.

eg: whatsapp. [No requirement]



- chats to user A, create a group add the users B, C, D, change the group icon, Exit group, video call

- ST [Exploratory testing].

• FT [Exploratory testing]
(+ve and -ve)

1. I will understand the application and enter all possible inputs for each and every component and do exploratory testing.

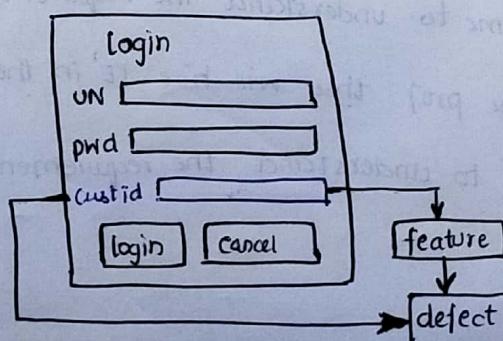
2. I will explore the application, understand how each and every feature works and test dataflow by doing exploratory testing.

3. I will understand the application and try to cover all the end-to-end scenarios by doing exploratory testing.

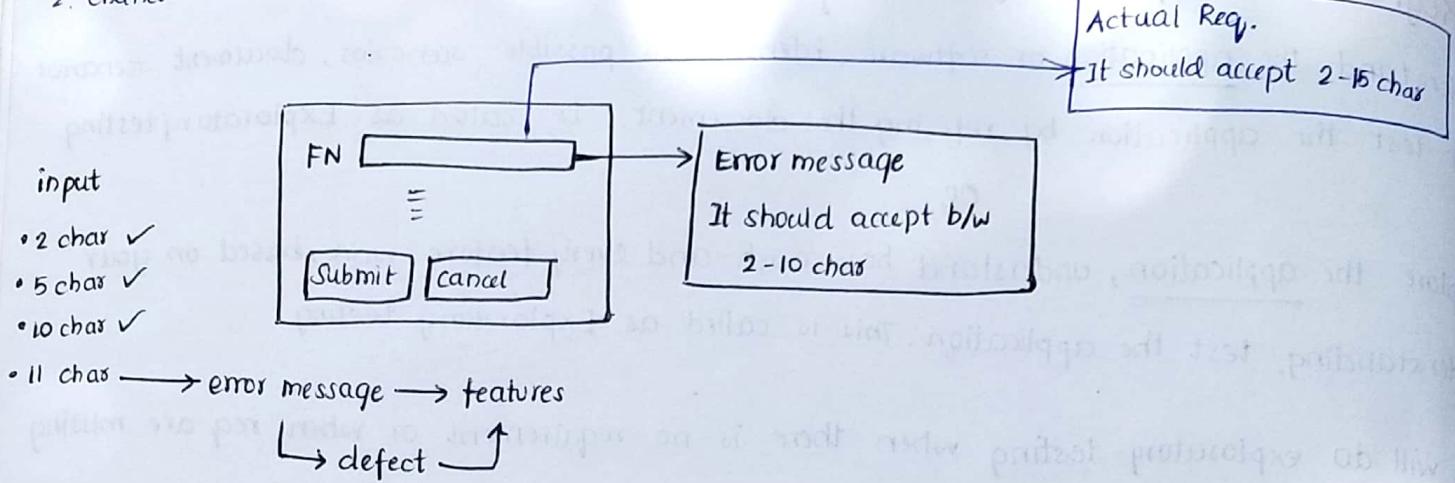
Drawbacks of Exploratory Testing.

1. Chances are there we might misunderstand feature as defect

www.citibank.com



2. Chances are there we might misunderstand defect as feature.



3. If there is a feature missing we will never get to know really the feature is missing or not.

4. Time taken is more because we should spend more time in understanding the application.

How to overcome drawbacks.

1. TE will interact with senior TE, senior Dev, BA, customers.

2. Based on product knowledge. [If a person has worked on the same project for couple of years and has got very good knowledge on project, then we can tell he is a product expert].

3. Based on domain knowledge [If a person has worked for multiple projects which belongs to same domain and has got very good knowledge on domain then we can say he is a very good domain].

4. By using common sense

Eg: click on signup → go to sign up page

click on login → go to login page

5. By comparing our application with similar kind of application. we can overcome the drawbacks.

When to go for Exploratory testing.

1. When there is no requirement.

2. There is a requirement, there is no time to understand the requirement.

[Some companies for some of the proj they will hire TE in the mid of project and will not give sufficient time to understand the requirement]

3. There is a requirement, but it is not understandable ie, it is very complex
eg: Requirements of health sector.

Why we should do exploratory testing when there is no requirement.

1. By doing exploratory testing TE will find all the blocker defects and we might miss some minor defects and the software is given to customer, customer can use software and run business without finding any blocker problems. So, when req is not there, instead of releasing software to cust without testing, it is always best to do exploratory testing.

? How do you test the product when there is no requirement.

I will understand the ~~requirement~~ application and do exploratory testing. Chances are there I might misunderstand feature as defect, and defect as feature so to overcome this, I'll communicate with sr. Dev, TE, BA, cust and based on my product knowledge I will test the application.

? Do you write exploratory scenarios and test cases

Yes.

? What is the difference between smoke, Adhoc and Exploratory.

	Smoke Testing	Adhoc Testing	Exploratory Testing
What	Testing the basic and critical features of an application or software before doing thorough or rigorous testing is called Smoke testing	Testing the software or application randomly is called Adhoc testing	understanding the application, identifying all possible scenarios, document all the scenarios and test it by referring the scenarios is called Exploratory testing
When	As soon as we get a new build, we do smoke testing	When TE get creative ideas we can do adhoc testing	No requirement
Why	When software is testable or not	customer will use the software randomly, chances are there cust might find defect so do Adhoc T	No requirement
Types	Formal , Informal	Buddy, pair, monkey	No types.
Requirement /-ve	Yes	Yes	No
you w req	Positive	Negative	Positive and Negative
	Yes	No	No

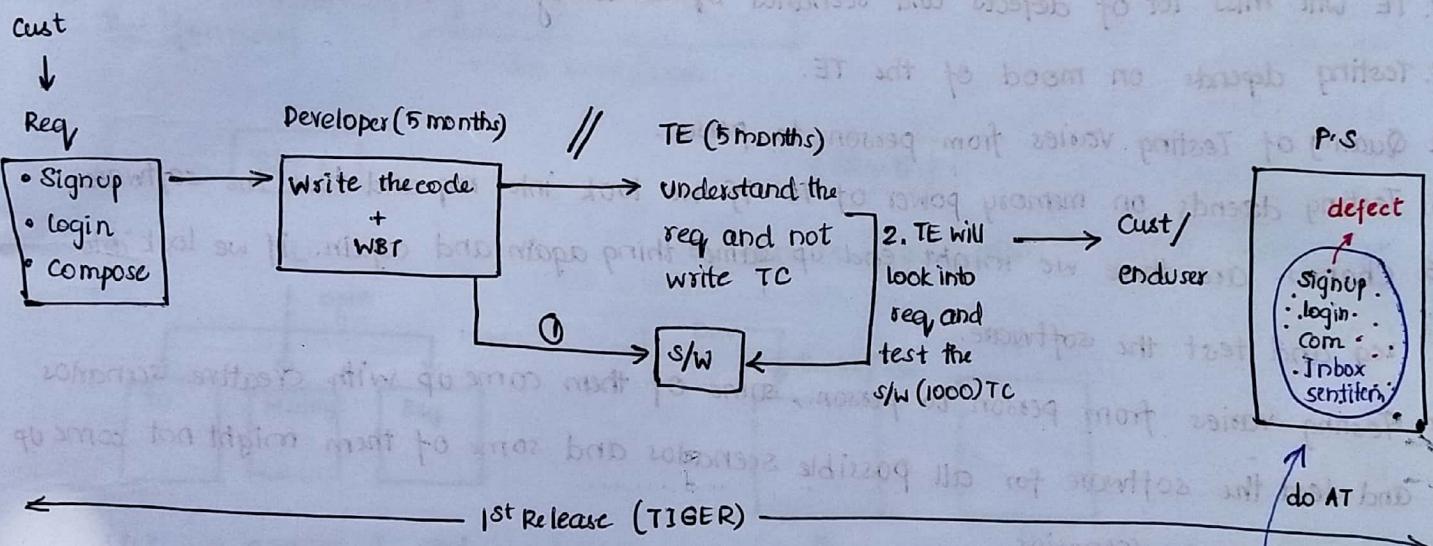
Migration Testing.

When the application is migrated from 1 technology to other technology, we will do Migration testing, to make sure that newly build application is working for new customer data and old customer data and also we make sure that all the end to end customer business workflow is working as expected or not is called as Migration Testing.

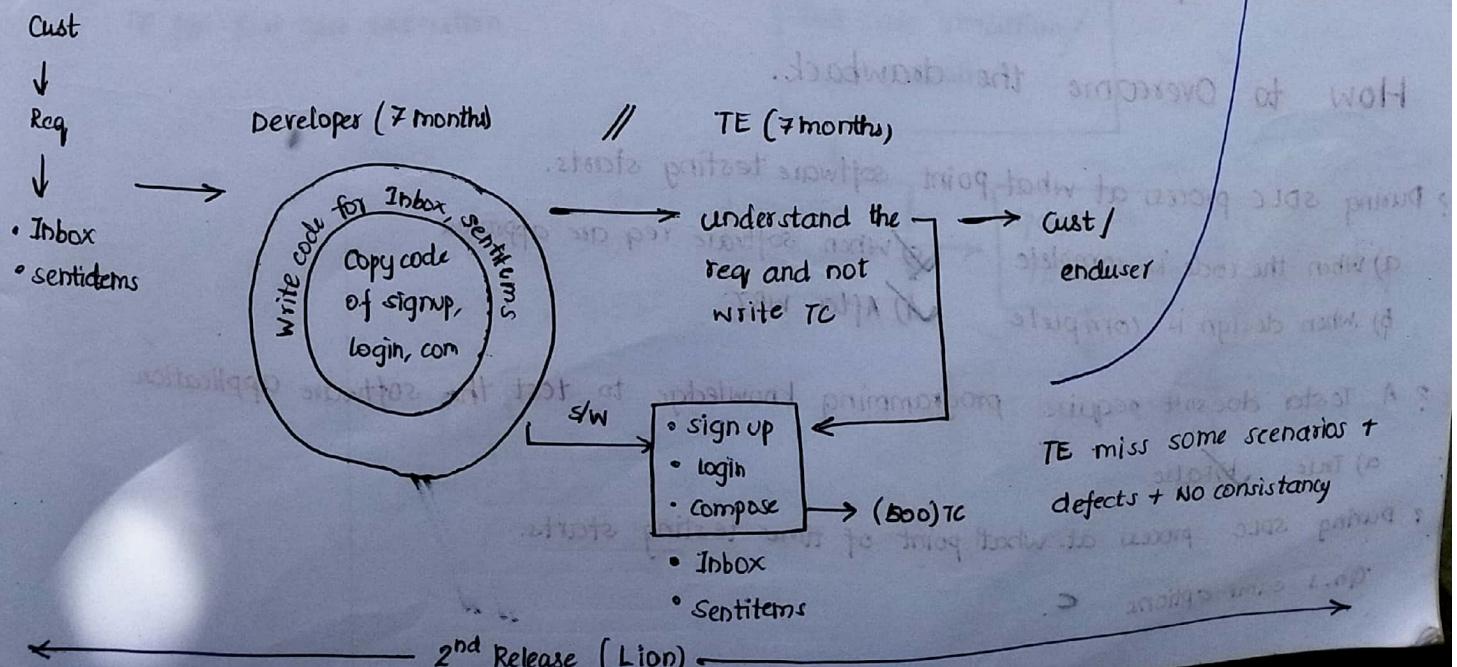
Test case [80% for IQ]

Test scenarios are high level documentation for all the customer business workflow according to the requirement is called Test Scenarios.

Release 1 (1 year)



Release 2 (1.5 years)



What are the drawbacks of not writing test cases OR

What will happen if you look into requirement and test the software

1. There will no consistency in testing if you look into requirement and test the software

WhatsApp

1st Release

Chats → Dev → TE → will not write TC

↳ Chats (500 TC)

2nd Release

Status → Dev → TE → will not write TC

↳ Chats [old - 150 TC]
↳ Status

→ No consistency

2. TE will miss lot of defects and scenarios b/f looking into requirement and test the app.

3. Testing depends on mood of the TE.

4. Quality of Testing varies from person to person.

5. Testing depends on memory power of TE. If we look into req and test the software

6. Chances are there we might end up same thing again and again. If we look into req and test the software.

7. Testing varies from person to person. Some of them come up with creative scenarios and test the software for all possible scenarios and some of them might not come with creative scenarios.

8. Test case coverage will not be good.

How to Overcome the drawback.

? During SDLC process at what point software testing starts.

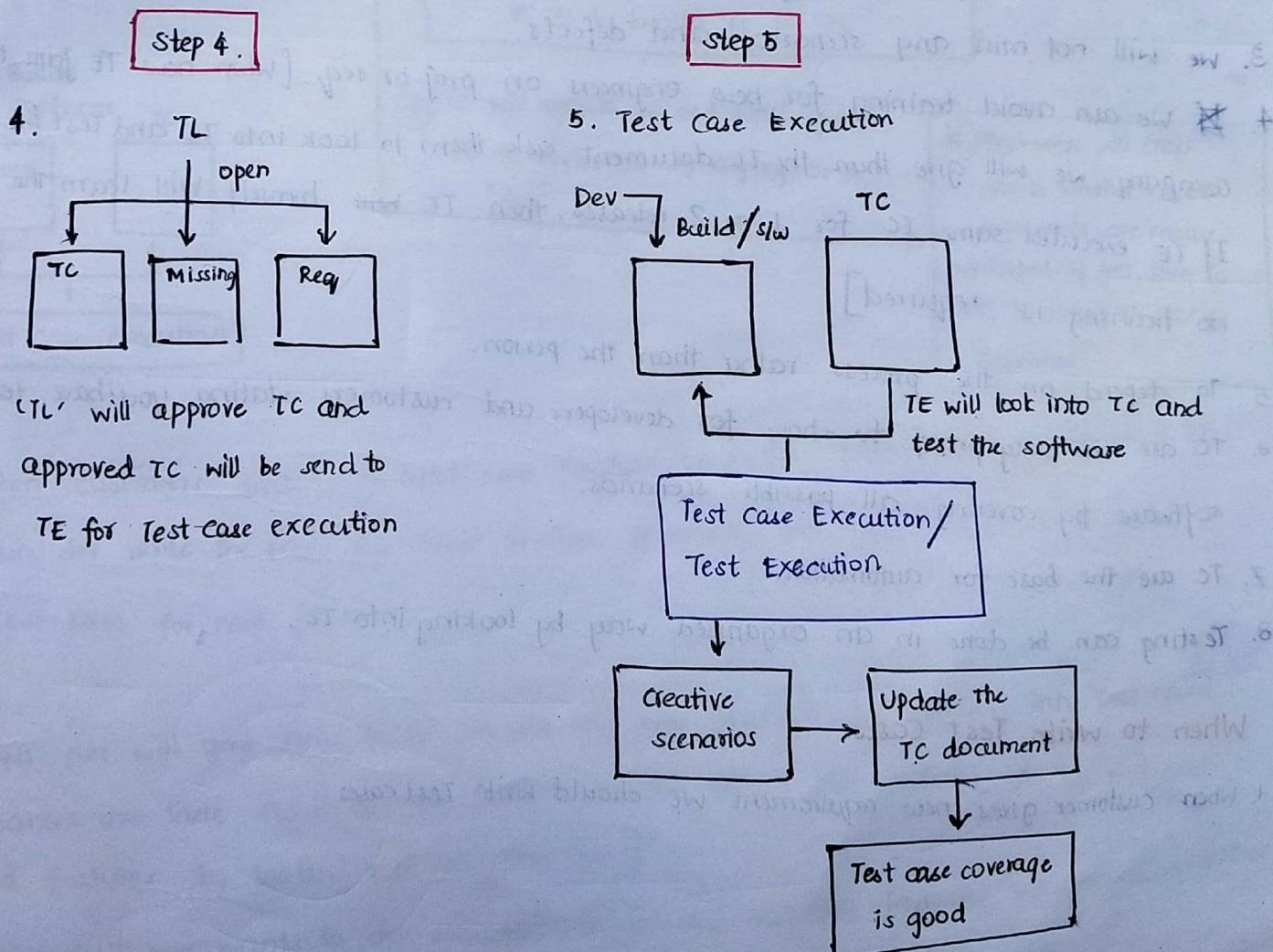
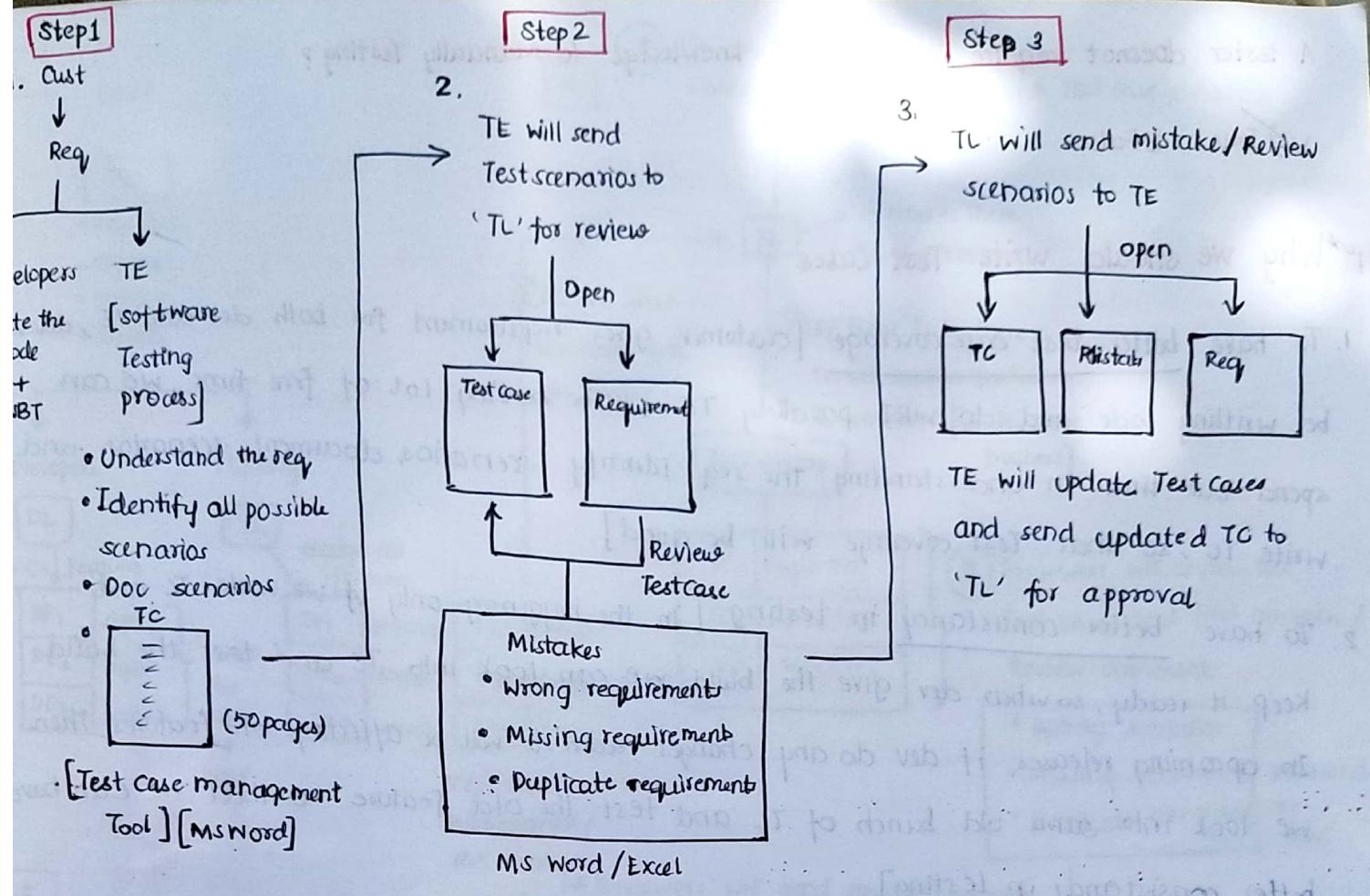
- a) When the code is complete ✓ When software req are approved
- b) When design is complete ✓ After WBT

? A Tester doesn't require programming knowledge to test the software application.

- a) True ✓
- b) False

? During SDLC process at what point of time testing starts.

- Qn. 1 same options c.



A tester does not require programming knowledge to Manually Testing?

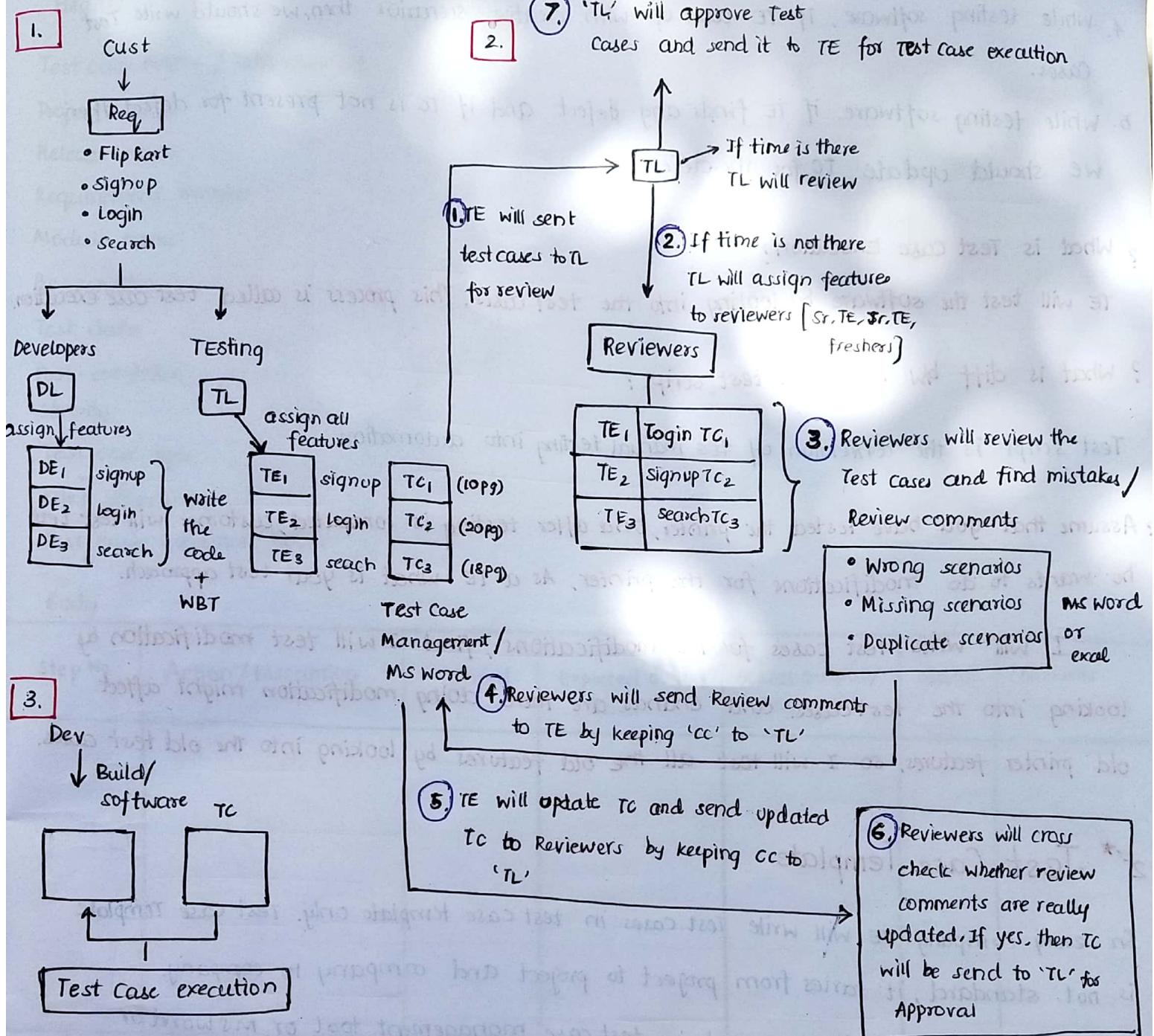
- a) True b) False

Q1* Why we should write Test Cases.

1. To have better test case coverage [customer gives requirement for both dev and TE, dev be writing code and do WBT parallelly TE will be having lot of free time, we can spend our time in understanding the req, identify scenarios, document scenarios and write TC, so that Test coverage will be good].
2. To have better consistency in testing. [In the beginning only if we write TC and keep it ready, so when dev give the build, we can look into TC and test the build. In upcoming releases if dev do any changes, add it will be affecting old features, then we look into same old bunch of TC and test the old feature so that we can have better consistency in testing].
3. We will not miss any scenarios and defects.
4. ~~We can avoid training for new engineers on proj or req.~~ [When new TE join the company we will give them the TC document ask them to look into TC and test the software. If TE executes same TC for 1 or 2 releases, then TE will himself will learn the project no training is required].
5. To depend on the process rather than the person.
6. TC are the only proof to show for developers and customers stating you have tested software by covering all possible scenarios.
7. TC are the base for automation.
8. Testing can be done in an organised way by looking into TC.

When to write Test Cases.

1. When customer gives new requirement we should write Test cases



2. When customers wants to add new feature or extra feature, we should write TC.

[When dev write the code for new feature parallelly Test engineers will write the Test case for new feature.]

When dev will give new build, TE will test new feature by looking into Test cases, chances are there adding new feature might affect old features, so TE will test old features by looking into old Test cases]

3. When customer wants to do modification on the existing features we should write Test cases.

4. While testing software, if TE come up with creative scenarios then, we should write Test Cases.
5. While testing software, if TE finds any defect and if TC is not present for defect then we should update TC for the defect.

? What is Test Case Execution?

TE will test the software by looking into the test cases. This process is called test case execution.

? What is diff b/w TS, TC and test script?

Test script is the conversion of manual testing into automation.

? Assume that you have tested the printer, once after testing is completed, customer will test he wants to do modifications for the printer. As a TE what is your test approach.

I will write test cases for the modifications, first I will test modification by looking into the testcases. and chances are there doing modification might affect old printer features, so I will test all the old features by looking into the old test cases.

2^{cr}* Test Case Template

In every company TE will write Test cases in test case template only. Test case template is not standard, it varies from project to project and company to company.

Test case template will be prepared in test case management tool or MS Word or MS Excel.

Header

Test Case Name / Test case Id :

Project Name :

Release Name :

Requirement Number :

Module name :

Pre-condition :

Test data :

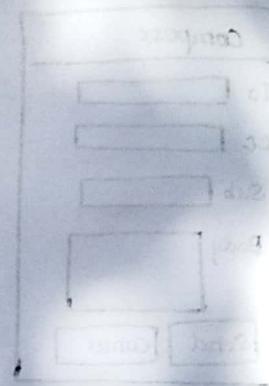
Post-condition :

Severity :

Test case type :

Brief description :

Test case execution hours :



Body

Step No	Action / Description	Input	Expected output / Result	Actual Output / Result	Status	Comments
1	Initial setup and configuration.	Not specified	Not specified	Not specified	Not specified	Not specified

Before Test case execution

After test case execution

Footer

Author :

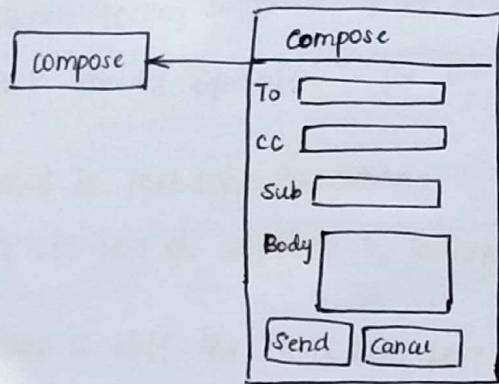
Reviewed by :

Approved by :

Approval date:

Requirements

gmail.com



Test case name :

Project name

Release name

Requirement number :

Module name

Pre-conditions

Test data

Post-conditions

Severity

Test case type

Brief description

Test case execution hours

Step no	Action / Description	Input	ER	AR	Status	Comment
1.	Open browser and enter URL	www.gmail.com	'login' pg should be displayed	Blank page	Fail	defect
2.	Enter 'UN' and 'pwd' click on login button	UN: DJINGA pwd: DJINGA	'Home' pg should be displayed	'Home' pg is displayed	Pass	NA
3.	Click on 'compose' button	NA	'compose' pg should be displayed with following fields 'To' textfield 'cc' text field 'sub' text field 'Body' Text area field 'Send' button 'cancel' button	Blank page	Fail	defect
4.	Click on 'logout' button	NA	User should be successfully loged out from the application			

Author:

Reviewed by :

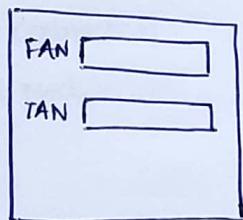
Approved by :

Approval date:

Test Case name:

TE will write unique test case name for every individual TC [test case id will be automatically generated by Test case Management tool].

e.g. Citibank. Amount transfer



Project name - Module name - scenario

- CBO - Amount transfer - FAN textfield should accept 10 digit
- CBO - Amount transfer - TAN textfield should accept 10 digit

Release Name

For every test cases TE should write Release name so that in feature, we will get know in which release how many TC were written.

In the company release name will be given or decide by customer, PM, development manager, Test manager, BA, they will give the Release names.

e.g. customer, BA, DM, TM

- CAT 2019-1 Acti-01 Agile 1.0
- MAT 2019-2 Acti-02 Agile 1.1
- RAT 2019-3 Acti-03 Agile 1.2

Requirement Number

When BA convert CRS to SRS in the SRS, BA will mention requirement number for every modules and components so TE should copy past requirement number into test case template.

e.g. RAN: Req no : 30.1.1

1st Pre-condition

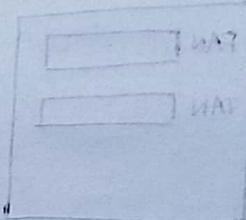
Pre-conditions in the settings or actions that should be done before executing test case or step no.1.

Check for below mentioned settings.

- check whether software is installed or not
- check for internet connection
- Browser settings

[Tools → Internet option → security → customer level]

- Disable excel download
- Enable excel download
- Enable pdf
- disable pdf
- Disable Java script



- check for roles and permission
- Check for UN and pwd.

Test data / Test Harness

It is the data created by TE by using the application before executing TC or step no.1.

Test data can be created manually for less number of data, we can also use automation tool to create test data for more number of users.

e.g: gmail.com [scenario 1]

Login as user A, compose a mail send a mail click on sent items and check whether mail is displayed.

Pre-condition

Check for internet connection.

Test data

create user A account

scenario 2

Login as userA click on inbox select 10 mails, delete mail click Trash check whether mails are displayed. in Trash.

Precondition

1. Check for internet connection
2. Check for UN and pwd.

Test data

Create 10 mails should be present in inbox.

Scenario 3.

Login to userA click on compose enter values for all the fields click on cancel, click on draft, select a mail, click on sent, click on sent items check whether mail is displayed again; compose a mail send a mail to user C .click on Trash, select a mail, empty the trash, logout .

Precondition

check for internet connection

check for UN and pwd

Test data

create user C account

mail should be in trash.

Acti time

Login as admin delete a TL , create New TL Logout as admin, login as TL delete the existing customer. create a new customer and create a new project to customer add existing task for the project and create new task for project assigned task to existing TE .create new TE assign one more task to the new TE, logout as TL.

Pre-condition

Check whether software is installed or not

check UN and pwd for admin

check for roles and permission

Test data

One test Lead should be present

One customer should be present

One task should be present

One TE should be present

Flipkart

Scenario

Login to flipkart app remove product from cart, click on wishlist, select one product in wishlist and move to cart. Click on cart go to the product buy the product by using debit card payment option click on order, check for the order details

Pre condition

Check for software installed or not

Check for internet connection

Check for UN and pwd [If Account is present]

Debit card details should be ready

Test data

Check for user account [If account is present]

One product should be in cart

One product should be in wishlist.

Whatsapp

Login to whatsapp remove profile picture, unblock friend / unblock contact, chat with same contact delete the chat close the whatsapp.

Pre condition

Check whether software installed or not

Check whether the internet condition for

Test data

Profile should be present

One contact should be blocked.

Assignment - Write precondition and Test data for 10 scenarios.

Past Condition

Past condition will consist of expected result [don't write functional testing expected result in past condition].

e.g: gmail.com

scenario.

Login as user A compose a mail and save it as draft and click on draft check whether mail is displayed.

Past condition

Mail should be present in draft.

Severity

TE will give severity for every individual test cases based on how important and complex.

The features are from customer business point of view.

There are 3 types of severity.

- Critical
- Major
- Minor.

TE will execute test case based on the security.

Features	Testcases	Security	Execution Pattern
Login	TC ₁	Critical	TC ₁
Inbox	TC ₂	Critical	TC ₃
Compose	TC ₃	Critical	TC ₂
Draft	TC ₄	Major	TC ₄
Trash	TC ₅	Minor	TC ₅

Manager Round.

Assume that you have one day time and there are 20 TC to execute every TC It will take 1 hour time. As a TE How do you test the software and ensure the quality?

I will 1st executes all the critical test cases which are very important from customer business point of view, Later I execute major TC, if time is permit, I execute minor TC if required I will extend my working hours and try to complete the test execution.

Brief Description

This section describes the company behaviour of the TC

e.g: Bank

FAN

Brief

In amount transfer module, FAN textfield should accept 10 digit account no. and account should be created by manager.

Test case execution hours

Test case execution hours = time taken to write TC + execute TC

$$\begin{aligned} &= 60 \text{ hrs} + 30 \text{ hrs} \\ &= 90 \text{ hrs.} \end{aligned}$$

Body

Step no

TE should give unique step no for every individual steps.

Action / Description

It consist of all the navigation steps. → how to use app

Input

While testing whatever data will be entered into the software, all the data should be mentioned in input [URL, UN, pwd, positive, Negative input]

Expected Result

TE will write expected result by looking into the requirement before test execution

Actual Result

TE will write actual result after test execution by looking into the software or application

Status

There are 3 types of status.

1. Pass [Actual result is same expected result]
2. Fail If actual result not matching with expected.
3. Not executed If not executed.

Comments

If the status is fail then we have to give reason in comment section stating by test case fail.

Author

A person or TE one who write the test case

Reviewed by

A TE one who review test cases is called reviewed by

Approved by

A person one who approved TC is called approved by

Approval Date

The date of TC approval date

? Which is the key element in the TC template ? Expected Result.

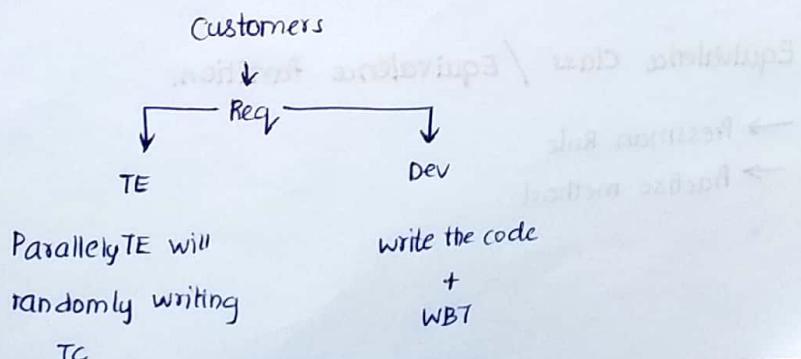
? What is the Test basis? set of information required in order to test the software.

Test case Design Techniques / Testing techniques / Design Techniques / Black box Techniques

TC design techniques are the techniques which are applied while writing TC in order to improve the test case coverage

Drawbacks

1. Miss lot of scenarios
2. Miss lot of defects.
3. There will not be good coverage.



Types of Case design Techniques.

- 1) Error Guessing
- 2) Equivalence class or equivalence Partitioning
- 3) Boundary value Analysis [BVA]
- 4) Decision table technique
- 5) State transition Diagram / Technique

Error guessing.

Here TE will guess the error and derive more no.of scenarios. Error guessing depends on analytical thinking of every individual TE. Here TE will enter ONLY negative values or invalid values and try to guess the error messages. This is called error guessing.

Amount Req: Should accept only the integers.

Invalid / -ve input	Error
Hundred	Error msg
100R5	Error msg
* \$ / #	Error msg
110.726	Error msg
Blank	Error msg

Error guessing is best used

- a) As the first approach to deriving TC
- b) After more formal techniques have been applied
- c) By inexperienced testers.
- d) After the system has gone live
- e) Only by end users.

Equivalence class / Equivalence Partition.

- Pressman Rule
- Practise method.

Pressman Rule

Rule 1.

If the input is range of values then designed TC for 1 valid and 2 invalid inputs

Amount

Req: should accept between 100-5000

Test cases → 2000 (Valid)

→ 90 (invalid)

→ 6000 (invalid)

Rule 2

If the input is set of values then design TC for 1 valid and 2 invalid inputs

Req.

Product	Product_id
Samsung	91
HP pendrive	83
Dell laptop	97

onlineshopping

Search

Test cases

→ 91 (Valid)

→ 80 (Invalid)

→ 99 (Invalid)

It has editable textfields and non-editable textfields. For editable textfield we check for valid and invalid whereas in the case of non-editable textfield we check for only valid. (Because invalid data can't be given as ~~assignment~~ ^{input})

Rule 3

If the input is boolean then designed TC for both truth and false values

sign up

FN
LN
=

Gender OM OF

Drawbacks of Pressman Rule

Tested	
100, 101	1000 - A
1001, 1002	2000 - B
2001, 2002	3000 - C
3001, 3002	4000 - D

Not Testcases - drawback

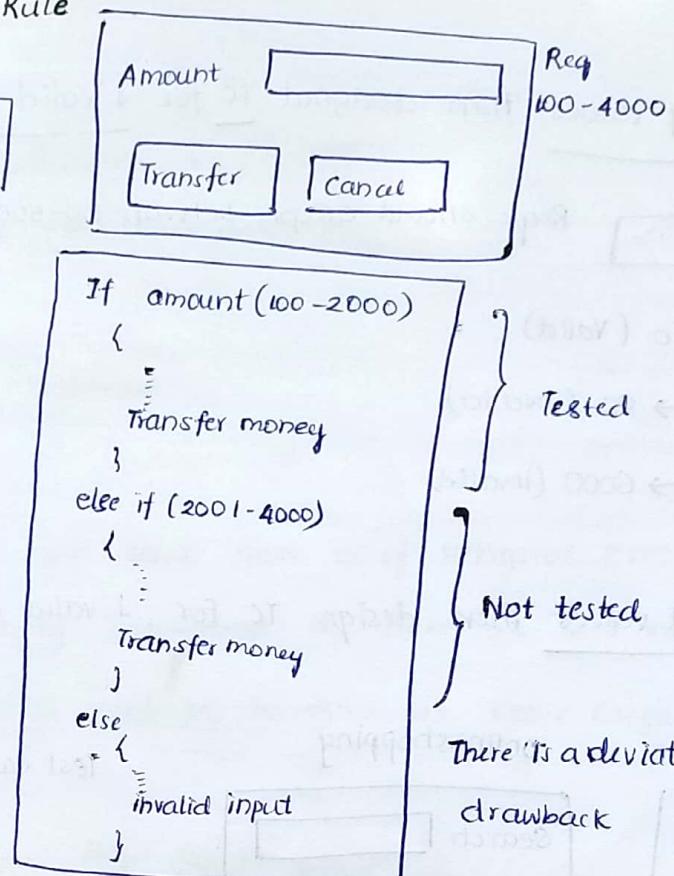
If amt (100 - 4000)

```

{
    :
    Transfer money
}
else
{
    :
    invalid
}

```

There is no deviation



There is a deviation

drawback

In order to overcome drawback of pressman rule, we go for practise method.

Practise Method.

If the input is range of values then divide the range into equivalent parts and test for all those values and make sure that we are testing for atleast 2 invalid values.

Req. = (100 - 5000)

90	-	Invalid
1000	-	Valid
2000	-	Valid
3000	-	Valid
4000	-	Valid
5000	-	Invalid

Boundary Value Analysis (BVA)

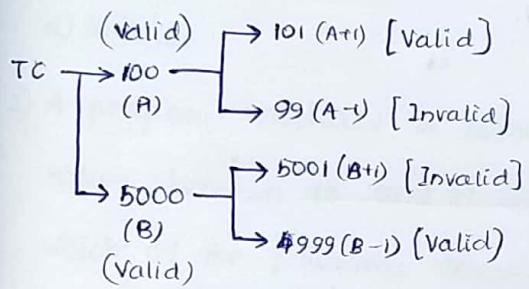
If the input is range of values between A to B, then design TC for A, A+1, A-1 and B, B+1, B-1

e.g: Amount

Req

100 - 5000

(A) - (B)



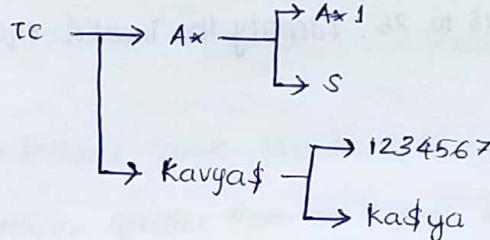
pwd

Req.

(A) min2 - max6 (B)

at least 1 spl char

1 UC.



e.g: Amount Req (100-5000)

If (amount \geq 100 and amount \leq 5000)

{

Transfer money;

}

else

{

Throw error message

}

conditional operator

>, >=, <, <=, =, !=

According to above eg chances of finding example will be more because dev might do some mistakes while writing conditional operators. so, inorder to find this defects TE should write TC by applying BVA.

Test case Optimization

The process of removing duplicate TC in the process of TC is called Test case optimization.

ISTQB.

- 1) Order number on a stock control system can range between 10000 and 99999 inclusive which of the following input might be a result of designing test for only valid equivalence classes and valid boundaries.

a) 1000, 5000, 99999

actual code ($\text{stock} \geq 10,000$) and ($\text{stock} \leq 999,999$)

b) 9999, 50000, 100000

dev might write

($\text{stock} \leq 10,000$) and ($\text{stock} \geq 9,99,999$)

~~c) 10000, 50000, 99999~~

d) 10000, 99999

e) 9999, 10000, 50000, 99999, 100000

- 2) One of the fields on a form contains a text box which accepts numeric values in the range of 18 to 25. Identify the invalid equivalence class.

~~a) 17~~

b) 19

c) 24

d) 21.

- 3) In an examination, a candidate has to score a min of 24 marks in order to clear the exam. The maximum that he can score is 40 marks. Identify the valid equivalence values if the student clears the exam.

a) 22, 23, 26 ~~✓ 29, 30, 31~~

b) 21, 29, 40 d) 0, 15, 22

- 4) One of the fields on a form contains a text box which accepts alphanumeric values, Identify the valid equivalence class.

a) BOOK ~~✓ Boo01K~~

b) Book ~~✓ Boo k~~

- 5) The switch is switched off once the temperature falls below 18 and then it is turned on when the temperature is more than 21. When the temperature is more than 21, Identify the equivalence values which belong to the same class

a) 12, 16, 22

b) 24, 27, 17

c) 22, 23, 24

d) 14, 15, 19

1st class
(12-17), 2nd class
(18-21), 3rd class
(22-27)

ISTQB Qn solve -

ISTQB Exam Questions on Equivalence partitioning and Boundary

How to test login page OR Write Test scenario's for login page.

1) A program validates a numeric field as follows values less than 10 are rejected, values between 10 and 21 are accepted, values greater than or equal to 22 are rejected, which of the following input values cover all of the equivalence partitions?

- a) 10, 11, 21
- b) 3, 20, 21
- c) 3, 10, 22
- d) 10, 21, 22

	1 st class	2 nd class	3 rd class
9 - 1		10 - 21	22 - ..
Invalid	Valid	invalid	

2) A program validates a numeric field as follows value less than 10 are rejected, values between 10 and 21 are accepted, values greater than or equal to 22 are rejected which of the following covers the most boundary value?

- a) 9, 10, 11, 22
- b) 9, 10, 21, 22
- c) 10, 11, 21, 22
- d) 10, 11, 20, 21

9, 8, 7 10 - 21 22, 23 ..

3) Given the following specification, which of the following values for age are in the same equivalence partition? If you are less than 18, you are too young to be insured, Between 18 and 30 inclusive, you will receive a 20% discount. Any one over 30 is not eligible for a discount?

- a) 17, 18, 19
- b) 20, 30, 31
- c) 18, 29, 30
- d) 17, 29, 31

4) In a system designed to work out the tax to be paid, An employee has 4000 of rate tax-free. The next 1500 is taxed at 10%, The next 28,000 after that is taxed at 22%. Any further amount is taxed at 40%. To the nearest whole pound, which of these is a valid BVA . Test case?

- a) 28000
- b) 33501
- c) 32001
- d) 1500

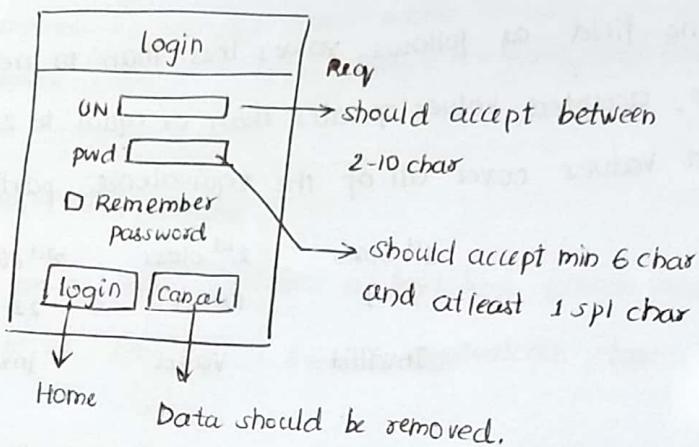
<4000 - notax
4001 - 5500 - 10%
5501 - 33500 - 20% >33501 - 22%

5) In a system designed to work out the tax to paid, An employee has 4000 to salary tax free. The next 1500 is taxed at 10%. The next 28000 after that is taxed at 22%. Any further amount is taxed at 40%. To the nearest whole pound, which of these groups of numbers fall into 3 different equivalence classes?

- a) 4000, 5000, 5500
- b) 32001, 34000, 36500
- c) 28000, 28001, 32001
- d) 4000, 4200, 5600

1) Ask for the requirement.

e.g: gmail.com



If question says write test scenarios then always start with business workflow of customer.

We should write both scenarios and TC.

1. Open the browser, enter URL for gmail, login pg should be displayed. Enter valid UN and pwd, click on login, home page should be displayed.
2. Enter URL for gmail, login page should be displayed, enter valid UN and pwd select remember password, click on login, home page should be displayed.
3. To check that when user enters valid UN and pwd into gmail login page and click on cancel, data should be removed.
4. To check that when user enters valid UN and pwd set remember password, click on login and logout from the application and if the user enters the same UN, pwd should be automatically remembered and displayed.
5. Login to gmail in one of the browser, open the other browser, enter the URL for gmail, login page should be displayed. and user should enter UN and pwd. click on login, home page should be displayed.
6. To check that in one of the browser, enter the URL for gmail, login page should be displayed, enter valid UN and pwd, select remember password, click on login, home page should be displayed. Open the other browser, enter URL for gmail, enter the same UN, pwd should not be remembered and displayed.

7. To check that when user enters valid UN and pwd, click on login, logout and if user enters different UN, pwd should not be remembered and displayed. the home page.
8. To check that when user don't enter UN and pwd, click on login, proper error msg should be displayed.
9. To check that when user enters valid User name and invalid password then proper error msg should be displayed.
10. To check that enter valid valid UN and pwd, select remember pwd, click on login, logout, enter the pwd and check whether UN is automatically displayed or not.
11. To check that the user enter valid UN, pwd select remember pwd click on cancel, check whether pwd is automatically remembered and displayed.
12. To check that when user enter valid UN and pwd, select remember pwd, click on login, home page should be displayed, change the pwd, logout, enter the UN and change the pwd, click on login, home pg should be displayed.
13. To check that when user enters valid UN and old pwd, click on login, proper error msg should be displayed.
14. To check that when user enters valid UN and pwd, click on login and immediately click on cancel. and check whether home page is displayed.
15. To check that when user enters valid UN and pwd, click on login, logout go to browser history , click on URL, check whether login page should be displayed.
16. To check that gmail,login page should work for different os and different browser
17. To check that when user enter valid UN and pwd, select remember pwd, click on login, home page should be displayed, change the pwd, logout, enter the UN and check whether old pwd is automatically ~~displayed~~ remembered and displayed.
18. To check that when user enter valid UN and pwd, click on login,home pg should be displayed, copy the URL , logout,~~enter the~~ paste the same URL, login ~~the~~ page should be displayed.
19. To check that when 10,000 users enter URL for gmail at the same time, login page should be displayed within a second.
- ✓20. To check that gmail login page should work for different service providers.
- ✓21. To check that gmail login page should work for different speed of internet.

1) Write 20 Test scenarios for same login page

2) Write 25 Test scenarios for google page →

f) Write Test scenarios for Pen OR How do you test pen

Requirement.

- Black marker pen

- Camlin manufacturer

Scenarios.

1. Take a pen open the cap, write few lines and check whether pen is working or not (smoke)
2. Take a pen and check every components of pen is according to the requirement, (pen cap, body, refill, nib, etc) (functional)
3. Take a pen, open the cap, reinsert the cap and check whether cap is integrated properly to the pen [integration]
4. Take a pen, open the cap, remove the refill, fill the ink to the refill, re-insert refill cap to body, write few lines, close the cap [end to end]
5. Take a pen, ~~through~~ throw the pen, take back the pen, check pen is working or not [Adhoc]
6. Take a pen, open the cap, write on board, floor, wall, clothes, body and check whether pen is working or not. [compatibility].
7. Take a pen, open the cap, write continuously for particular period of time and check whether pen is working or not. [Reliability]
8. Take a pen, put ~~body weight~~ load on the pen and write and check whether pen is working or not [Performance]
9. Take a pen, check for Language, currency, date ~~by changing~~ (globalization)
10. Take a pen, crash the pen, take back same pen, write and check whether pen is working or not. [
11. Take a pen, open the cap, hold and write and check whether pen is comfortable to write. [usability]
12. Take a pen, open the cap, remove the nib, again insert the nib, write and check whether pen is working or not.
13. Take a pen, open the cap, write and go to refill and check whether ink is reduced.

14. Take a pen, remove the refil, ref try to write using refil.
15. Check for the brand of manufacturer
16. Take a pen, check for colour of pen.
17. ~~Take~~ Take a pen, Open the cap, write in any angle, pen should work.
18. Take a pen, check whether pen is working for different temperatures.
19. Take a pen, dip in water, check whether pen is working properly or not.
20. Take a pen, open the cap, fill different colour ink and write and check pen is working or not.

How to test paper or Write Test scenarios for paper.

Req.

- A4 size
- White colour.

- 1) Take a paper write on the paper, by using blue ink and check whether contents are displayed.
- 2) Take a paper, crush the paper and check whether paper get crushed.
- 3) Take a paper, check for size and thickness of the paper.
- 4) Take a paper, write on the paper, turn back the paper, check whether contents are visible.
- 5) Take a paper, dip inside water, takeout the paper, dry it, and check whether user is able to write.
- 6) Take a paper, check for white colour.
- 7) Take a paper, check for brand of the paper.
- 8) Take a paper, write the paper with white ink, check whether its visible.
- 9) Take a paper, put load on the pen and write on paper and check whether paper is torn or not.
- 10) Take a paper, check the weight of the paper.
- 11) Take a paper, give it to a printer, take the printout and check whether contents are displayed properly.
- 12) Take a paper, crush the paper, open the paper, iron the paper and check whether paper get turned into ashes.
- 13) Take a paper, put fire on paper, check whether paper get turned into ashes.
- 14) Take a paper, fold in rocket shape, paper should fly.
- 15) Take a paper, fold in boat shape, paper should be able to float on water.

- 16) Take a paper, write by using pencil, erase it by eraser, check whether the contents are erased.
- 17) Take a paper, cut into 2 pieces by using pieces, check whether paper is cutted into pieces.
- 18) Take a paper, write on paper using ink pen, pour the water, check whether ink spreads.
- 19) Take a paper, check whether it can be recycled.
- 20) Take a paper, write by using candle and put colour and check whether contents are displayed.

Test Cases.

Test case name : Paper - Test case

Project name : Paper

Release name : Lion

Requirement number :

Module name : Paper

Pre-conditions :

Test data :

Post-conditions :

Severity : Critical

Test case type :

Brief description :

Test execution hours :

stepno	Action / Description	Input	ER	AR	Status	Comments
1.	Take a paper, write by using blue ink	NA	Contents should be displayed			
2.	Take a paper and crush the paper	NA	Paper should be crushed			
3.	Take a paper, check for size and thickness	NA	It should be As per the req			

Stepno.	Action / Description	I/p	ER	AR	Status	Comments
4.	Take a paper, write on the paper and turn the paper	NA	contents should not be visible			
5.	Take a paper, dip in water, take out the paper, dry the paper and write on paper	NA	User should be able to write.			
6.	Take a paper, check for the colour	NA	It should be as per req.			
7.	Take a paper, check for the brand	NA	It should be classmate brand.			
8.	Take a paper, write by using white ink	NA	Content should not be visible			
9.	Take a paper, put lead on pen, write on paper	NA	contents should be displayed			
10.	Take a paper, check for the weight	NA	Its per the req.			

Author : Sukanya CR

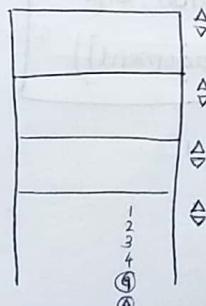
Reviewed by : Vignesh R

Approved by : Pradeep

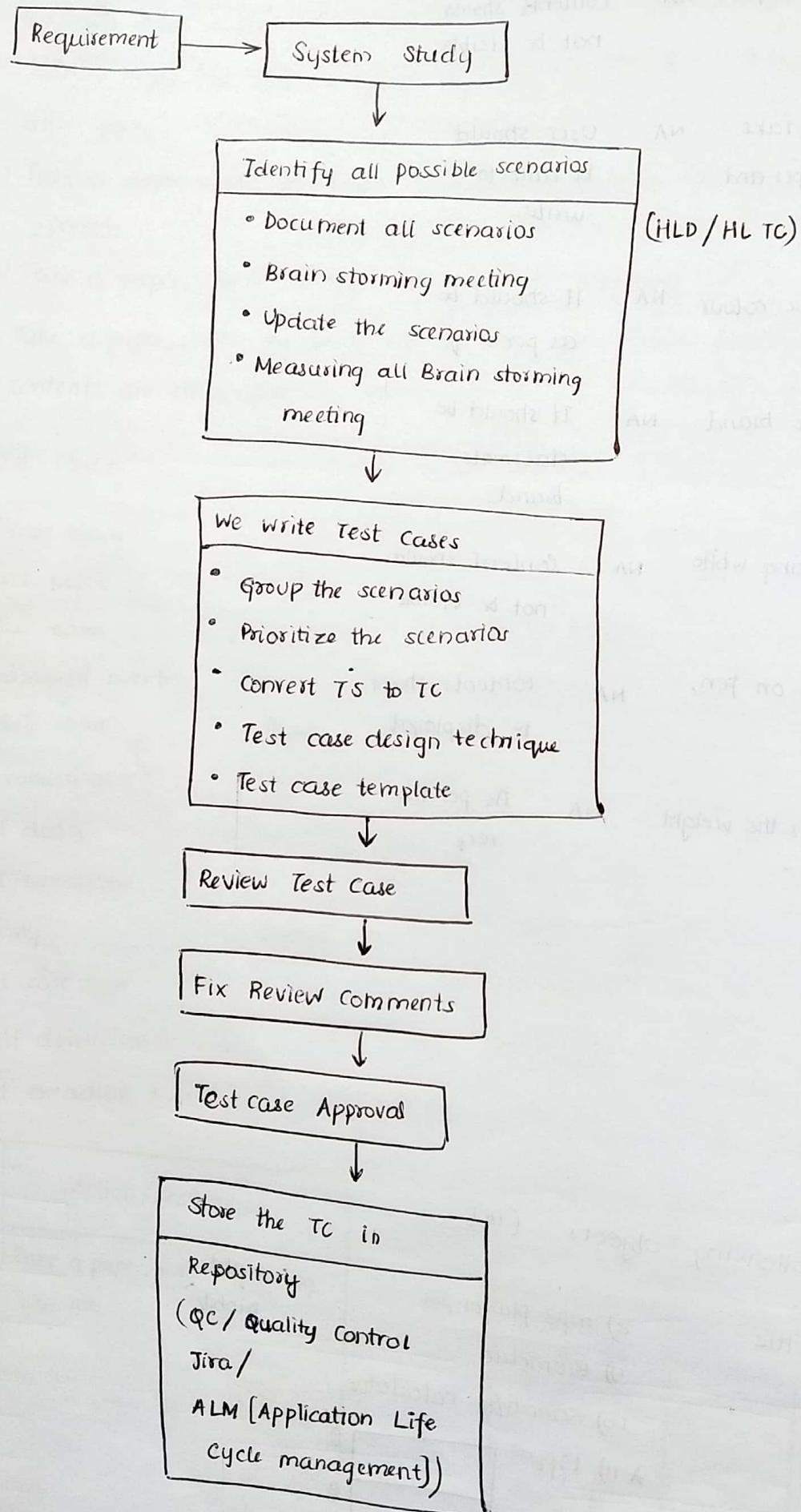
Approval date : 8.7.2019

Write TC and TS for following objects. (15)

- 1) Water bottle - own bottle
- 2) chair
- 3) AC - bluestar brand
- 4) Fan
- *5) coffee mug
- 6) 1 Rupee coin phone
- 7) Triangle
- 8) mp3 player
- 9) Biometric
- 10) scientific calculator
- *11) Lift



Procedure to write the Test case



System study

BA from the company will go to customer place to collect the requirement, he will collect requirement in BL, he will come back to company and will convert to software language. Once the conversion is done, BA will explain the collected req to PM, dev and TE. He will also explain how each and every feature work as a TE we will try to understand the req, If we have any doubts, it will be cleared by the BA. This is called as system study. [Understanding the Requirement].

Identify all possible scenarios

Once after understanding the req, the modules will be assigned to TE. The TE will try to identify all possible scenarios. [FT, IT, ST, smoke, Adhoc] on the allotted modules. Once after identifying the scenarios it will be documented, we can call this document as high level design for testing or high level TC.

Once after documenting the scenarios, the TE will perform Brain storming meeting. In this meeting, we mainly discuss about the scenarios. If we have missed any scenarios or if we have written any wrong scenarios or duplicate scenarios It will be updated in this meeting.

It is a meeting conducted by TE, it will be measured by TL.

Note: Once after done with Brain storming meeting we update our scenarios.

Writing Test Cases.

In this stage, we group all the scenarios based on modules and sub-modules. Once we are done with grouping, we prioritise the scenarios. Once prioritization is done we convert scenarios to TC. with the help of Test Case design technique in TC template.

Once the TC are ready, it will be send for TC review

Here we check the quality of the TC, here we identify the defects in the TC. (defects like wrong, missing and duplicate scenarios or wrong, missing and duplicate TC)

These defects will be communicated the author as review comments. The Author fix the comments. Again a re-review will be done by the reviewer to check whether the review comments are fixed or not.

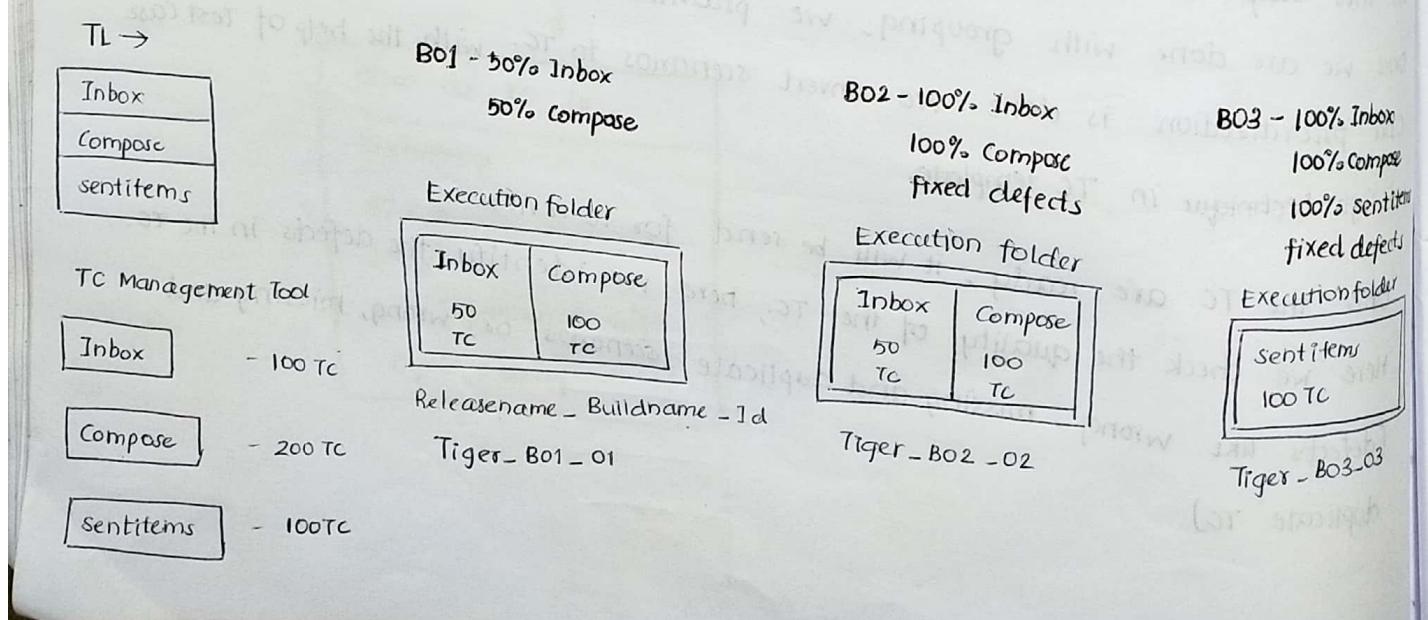
If the review comments are fixed, it will be send for TC approve. PM / Sr. TE / customer might approve the TC. Once the TC are approved, it will be to the repositories. [QC/jira/ALM]

QC - Quality center ALM - Application Life cycle management.

What is Brain storming Meeting?

- It is a meeting conducted by TE, In this meeting the group of TE who are working requirement will discuss about scenarios what they have identified.
- Each and every TE will explain the scenarios what they have identified where the other TE will try to understand the scenarios what he is explaining.
- If the other TE, if they find any defects in scenarios like wrong, missing, duplicate scenarios that will be given as a feedback to the TE.
- The TE will update the scenarios.
- The cycle repeats on every TE
- This meeting will be measured by TL once after the meeting.
- With the help of this meeting, we will have a good coverage in the scenarios.

Procedure to execute the Test Case.



Customer will give the requirement on inbox, compose and sentitems. BA will explain the requirements on Inbox, compose and sentitems. These module will be allotted to the TE. The TE will create the folders with the name of the module and will write the TC in the respected folder of respected module. (Observe stage II)

Dev will give build B01 which consist of 50% of the inbox, 50% of compose module. Whenever we get a build the testing team will create an execution folder, that name consist of Release name - Buildname - id. In this execution folder the TC which are required for the execution of the modules which are present in this build will be copy, pasted in the folder. Once the copy, paste is done we execute this TC, if we find any defects, it will be communicated to the developer.

Now the dev will give a new build i.e., Build B02 which consist of the remaining 50% of compose and inbox module. Again for this build we create 1 more execution folder. In this folder we copy, paste the TC which are required to execute the modules which are present in this build. In this execution folder, we also copy, paste the TC which help to find the defects in Build B01. In the folder we also copy, paste the dependency TC. Once after copy, paste is done, we execute all the TC and if we find any defects we will communicate it to dev. Again the cycle repeats.

We create the execution folder each and every build. In the execution folder, we will be having

- 1) The TC required to test new feature
- 2) The TC which help to find the defects in previous build.
- 3) Dependency TC wrt defects and modules.

Execution Report.

Modules	Total TC written	Total TC executed	Total TC not executed	No. of TC pass	No. of TC fail	Pass %	Fail %
Inbox	100	60	40	40	20	66.6%	33.3%
Compose	200	100	100	60	40	60%	40%
Sent Items	150	110	40	70	40	63.63%	36.36%

$$\text{Pass \%} = \frac{\text{No. of TC pass}}{\text{No. of TC executed}} \times 100$$

$$\text{Fail \%} = \frac{\text{No. of TC fail}}{\text{No. of TC executed}} \times 100$$

The test execution report will be prepared once the TC execution is completed.

The template of this report is not standard., The template will be keep on changing company to company and project to project. we prepare the execution report with feature, release. Few companies will prepare the execution report at the end of project.

Test Case Execution Report. with respect to feature

Step no.	Test casename	Status	Comments
1.	CBA - FT - 01	pass	
2.		pass	-
3		fail	defect
15			
1.	CBA - FT - 02	pass	
2.		fail	defect
1			
1			

Test Case Execution Report with respect to compatibility Testing.

Test case execution Report with respect to browsers.

Stepno	Test case name	Google chrome		Fire Fox		Internet Explorer	
		Status	Comments	Status	Comments	Status	Comments
1.	CBA - FT - 01	pass	-	pass	-	pass	-
2.		fail	defect	pass	-	fail	defect
3.		pass	-	pass	-	fail	defect
4.							
5.							
1.	CBA - FT - 02	pass	-	pass	-	fail	defect
2.		Fail	defect	pass	-	pass	-
3.							
4.							
5.							

Test Case Execution Report with respect to OS.

Stepno	Testcasename	Windows 7		Windows 8		Windows 10	
		Status	Comments	Status	Comments	Status	Comments
1.	CBA - FT - 01	pass	-	pass	-	pass	-
2.		fail	defect	pass	-	fail	defect
3.							
4.							
5.							
1.	CBA - FT - 02	pass	-	fail	defect	pass	-
2.							
3.							
4.							
5.							

JSTQB Questions.

? What do you mean by bug release / defect Release?

Releasing the software know no.of defects

These defects should be of low severity and low priority

(severity - what kind of defect

priority - when defect has to be fixed)

? What do you mean by bug leakage / defect leakage

The defect missed by the TE but found by customer is called as Bug leakage / defect leakage.

? What is pesticide Paradox OR What is pesticide Paradise

When we execute the same TC again and again for a longer run will not help in finding new defects. This is called pesticide paradox.
To overcome this As a TE we should come up with creative scenarios. And update the TC.

? What is defect clustering?

The process of having n no.of defects in one module, where these defects are not uniformly distributed with other modules, is called as defect clustering.
This results in software failure. As a TE we should identify these modules in the initial stage of testing.

? What is defect cascading?

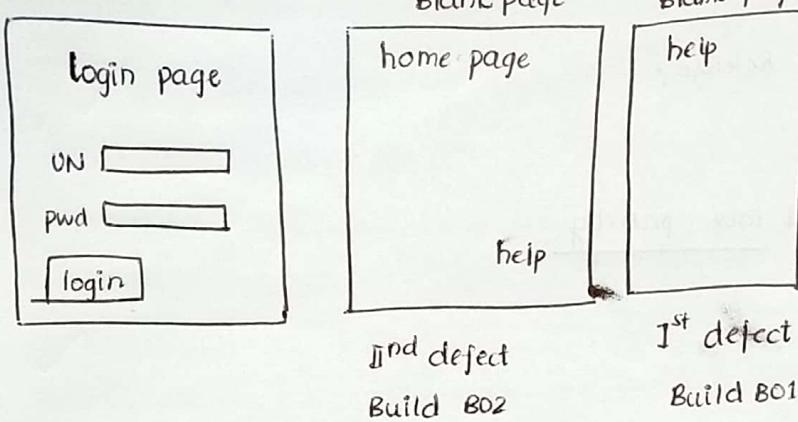
One defect triggering one more defect is called as defect cascading.

Phone no - Accept 10 digits Accepting 9 digit input - 1st stage
 └→ 9 digits

Amt - Accept 10-5000 +
 in balance the Amt is also deducted 2nd stage.

OK

? What is defect masking? OR defect hiding?



One defect hidden behind 1 more defect is defect hiding / defect masking.

? What is defect seeding?

Purposefully adding defects into the software is called as defect seeding.

? What is the meaning of formal testing?

The process of testing the application with some set of guidance of the Sr. TE (TM) and documenting the TC, TS, TP document and test strategy etc. In formal testing we prepare all kinds of execution reports, we prepare all kinds of defect report all these reports will be given to TL. In formal testing, we maintain the track of testing.

? What is functional requirement and non-functional requirement?

Functional requirement talks about how the feature should work. For eg: Username text field should accept 10 characters, pwd textfield should accept 10 characters, when we enter UN and pwd and click on login button, the home page should be displayed.

Non-functional requirement talks about speed of the application or look and feel of the application eg: We enter UN and pwd and click on login button, home page should be displayed within 1 sec.

? What is functional testing and non-functional testing
What are the types of testing.
OR.

The types of testings are

- A1) BBT and WBT [Pradeep Sir Ans]
- A2) Functional and non-functional [Google]
- A3) static and dynamic [Google]

Types of functional Testing.

Smoke Testing, FT, IT, ST, Adhoc Testing, Acceptance Testing, Regression Testing, α , β Testing

Types of Non-functional testing.

usability testing, GUI testing, performance testing, compatibility testing, Globalisation testing,

Yellow box testing, web-security testing, Reliability testing.

What is thick client?

Any software which is installed in client place or client sys and it will do lot of work. is called as Thick client.

Eg: MS office, paint etc.

What is thin client?

The software which are installed in the server place is called as thin client and that software will do lot of work in the server

Eg: www.google.com, www.facebook.com

What do you mean by accessibility testing?

The testing done from physically challenged people point of view is called as accessibility testing. The testing done to check whether the physically challenged people are able to use the application or not is called as accessibility testing

ISTQB Test.

1. The cost of fixing a fault? defect/Fault/Bug

- a) It is not important
- b) Increases as we move the product towards live use
- c) Decreases as we move the product towards live use
- d) is more expensive if found in requirements than in functional design
- e) Can never be determined.

2) Testing should be stopped when

- a) All the tests planned have been executed
- b) Time has run out
- c) All faults have been fixed
- d) Both a and c

e) It depends on the system's risk that being tested.

3. Error guessing is best used

- a. As the first approach to deriving TC
- b. After more formal techniques have been applied
- c. By inexperienced testers
- d. After the system has gone live
- e. Only by end users.

4. Non-functional system testing includes.

- a. Testing to see where the system does not function properly
- b. Testing quality attributes of the system including performance and usability
- c. Testing the software system using only the required action
- d. Testing the software system using only the required function
- e. Testing the function that should not exist.

5. Order numbers on the stock control system can range between 10,000 and 99999 inclusive. Which of the following

6. Could reviews or inspections be considered as part of testing.

- a. No, because they apply to development documentation
- b. No, because they are normally applied before testing.
- c. Yes, because they do not apply to the test case documentation
- e. Yes, because testing includes non-constructive activities.

7. What is visible to end-users is a deviation from the specific or expected behaviour, this is called

- a) An error
- b) A failure
- c) a mistake
- d) A fault
- e) A defect

8. Regression testing are performed

- a) Every week
- b) After the software has changed
- c) As often as possible
- d) When the environment/platform is changed
- e) When the project manager says.
- f) v and w are true, x-y-z are false
- g) w, x and y are true, v and z are false
- h) w and y are true, v-x-z are false
- i) w is true, v-x-y-z are false
- j) all of the above are true

9) The difference between Retesting and Regression Testing is.

- a) Retesting is running a test again, Regression testing looks for unexpected side effects
- b) Retesting looks for unexpected side effects, Regression testing is repeating a test again
- c) Retesting is done after faults are fixed, Regression testing is done earlier
- d) Retesting uses different environment, Regression uses same environment.
- e) Retesting is done by developers, Regression testing is done by testers

10) software testing is done ~~to~~ order to find bugs only

- a. True
- b. False

11) Bug is a software application is always due to wrong code written by the developer

- a. True
- b. False

12) One error in the software application leads to failure of the entire system.

- a. True
- b. False

13) Testing the software manually is.

- a. Time consuming
- c. Requires more testers
- b. Less accurate
- d. All of the above

- 14) We go for the automation when the project is
- a. Long term b. short term.
- 15) We can test a software application using automation tool and avoid manual testing completely?
- a. True b. False
- 16) As a tester when you find bugs in a software
- a. You will fix it
 - b. You will report it to developer
- 17) When a bug is found in the production environment the cost of fixing is high.
- a. True b. False
- 18) There is no difference between defect and a bug
- a. True b. False
- 19) Which one of the following are non-functional testing methods?
- a. System Testing
 - b. Usability Testing
 - c. Performance testing
 - d. both b and c
- 20) Boundary value testing
- a. Is same as equivalence partitioning tests
 - b. Test boundary conditions on below and above the edges of input and output equivalence classes
 - c. Testing combinations of input circumstances
 - d. is used in WBT strategy.
- 21) Acceptance test cases are based on what?
- a. Requirements
 - b. design
 - c. code
 - d. Decision table.
- 22) Pick the best definition of quality
- a. Quality is job one b. zero defects. c. Conformance to req
 - d. Work as designed.

Decision Table Technique

In this technique we check for multiple conditions, combinations and rule criterias.

$$2^{\text{no. of conditions}} = \text{Total no. of rules / scenarios.}$$

Eg: If you're a new customer, if you're having credit card account then on your purchase Bank will give 15% discount.

If you're an old customer, already you have credit old account then on your purchase Bank will give 10% discount.

If you have coupon code, then on your purchase Bank will give 20% discount.

[Note: Coupon code can't be used together with new customer discount. If it is used then highest discount will be considered].

$$2^3 = 8 \text{ scenarios.}$$

Scenarios.

No. of conditions	Rule 1	Rule 2	Rule 3	Rule 4	Rule 5	Rule 6	Rule 7	Rule 8
New customer discount (15%)	T	T	T	T	F	F	F	F
Old customer discount (10%)	T	T	F	F	T	T	F	F
Coupon code (20%)	F	F	T	F	T	F	T	F
Overall discount	No discount	No discount	20%	15%	10+20 30%	10%	20%	No discount

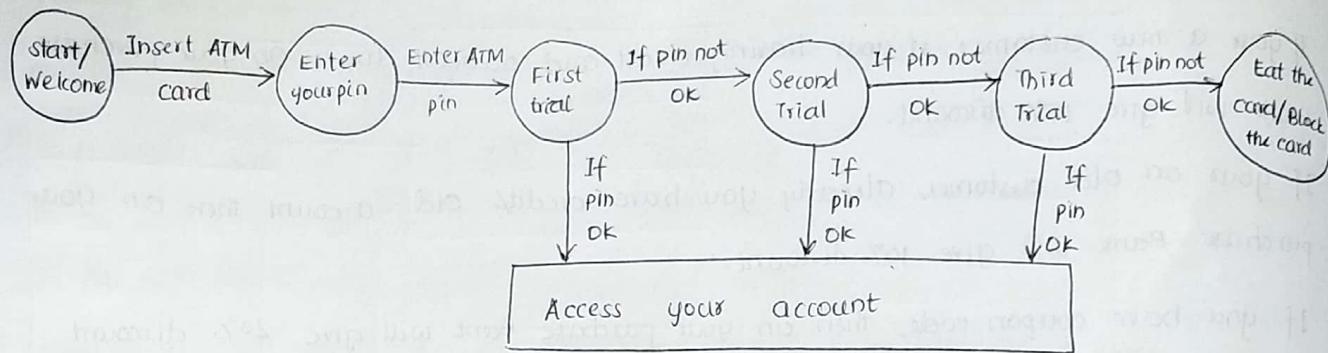
Eg: For UN and pwd

No. of condition	R ₁	R ₂	R ₃	R ₄
UN	T	T	F	F
pwd	T	F	T	F
Final Result	Home	Error	Error	Error

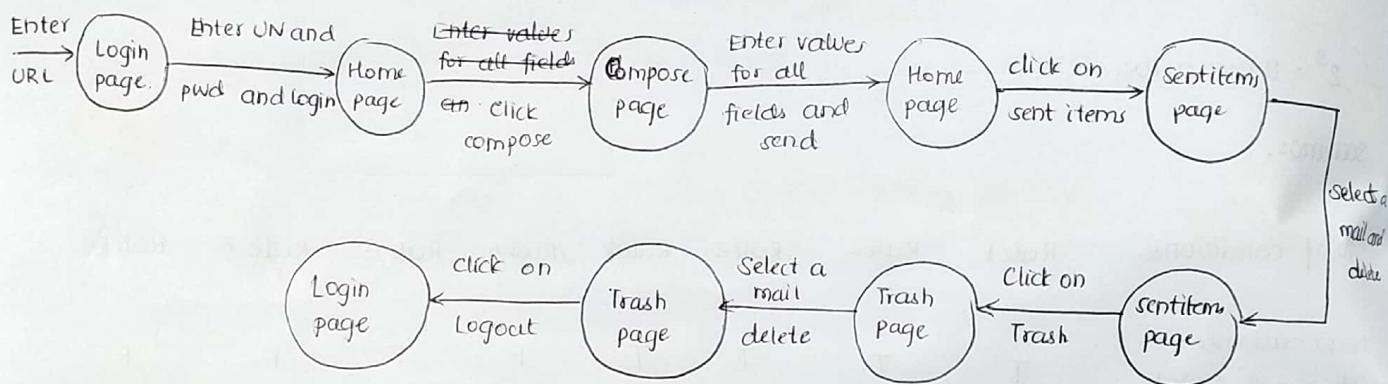
State Diagram

In this technique we will check for different state of a system / software / machine

e.g. ATM machine



e.g. gmail.



I Write state diagram for 1 positive and 1 negative end to end scenarios for any app.

II Write decision table technique for your own application.

Mistakes done while writing Test Cases.

1. Open browser and enter URL X } Action
- Open browser and enter Test URL ✓ }
2. www.gmail.com X } Input
- https://QA-gmail/login.jsp ✓ }
- ↓
- server name compulsory
3. Click on compose X } Action.
- Click on compose button ✓ }
4. 'compose' page should be displayed with following fields X } ER.
- 'To' text field 'sub' text field 'send' button
- 'cc' text field 'Body' text area field 'cancel' button ✓ }

5. No shortcuts.

Step no1:

Step no2: Enter UN and pwd and click on 'Login' button

Step no60: Same as step no.2 / Repeat as step no.2 X Step no.60: Enter UN and pwd and click on 'Login' button ✓

6. Use simple Language.

- 'WP' shld X

- 'Om' pg should X

- ck 'k' b/n X

7. Don't split 1 step into 2 or more steps.

Step 1: Enter UN

Step 2: Enter pwd

Step 3: Click on 'login' button X

Step 1: Enter UN and pwd and click on login button ✓

8. Don't merge 2 steps or more steps into 1 step.

Step 1: Click on compose button and enter values for all the fields and click on 'send' button. X

Step 1: Click on compose button

Step 2: Enter values for all the fields and ~~not~~ click on 'send' button. ✓

9. Always highlight the Object name.

Step 1: click on compose button X Step 1: click on 'compose' button. ✓

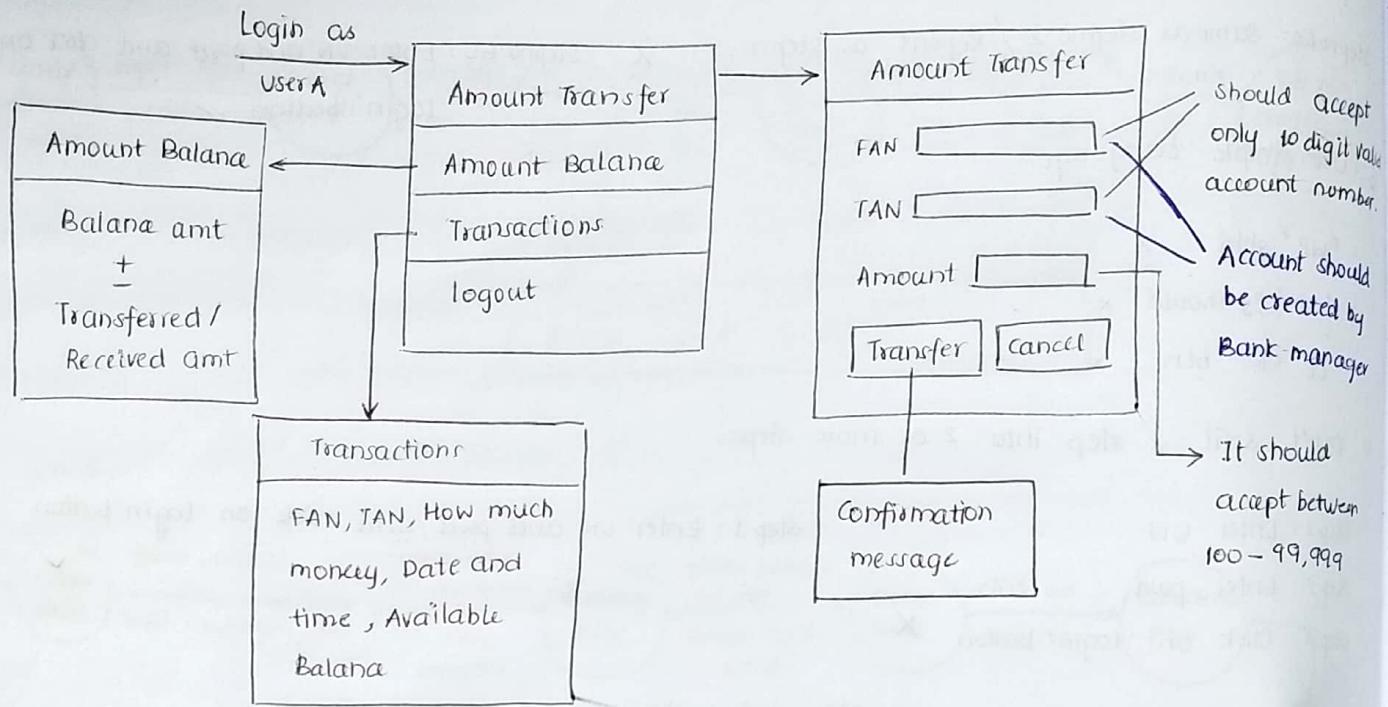
10. Don't hard code test cases.

Action	Input	ER
Enter 'UN' as 'DINGA' and 'pwd' as 'DINGA' and click on 'Login'	NA	Home 'Compose' page should be displayed X

11. If something is covered in functional Test case then don't cover in Integration Test case and if something is covered in IT then don't cover in s/m Test case.

How to write Functional Test Case in Test Case Template [By applying design technique]

e.g: Requirement



Agile - Fast, Easy and quick.

Agile is a iterative and incremental approach where customer can change the requirement at any stage. As a company we should be flexible enough to accept the changes, develop the changes, test the changes and deliver the quality software to the customer within a short span of time.

In agile we will have a good communication between the BA, customer, developer and TE. As a TE if we have any doubts we can communicate to the client, the developer and TE they can communicate to client, collect the feedback from the client, develop it, test it and give the quantity software to the customer. This software which it is released to the customer is a working piece of software.

The main goal of agile is customer satisfaction by giving a quick delivery of working software with good quality.

What are the principles of Agile?

The principles of agile are:

1. In agile we will have a good communication between the customer, BA, dev, TE and scrum master.
2. In agile the releases are short (4 weeks once or 5 week once or 6 weeks once or 7 weeks once or 8 weeks once or 9 weeks once. Few companies will also give 3 weeks once release or 16 weeks once release).
3. In agile requirement changes are allowed
4. The main goal of agile is customer satisfaction by giving a quick delivery of working piece of software with good quality.
5. It is a simple model to adopt.
6. In agile the dev and TE will be conducting more meetings to deliver a good quality software to the customer.

What is Agile Testing?

If we are testing the application by following the principles of agile methodology is called Agile Testing.

Types of Agile Methodology

- * 1. Scrum
- 2. XP [Extreme Programming]
- 3. FDD [Feature Driven Development]
- 4. Lean and ka
- 5. Crystal clear
- 6. ASDM - Adaptive software development Method
- 7. DSDM - Dynamic software development method.

Scrum Methodology

1. Release - Release is a combination of sprints.
2. EPIC - It is a complete set of requirements. In EPIC we will get the requirement of individual feature functionality and module.
BA will create EPIC in the project management tool.

3. Stories / User stories / User cards.

Features , Functionalities or modules is called as stories.

The BA will create the EPIC for each and every individual stories.

Scrum Master.

Scrum master is a person who takes complete responsibility for the devan testing of the project, and delivering the project to the customer with good quality scrum master will track each and every developers and TE, he will assign work for the dev and TE , and he will also cross check whether the task is completed within a given period of time.

Scrum master will be working in project management tool where he will be creating stories and task for the stories and assigning these tasks for dev,TE and tracking these tasks.

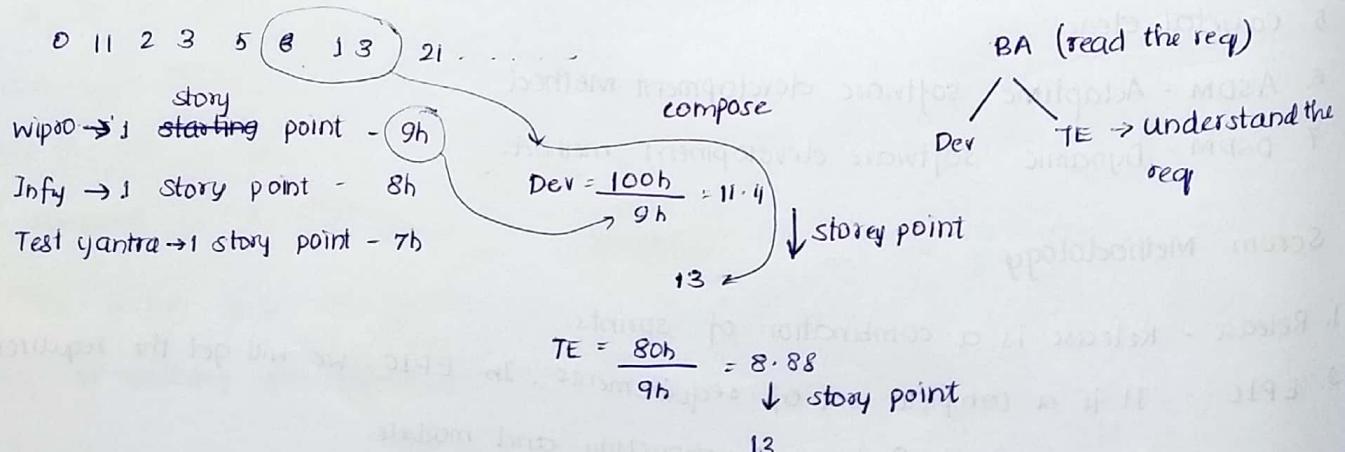
Who can become a scrum Master?

BA or sr. TE or sr.Dev or PM can become . sometimes even customer can play the role of the scrum master.

Story Points.

It is a rough estimation given by the dev and by TE for the dev and Testing of individual stories.

story points will be given in a form of fibonacci series.



In this meeting we discuss

1. Yesterday what you did
2. Today what are your plans.
3. Are there any obstacles/impediments.

Sprint Retrospective medium.

It is the meeting conducted by scrum master on the last day of every sprint

BA, dev, TE, PM will be part of this meeting.

In this meeting we discuss the below mentioned points.

1. What went well
2. What didn't go well
3. Are there any action plans.

Release Retrospective

This is a meeting conducted by scrum master on the last day of every release.

Note: Remaining points are same with respect to retrospective meeting.

3w	2w	8w	4w
1 2 3 4 5		1	1

Swag

It is a rough estimation given by the dev and TE for the dev and testing of individual stories in the form of hrs.

Sprint.

Sprint is the actual duration took by dev and TE for the dev and testing of individual stories.

Sprint planning.

It is a meeting conducted by scrum master on the 1st day of every sprint.

In this meeting BA, dev, TE, customer (optional) and PM will be part of this meeting.

In this meeting BA will explain the requirement, dev and TE will try to understand the requirement, if they have any doubt, they will contact BA, BA will clear the doubts.

Scrum master will assign the task to the dev and TE based on their availability.

In this meeting the scrum masters master will prioritize the requirement based on their customer needs.

Scrum meeting.

It is a meeting conducted by scrum master on daily basis, It is also called as daily stand up meeting, This meeting will be strictly bounded for 15-20 mins. In this meeting BA, PM, Dev, TE will be part of it.

Bug triage meeting

It is a meeting conducted by TE / BA. This meeting will be conducted one or two weeks before the release.

BA, dev, scrum master will be part of this meeting. In this meeting the TE will get the list of defects which are open and which are not fixed by the dev, in this meeting we will discuss when the defects will be fixed by the dev.

The scrum master has a team will prioritize the defects and will group the defects in the form of which ones to be fixed in the current release and which can be postponed to the next release.

In this meeting we will also discuss about the defects which were postponed from the previous release.

Product Backlog Groomings.

It is a meeting conducted by BA or scrum master One or two weeks before the release.

Dev, TE and scrum master will be part of this meeting

In this meeting we will discuss about the features / stories which are not yet developed, the scrum master will prioritize the features and will decide to let which are the feature should be developed in current release and which are the features to be postponed in next release.

In this meeting we will also discuss about the features which were postponed from the previous release.

Bug Bucket

It is an epic which is created by the scrum master or BA. Inside the EPIC we will have the information of the defect which are yet to be retested by the TE. This will be created in the project management tool. This folder will be addressed by the TE, the dev, scrum master, BA for every individual release. In this bug bucket, it consists of the defect like

1. NFT - New feature Test
2. Regression Bugs.
3. Performance issues
4. Web security issues.

(OR)

It is an EPIC or a story which is prepared by a scrum master or BA in the project management tool.

Chickens and Pigs

Chickens are the people who are not committed to the work, they get the work done by others.

e.g.: customers, stake holder, Management team, Project manager.

Pigs

Pigs are the people who are committed to the work and will complete the work within given period of time.

e.g.: scrum master, BA, dev, TE.

Agile Process

Above is a process which is an example for a project of 12 months. In this project for every 6 weeks once we give a release to the customers. In one release we have 3 sprints, each sprint is a duration of 2 weeks. Therefore in this project we will have 8 releases.

Burn down chart.

It is a graphical representation of total no. of work done v/s no. of weeks which will be prepared by the scrum master or BA or PM at the end of every release to track the progress of the project. With the help of this we will come to know or we can calculate the total amount of work remaining at the start of the every sprint or at the beginning of every sprint.

Burn up chart

It is a graphical representation which helps to calculate total amount of work completed at the end of every sprint.

Drawbacks of Agile

1. The scope of documentation is less.
2. We can't go for long term and complex projects.
3. We prefer experienced candidates to work in Agile.
4. It is difficult to do the effort estimation for critical and complex project in the beginning.
5. Suppose if customers are not aware of the requirement chances are there we might mess up the project.

Test Case Template. (Web application)

Component name : FAN Textfield

Test scenario 1.

1. To check that 'FAN' textfield should accept only 10 digit valid account no.
2. To check that 'FAN' textfield should not accept 9 digit account no.
3. To check that 'FAN' textfield should not accept alphabets.
4. To check that 'FAN' textfield should not accept alphanumeric.
5. To check that 'FAN' textfield should not accept special characters.
6. To check that 'FAN' textfield should not accept multiple account number.
7. To check that 'FAN' textfield should not accept blank.

8. To check that 'FAN' textfield should not accept other language. (2-10 → Erroneous)
9. To check that 'FAN' textfield should not accept 9 digit account number
10. To check that 'FAN' textfield should not accept other bank account number
- (11-12) Pressman Rule 1 (since no deviation)
11. To check that 'FAN' textfield should not accept combination of alphabets, no., spl ch.
12. To check that 'FAN' textfield should not accept number with special character. invalid
- (13-) BVA A - B
S - 10
13. To check that 'FAN' textfield should not accept single digit account number. A
14. To check that 'FAN' textfield should not accept two digit account number At 1
15. To check that 'FAN' textfield should not accept decimal values.
- To check that 'FAN' textfield should accept first 10 char
To check that 'FAN' textfield should accept last 10 spl char

Test Case.

Test Case name : : CBO_Amounttransfer

Project name : : city Bank

Release name : : Lion

Requirement number : : 13.1.1

Module name : : Amount transfer

Pre-condition : : Check for internet connection

Test data : : user A account should be created with 10k balance

Post-condition : : NA

Severity : : Critical

Test case type : : Functional Testing

Brief description : : In amt transfer module, FAN textfield should accept valid acnt nos

Test case execution hours :

Step No.	Action / Description	Input	Expected Result	Actual Result	Status	Comments
1.	Open browser and enter test URL	https://QA.citibank/login.jsp	'Welcome'/'login' page should be displayed.			
2.	Enter 'UN' into 'un' textfield and 'pwd' into 'pwd' textfield and click on 'login' button. OR Login as user with valid 'un' and 'pwd'.	UN: User A pwd: User A	'Home' page should be displayed with following fields. 'Amount Transfer' button 'Amount Balance' button 'Transaction' button 'Logout' button.			
3.	Click on 'Amount Transfer' button	NA	'Amount Transfer' page should be displayed with following fields 'FAN' text field 'TAN' text field 'Amount' text field 'Transfer' button 'Cancel' button.			
4.	Enter valid 10 digit account number into 'FAN' textfield and click on 'Transfer' button.	FAN: 1234567891	should be accepted			
5.	Enter 9 digit account number into 'FAN' textfield and click on 'Transfer' button.	FAN: 123456789	should not accept 'Error message' should be displayed.			
6.	Enter alphabets into 'FAN' textfield and click on 'Transfer' button.	FAN: ABCD	'Error message' should be displayed.			
7.	Enter alphanumeric into 'FAN' textfield and click on 'Transfer' button.	FAN: AB12CD	'Error message' should be displayed			

8.	Enter special character into 'FAN' textfield and click on 'Transfer' button	FAN: @*\$*!	'Error message' should be displayed.
9.	Enter multiple account number into 'FAN' textfield and click on 'Transfer' button	FAN: 1234567891 1324567891	'Error message' should be displayed
10.	Enter Don't enter any values into 'FAN' textfield and click on 'Transfer' button.	FAN:	'Error message' should be displayed.
11.	Enter Other language into 'FAN' textfield and click on 'Transfer' button.	FAN: my@omj.	'Error message' should be displayed.
12.	Enter 11 digits into 'FAN' textfield and click on 'Transfer' button.	FAN: 12345678910	'Error message' should be displayed.
13.	Enter 10 digit Other bank account number into 'FAN' textfield and click on 'Transfer' button.	FAN: 5621325121 (Other Bank)	'Error message' should be displayed.
14.	Click on 'Logout' button	NA:	User A should be successfully logged out from the application.

Author : Sukanya CR

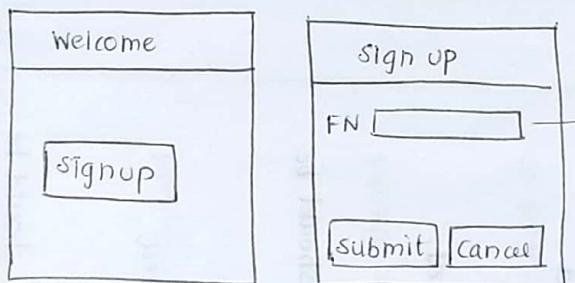
Reviewed by : Vicky

Approved by : Pradeep Aldur

Approval date : 15.07.2019.

eg: 2:

gmail- App [client - server]



Component name: 'FN' textfield

Scenario.

1. To check that 'FN' textfield should accept 3 alphabets.
2. To check that 'FN' textfield should accept 10 numbers.
3. To check that 'FN' textfield should accept 8 alphanumeric
4. To check that 'FN' textfield should accept 9 digits.
5. To check that 'FN' textfield shouldnot accept blank.
6. To check that 'FN' textfield should accept 2 alphanumeric
7. To check that 'FN' textfield shouldnot accept special character.
8. To check that 'FN' textfield should accept Uppercase and Lowercase.
9. To check that 'FN' textfield shouldnot accept decimal values.
10. To check that 'FN' textfield shouldnot accept alphanumeric with spl characters.

Test case - name : gmail-signup- FN textfield Functional scenarios.

Project name :

Release name :

Req number :

Module name

Pre condition

Test data

Post condition

severity

Test case type

: software should be installed, check for internet connection.

: NA

:

:

:

Brief description :
Test case execution
hours :

			Expected Result	AR	Status	Comments
1.	Open the gmail application Click on 'gmail' application icon	NA	'Welcome' page should be displayed with following field 'sign up' button.			
2.	click on 'sign up' button	NA	'signup' page should be displayed with following fields 'FN' text field should be mandatory 'submit' button 'Login' button.			
3.	Enter 3 alphabets into 'FN' textfield and click on 'submit' button	ABC	should be accepted.			
4.	Enter 10 numbers into 'FN' textfield and click on 'submit' button	1234567891	should be accepted.			
5.	Enter 8 alphanumeric into 'FN' textfield and click on 'submit' button	IA2B3C4D	should be accepted			
6.	Enter 9 numbers into 'FN' textfield and click on 'submit' button	i23456789	should be accepted.			
7.	Don't enter any values into 'FN' textfield and click on 'submit' button	IA	'Error message' should be displayed.			
8.	Enter 2 alphanumeric into 'FN' textfield and click on 'submit' button	*@#	should be accepted			
9.	Enter special characters into 'FN' textfield and click on 'submit' button	AaBb	'Error message' should be displayed			
10.	Enter uppercase and lowercase into 'FN' textfield and click on 'submit' button	0.183	should be accepted.			
11.	Enter decimal values into 'FN' textfield and click on 'submit' button.	12A\$	'Error message' should be displayed.			
12.	Enter alphanumeric and special characters into 'FN' textfield and click on 'submit' button	NA	'Error message' should be displayed. Application should be closed.			
13.	Click on 'close'					

Author : sukanya CR

Reviewed by : Vicky

Approved by : Pradeep Aldur

Approval date : 14.07.2019.

I write functional TC in TCT in MS excel for below mentioned components [min 15 scenarios]
'UN' textfield, 'Address' text area field, 'pwd' text field.

1STQB Questions.

? What are types of software testing.

WBT and BBT X

Functional and non-functional X

static and dynamic

Difference between static and dynamic - Verification and Validation.

Static	Dynamic
<ul style="list-style-type: none">• Static testing involves verification activities.• Verification means it includes reviews, walk through, inspection and auditing.• To prevent the defects.• It is done before software is developed.• Here we don't execute the code.• We ensure are we building product right.	<ul style="list-style-type: none">• It involves validation activities.• Validation includes actual testing i.e., smoke, FT, IT, ST etc.• It is done to identify the defects.• After software is developed.• We execute the code.• We ensure are we building right product.

What is walkthrough?

It is a meeting conducted by the TE where in TE will explain Test scenarios and Test case to the entire team and get the input from the team and improve the code.

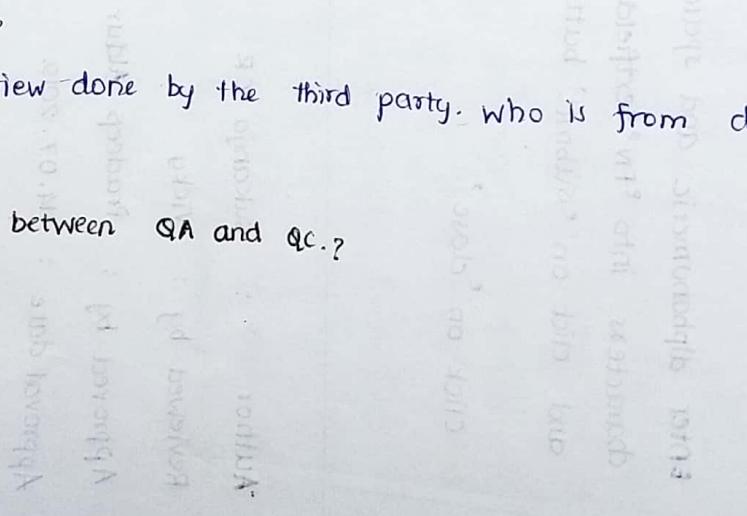
What is inspection?

Inspection is a type of review done by the team member working on the same project.

What is auditing?

It is a type of review done by the third party who is from different team or different company.

* What is difference between QA and QC?



QA (Quality Assurance)

- QA is the set of activities to improve quality in process
- QA is proactive process (planning)
- QA is process oriented
- QA will aims to prevent the defect
- The goal of QA is to improve development and testing process so that, defect should not arise in future
- QA means verification
- QA means planning
- QA means product right

QC (Quality Check / control)

- Set of activities to improve quality in product.
- QC is reactive process (executing)
- QC is product oriented.
- QC aims to identify the defect.
- The goal of QC is to find the defects after software is developed.
- QC means validation.
- QC means execution.
- QC means right product.

How to write Integration in Test Case Template.

module name: Amount transfer.

Scenarios.

1. Login as user A, transfer money to user B click on transaction and check whether proper transaction history is displayed.
2. Login as user A, transfer money to user B, ~~logout as user A~~ click on amount balance, check whether proper amount balance is displayed.
3. Login as user A, transfer money to user B, logout as user A, login as user B, click on amount Balance, check whether proper amount balance is displayed.
4. Login as user A, click on amount transfer, enter invalid FAN and enter values for all other fields, click on transfer, error message will be displayed, logout as user A, login as user B, click on amount balance, check whether proper balance amount is displayed.
5. Login as user A, click on amount transfer, enter valid FAN and TAN and enter the amount more than balance click on transfer, proper error message will be displayed. click on amount balance, and check whether proper balance amount is displayed.

Stepno.	Action / Description	Input	Expected Result	AR	Status	Comments
1.	Open the browser and enter test URL Login as UserA with valid "UN" and "pwd"	https://QA-citibank/Login.jsp UN - UserA pwd - UserA	"Welcome" page should be displayed			
2.		NA	"Home" page should be displayed with following fields. If something is covered in functional don't cover in IT			
3.	Click on "Amount Transfer" button	NA	"Amount transfer" button			
4.	Enter valid data into "FAN" textfield, "TAN" textfield, "Amount" text field and click on "Transfer" button	FAN : User A TAN : User B Amount : 10000	"Amount transfer" page should be displayed with following fields. "FAN" textfield "TAN" textfield "Amount" textfield "Transfer" button "cancel" button. Confirmation message should be displayed.			
5.	click on "Transaction" button.	NA	"Transaction" page should be displayed with proper Transaction history (User A , User B , 10000 , Date and time, 90000)			
6.	click on "Amount Balance" button	NA	"Amount Balance" page should be displayed appropriate Balance amount. (90,000)			

7.	click on 'logout' button	NA	UserA should be successfully logged out from the application.
8.	click on User with valid 'UN' and 'pwd'	UN: UserB pwd: UserB	'Home' page should be displayed.
9.	click on 'Amount Balance' button	NA	'Amount Balance' page should be displayed with proper fields.
10.	click on 'logout' button	NA BALANCE : 10000 NAME : NA AMOUNT : NA	10,000 User B should be successfully logged out from the application.
11.	login as user with valid 'UN' and 'pwd'	UN: UserA pwd: UserA	'Home' page should be displayed,
12.	click on 'Amount transfer' button	NA	'Amount Transfer' page should be displayed.
13.	Enter invalid data into 'FAN' textfield, Valid data into 'TAN' and 'Amount' Textfield and click on 'Logout' button	PAN: Invalid TAN: Valid Amount: 10000 NA	'Error message' should be displayed UserA should be successfully logged out from the application.
14.	click on 'Logout' button	NA	'Home' page should be displayed
15.	Login as user with valid 'UN' and 'pwd'	UN: UserB pwd: UserB	'Amount Balance' page should be displayed (10000)
16.	click on 'Amount Balance' button	NA	UserA should be successfully logged out from the application
17.	click on 'Logout' button	NA	'Home' page should be displayed
18.	Login as user with 'UN' and 'pwd'	UN: UserA pwd: UserA	'Amount Transfer' page should be displayed
19.	click on 'Amount transfer' button	NA	'Error message' should be displayed
20.	Enter valid data into 'FAN' textfield 'TAN' textfield and invalid data into 'Amount' and click on Transfer button	FAN: UserA TAN: UserB Amount: 200000	'Amount Balance' page should be displayed with proper Balance (90,000)
21.	Click on 'Amount Balance' button	NA	UserA should be successfully logged out from the application.
22.	click on 'Logout' button	NA	

Author : Sukanya CR

Reviewed by : Vicky

Approved by: Pradeep Aldur

Approval date:

I eg: gmail.

Component name: Inbox.

1. Login as userA, click on Inbox, select a mail, delete a mail, click on Trash and check whether mail is displayed
2. Login as userA, click on Inbox, select a mail, move a mail to spam, click on spam and check whether mail is displayed in spam.
3. Login as userA, click on Inbox, select a mail, move a mail to spam, click on dm and check whether mail is displayed in Trash.
4. Login as userA, click on Inbox, select a mail, switch off internet connection, delete mail, click on Trash and check whether mail is displayed in Trash.

Test case name : gmail_inbox_Integration scenarios.

Project name :

Release name :

Requirement number:

Module name :

Pre condition : Application should be installed, check for internet connection,

Test data : Create userA account, 15 mails should be in userA account.

1.	click on 'gmail' application icon	NA	'Welcome' page should be displayed
2.	Login as user with valid 'UN' and 'pwd'	UN:UserA pwd:UserA	'Home' page should be displayed.
3.	Click on 'Inbox' button	NA	'Inbox' page should be displayed with some mails.
4.	select a mail and click on 'delete' button	NA	Mail should be deleted successfully.
5.	Click on 'Trash' button	NA	'Trash' page should be displayed with all the deleted mails.
6.	click on 'Inbox' button	NA	'Inbox' page should be displayed with some mails.
7.	effe select a mail and click on 'move to Spam' button.	NA	Mail should be moved to 'Spam'.
8.	click on 'spam' button	NA	'Spam' page should be displayed with mail.
9.	click on 'Draft' button	NA	Mail should not be displayed in 'draft'.
i.	login as user and click on 'Inbox' button	NA	'Inbox' page should be displayed with some mails.
ii.	Select a mail and switch off internet connection and click on 'delete' button	NA	Mail should be deleted.
12.	Click on 'Trash' button	NA	Mail should not be displayed.
13.	Click on 'Logout' button	NA	UserA should be logged out from the application.

Author : Sukanya CR

Reviewed by : Vicky

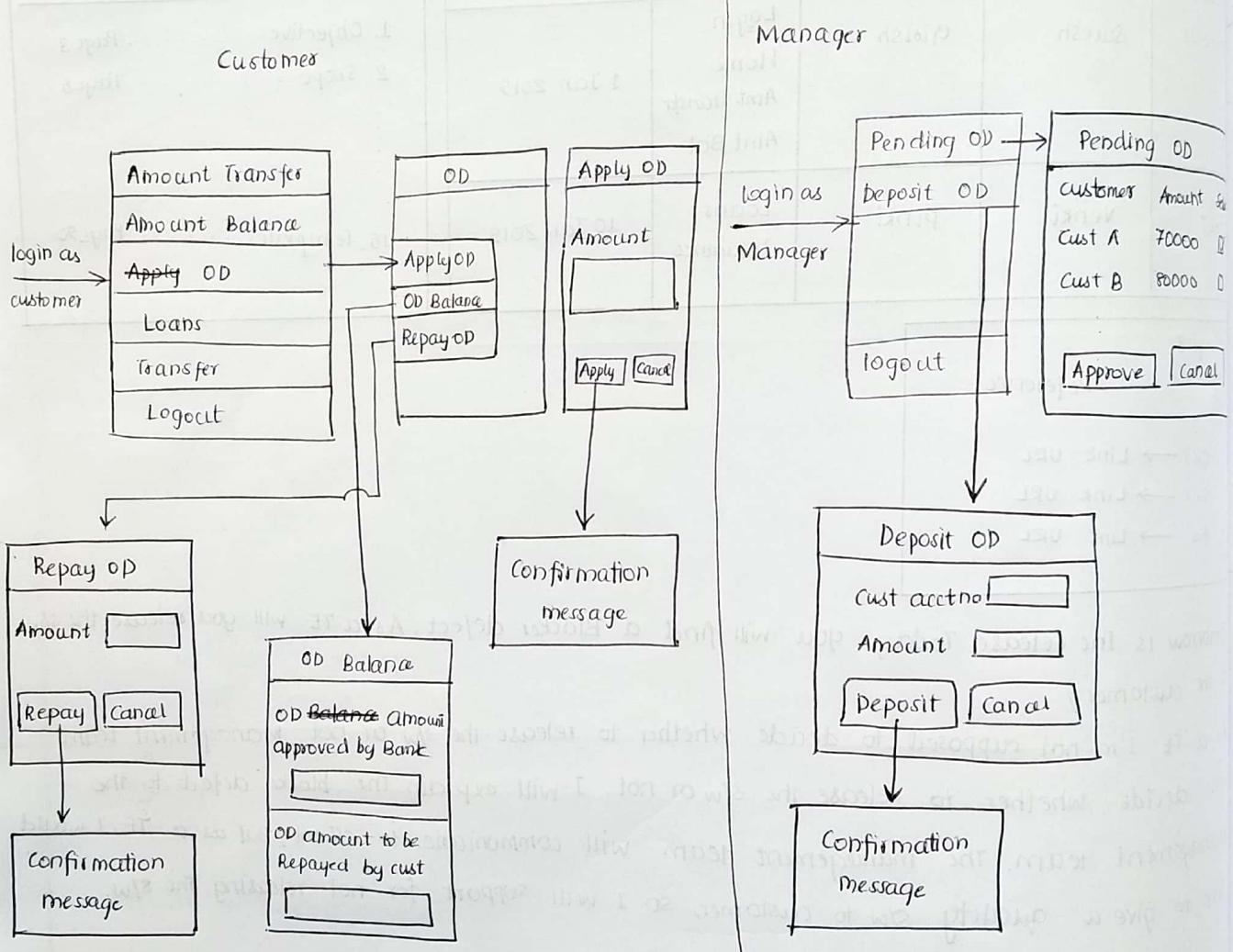
Approved by : ~~APP~~ Pradeep sir

Approval date : 16.7.2019

Write 4 Integration scenarios for below mentioned application

a) Facebook.com b) WhatsApp.com

How to write System Test Cases.



Scenarios

- I. Login as customer apply for OD for Rs.20,000 logout as customer. Login as Manager, Approve the pending OD, logout as Manager. Login as customer, check the OD Balance (20,000) After one month, make sure that 2% of rate of interest and 250 Activation fees is charged for customer (20650) Login as manager, deposit the OD logout as manager, login as customer. Check the OD Balance Repay the OD check the balance make sure that OD Balance is 000.

Test case name : CBO - OD - Apply OD System Testcase / scenario.
Project name : citibank
Release name : Dolly
Requirement no : 24-01-2016 24.1.06
Module name : OD
Precondition : Check for internet connection, check step to change system date
Test data : Create Manager account
Post condition : Some data should be displayed in OD Balance
Severity : Critical
Test case type : System
Brief description : In OD module, we are covering apply OD end to end flow
Test case execution hours

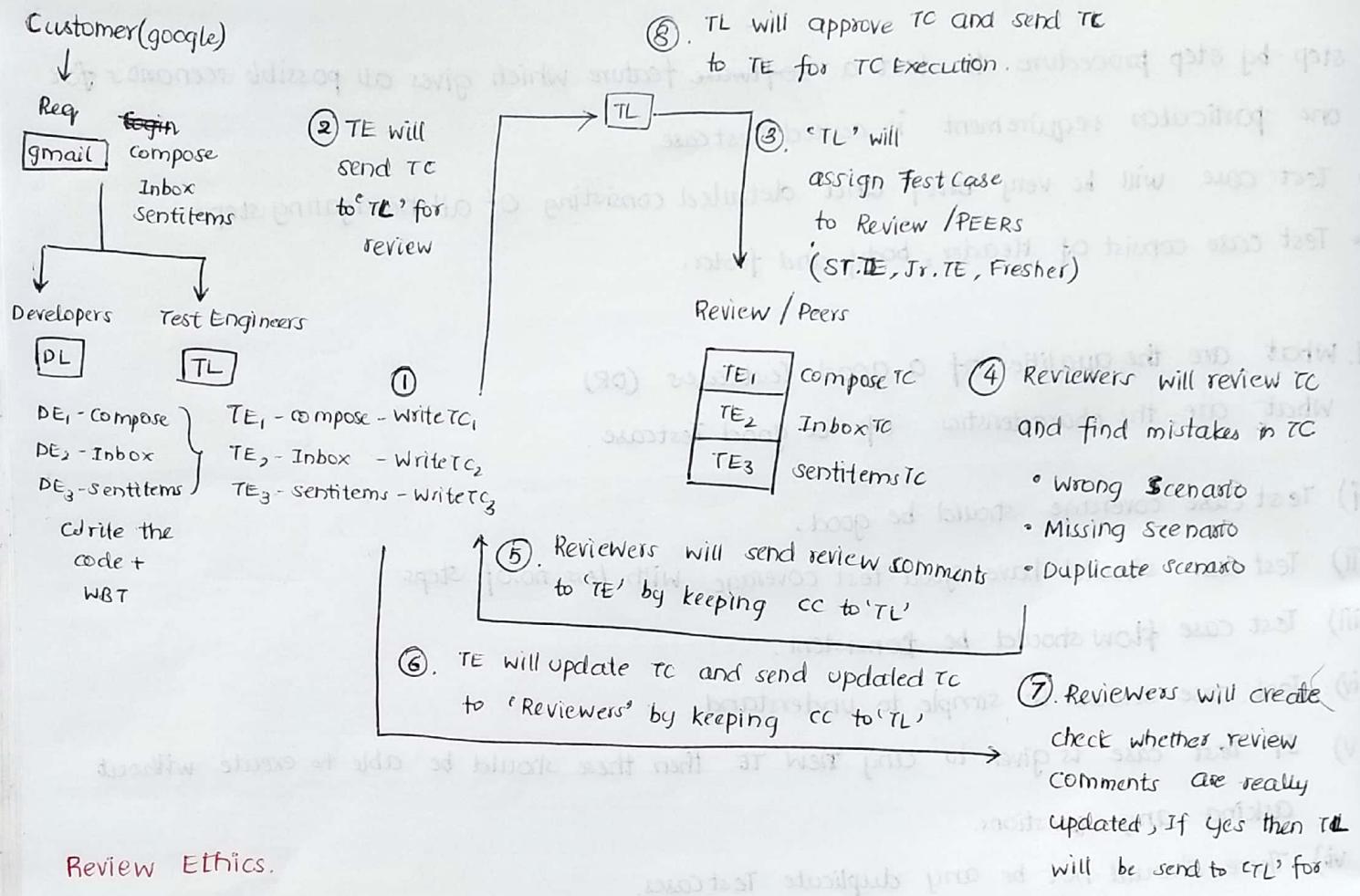
Test Case / Test specification.

- step by step procedure to test a ~~software~~ feature which gives all possible scenarios for one particular requirement. is called Test case.
- Test case will be very brief and detailed consisting Of all navigating steps.
- Test case consist of header, body and footer.

1. What are the qualities of a good Testcases (OR)
What are the characteristics of a good Testcase.

- i) Test case coverage should be good.
- ii) Test case should have good test coverage. with less no.of steps.
- iii) Test case flow should be consistent.
- iv) Test case should be simple to understand.
- v) If Test case is given to any new TE then these should be able to execute without asking any questions.
- vi) There should not be any duplicate Testcases.
- vii) Test case should be written by applying Test case design technique.
- viii) Test case should be written in test case template.
- ix) Test case consist of both positive and negative scenarios.
- x) Test case should be able to catch the defects.
- xi) Test case should be simple to convert into automation scripts.

Test Case Review Process / Review Process / Peer review Meeting.



Review Ethics.

1. Review the content not the Author.
2. For reviewing spent time only in finding the mistakes not in giving the solution.
3. Once after Test Case Review is completed and later if 'TL' or customer find the mistakes in TC then both are responsible.

On what basis 'TL' identify Reviewers.

1. A person who is having very good project knowledge.
2. A person who is working on same requirement.
3. We can assign TC to any TE's (Sr.TE, Jr.TE, Fresher) and ask them to learn the requirement and review the TC.

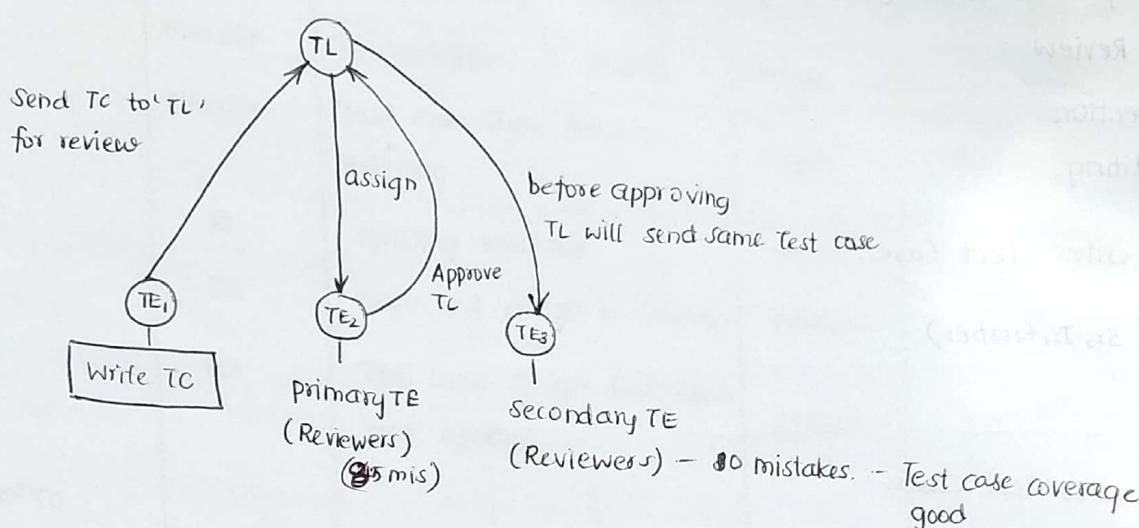
Note: If Reviewers find 100 mistakes, there is no rule that we should fix all the 100 mistakes. We should fix the mistakes which make sense. Because sometimes even reviewers will do mistakes while reviewing TC.

4. How 'TL' ensures that reviewers are reviewing TCs properly, or

How TL ensures that reviewers are reviewing Test cases properly (OR)

What is the guarantee that reviewers are really find mistakes where reviewing TC.

Case 1

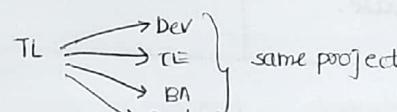


Case 2

TL will randomly pick some of the TC which are already reviewed by the reviewers, and TL himself will crosscheck whether the TC are reviewed properly or not.

Case 3.

TL will setup a meeting and invite developers, TE, BA, customers (sometimes) working on the same project and review the TC.



* What mistakes TEs will find while reviewing Test case.

1. TE will find missing scenarios, wrong scenarios and duplicate scenarios.
2. TE can find whether TC coverage is good or not.
3. Check whether TC flow is consistent or not.
4. Chances are there where TE copy, paste the TCs., they might not change the header, so check whether header is relevant or not.
5. Check for the spelling mistakes and sentence formation by looking into body of TC.
6. Check whether in the TC header, all the attributes are present or not.
7. Check whether in the header all the attributes are having relevant data or not.
8. Check whether TC are written in proper TC template or not.
9. Check whether TC are written by applying TC design techniques or not.
10. By looking into body of TC check whether all the steps, input and expected result is proper or not.
11. Check in the footer all the fields are present with proper data or not.

What are the types of TC Review.

1. Self Review :
2. Management Review (BA, TL, cust...)
3. Peer Review :
4. Inspection
5. Auditing.

Who writes Test Cases.

1. TE (sr, Jr, fresher)

2. TL

Q:

Who review Test Cases.

1. TE (sr, Jr, Fresher)

2. TL

3. Developer

4. BA

5. Customer (sometimes).

Who approve Test Case.

1. TL.

2. Test Manager.

3. BA

4. customers.

Test case review template. (It is not standard).

Project name :

Release name :

Feature/module/story :

Requirement number:

Author:

Reviewed:

Reviewed date:

Sno.	Test Case name	Step no	Reviewer		Author comments
			Comments	Severity	
1.	CBO_AT_FANTC	Header	Test Case name is wrong	major	Fixed
		Header	Pre-condition is wrong	major	Not fixed
		Header	Test Case Type field is missing	major	fixed
		10	Spelling mistake	minor	Fixed
		20	Expected result is missing	critical	Fixed
		40	Test case design techniques not applied	critical	Fixed
2.	CBO_AT_TANTC				

How do you ensure that your Test coverage is good. (OR) How do you convince your test manager or customer stating your test coverage is good.

1. I have written Test cases by applying Test case design technique, so my test coverage is good.
2. I have covered all positive and negative scenarios.
3. I have conducted brain storming meeting, so, my test coverage is good.
4. I have got my Test cases reviewed by other sr. and Junior TE.
5. I have written Test cases by following procedures to write TC. method.
6. While executing TC if I come up with any scenarios and it is not present in any TC so, my TC coverage will be good.
7. I will do adhoc testing and this will help me to improve my Test coverage.
8. I have spent more time in doing system study. Because of that I got very good Project knowledge which helped me to write more no. of scenarios.
9. I have prepared Traceability matrix and ensured each and every requirement has got atleast 1 TC, so my Test coverage is good.

Where do you write or store Test case coverage?

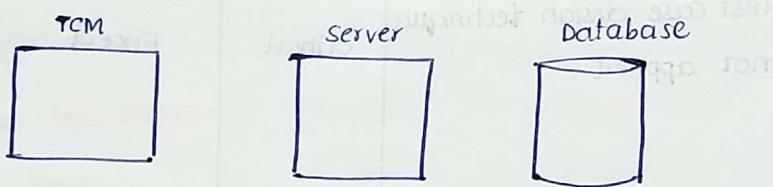
1. Test Case Management Tool

2. Ms/Excel / Ms Word

What is Test Case Management?

Test case management tool is a software which is mainly used to write or store Test cases in a centralised place in an organised way.

Test Case Management Tool.



✓ 1. QC (Quality Centre)

 ↳ ALM (Application Life cycle Management)

✓ 2. JIRA

✓ 3. Test link

4. TC-zilla

5. Test Rail

6. Test case Lab

7. QA complete.

? How many TC you write per day?

1. Depending upon complexity and type of TC, I used to write 6-8 TC

? How many TC you review per day?

1. Depending upon the complexity of req and size of TC, I will review 12 - 16 TC

How many TC you execute per day?

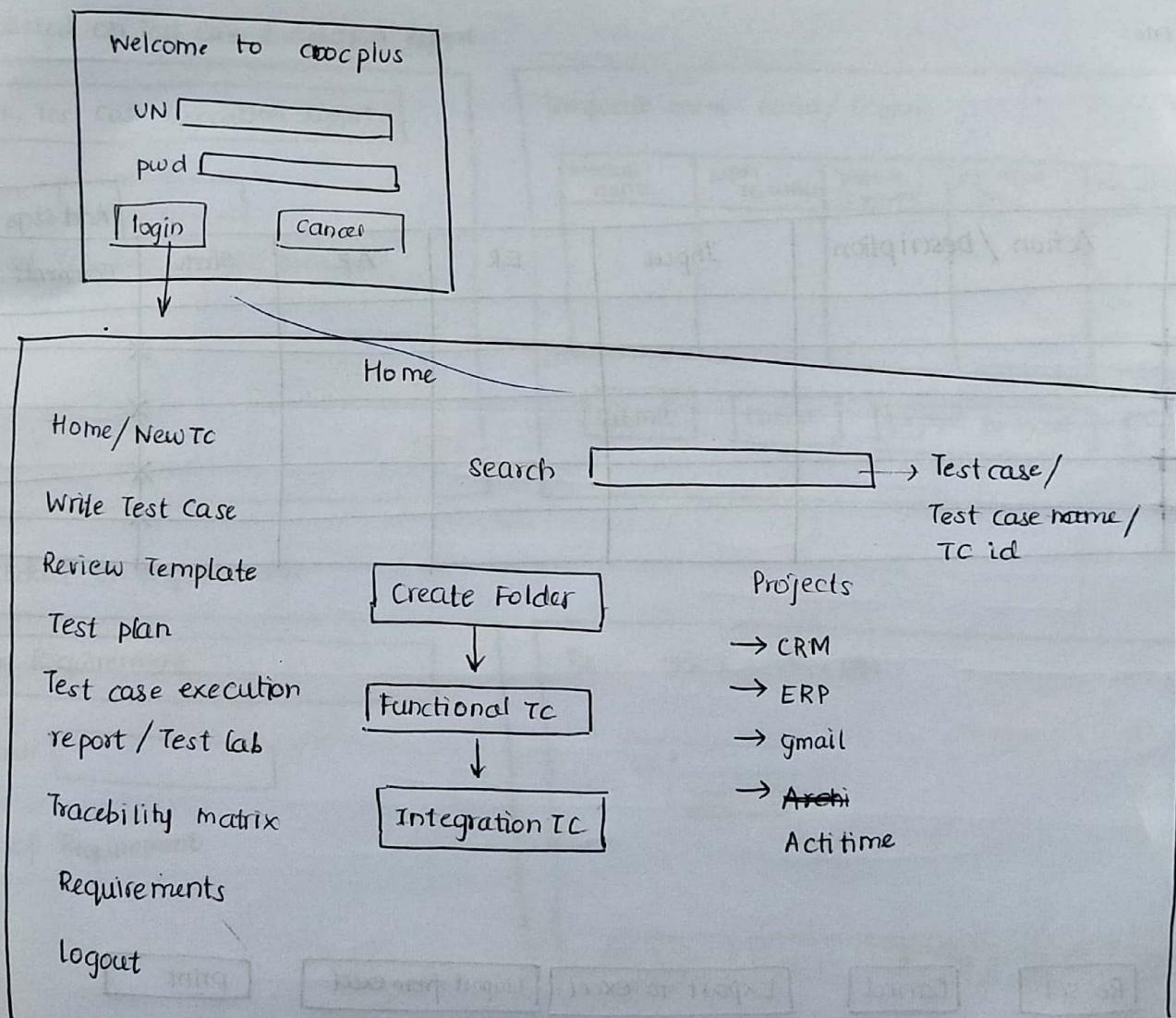
i. Depending upon the complexity of requirement, I will ~~execute~~ 18 - 24 TC

How Test case management tool works? Croc-plus / QC / JIRA

i. TL / TM / IT eng will give URL, UN, pwd. for the TE who raised for ticket.

Step 2.

Open browser enter URL : <https://servername-crocplus/login.jsp>

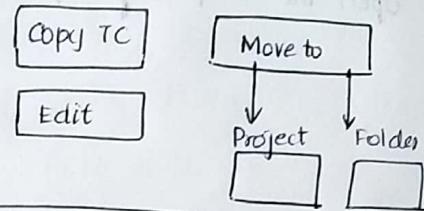


When Clicked Test case to write.

Test case name:

Test case id : Unique id

Test Scenarios

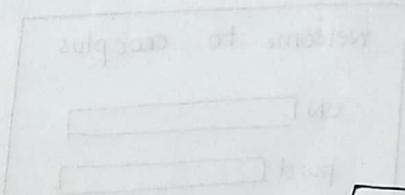


Test Case Name:	TC ID:	Category:	Priority:	Severity:	Test Status:	Test Result:	Comments:
Test Case Description:							
Test Case Details:							
Test Case Data:							
Test Case Notes:							

Test case name:

Project name:

Release name:



Step No.	Action / Description	Input	ER	AR	Status	Add steps	Comments
*						*	
*						*	
*						*	
*						*	

Author

Reviewed by:

Check status

Test Information

Test Details

Submit

Re-set

Cancel

Export to excel

Import from excel

Print

When clicked on Review Template.

Create Template

Search []

List of Templates.

1)
2)
3)
4)
5)

Step no	Testcase name	Step no	Reviewers	Comments	Severity	Author's comments
		NO				

Submit Cancel Export to excel Print

Unique id

When clicked on Test Case Execution Report

Create Test Case Execution Report

Search []

List of Templates for TC execution

1)
2)
3)
4)

Module name	Total TC written	Total TC executed	No. of TC pass	No. of TC fail	P% F%

Submit Cancel Export to excel Print

When clicked on requirements

Create Requirements

Search []

List of Requirements

1)
2)
3)

Re name = CBO - SRS

1. 1.1.

Submit Cancel Download to excel PDF Print

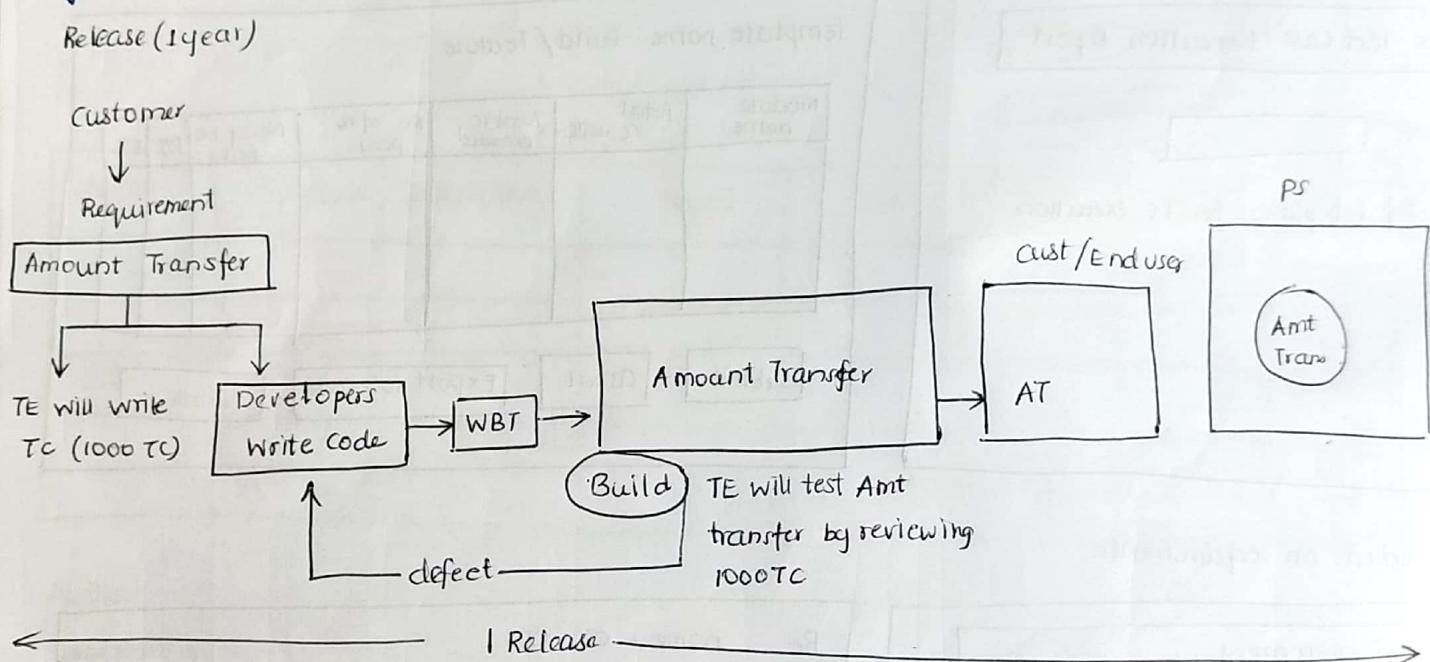
Regression Testing.

Testing the unchanged / old feature to make sure that changes like adding feature, deleting feature, modifying feature and fixing defects is not introducing any defect in the unchanged or old feature is called Regression Testing.

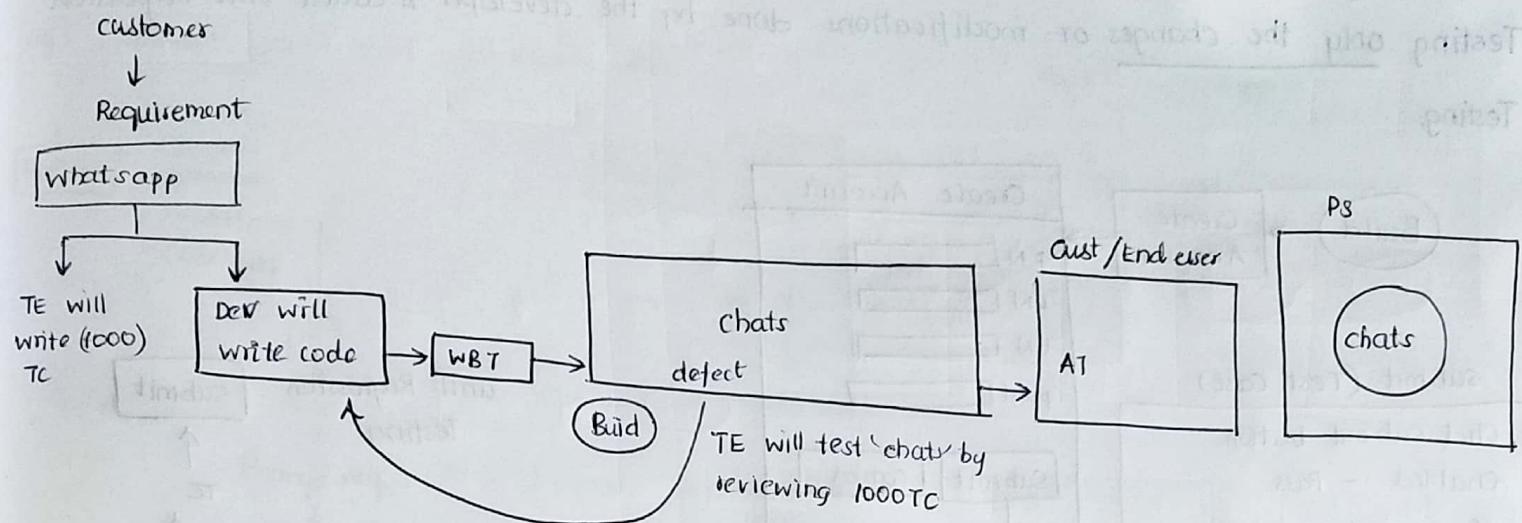
OR.

Re-execution of same testcases in different releases or builds or Test cycles to make sure that changes like adding feature, deleting feature, modifying feature and fixing defects is not introducing any defects in the unchanged or old features is called Regression Testing.

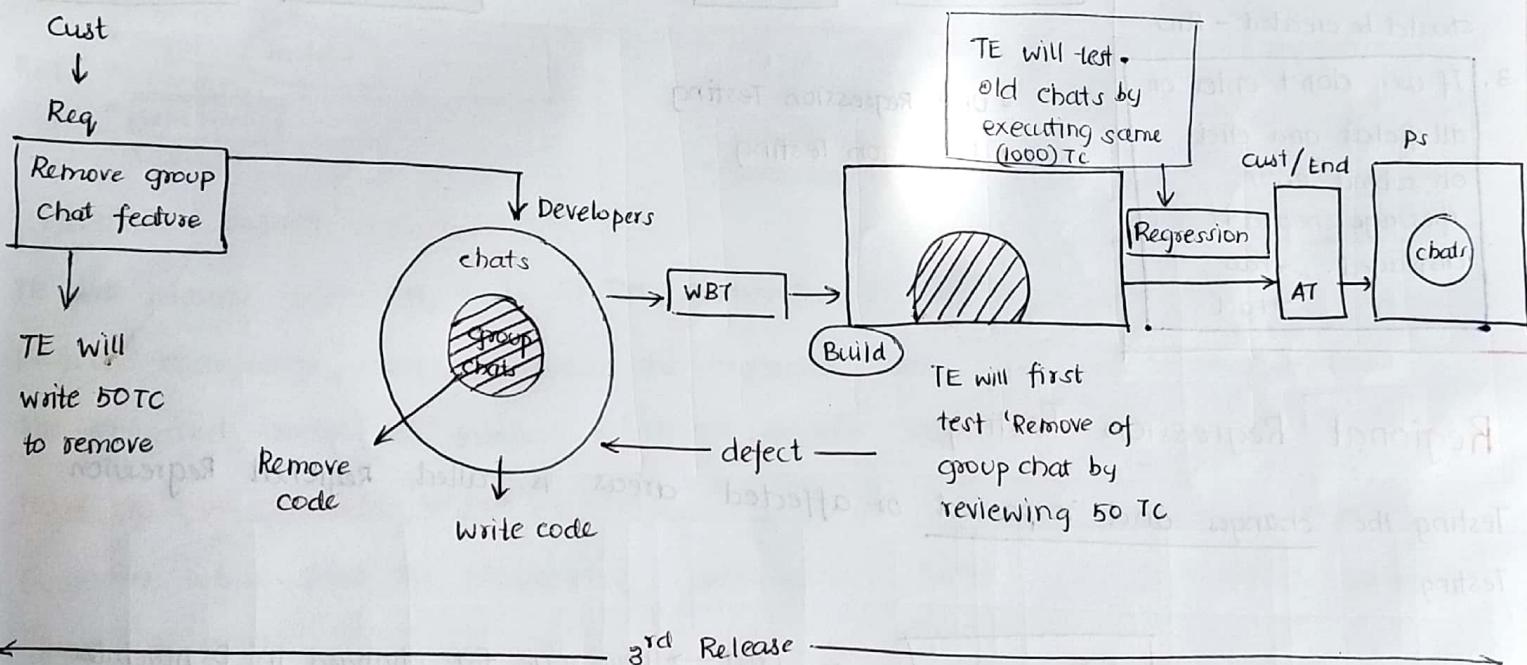
eg:



eg: Release (1 year)



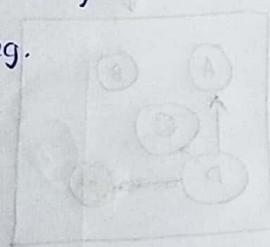
Release 3 (Remove group chats)



Types of Regression Testing.

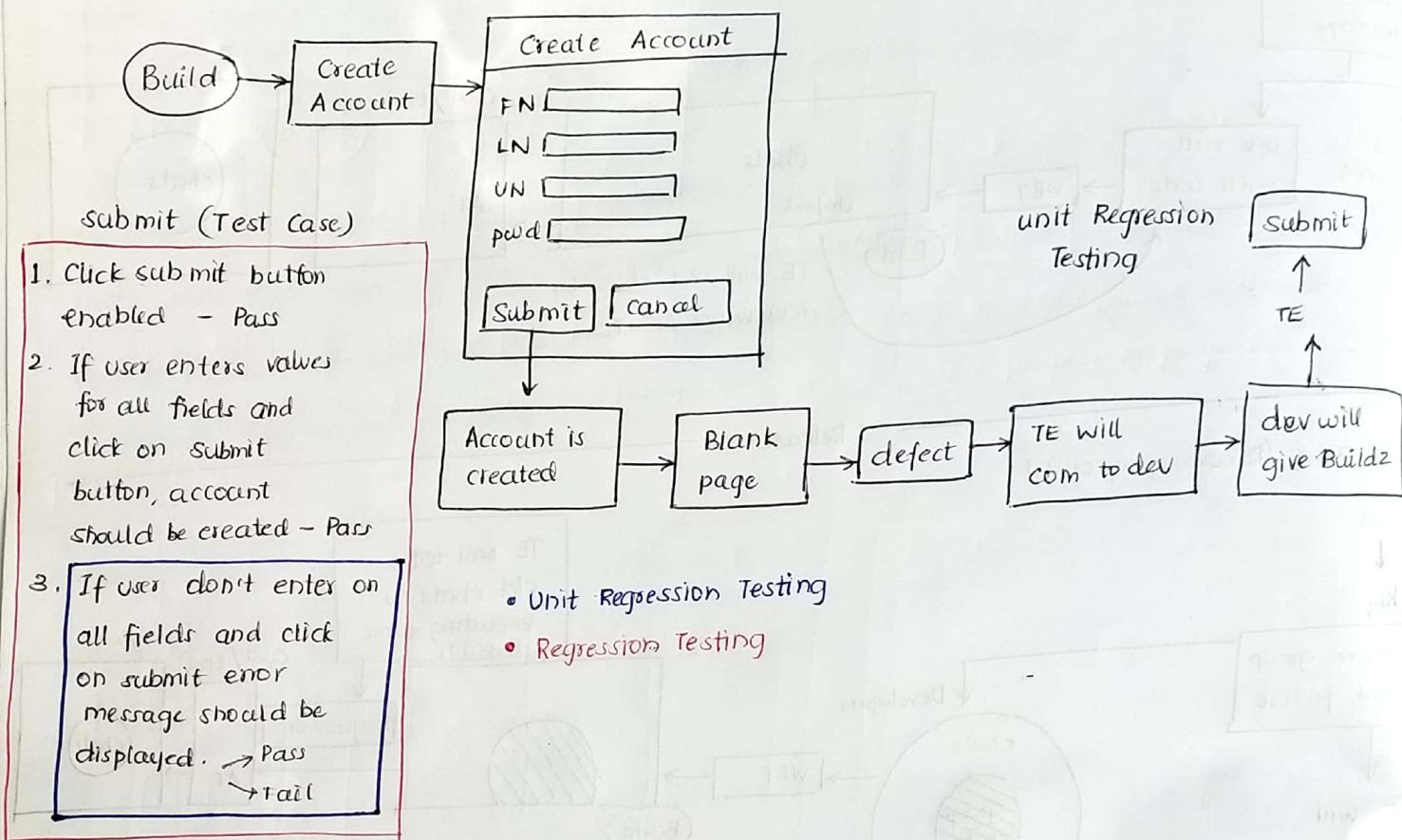
Depending upon the changes done by developer, Regression Testing can be done in 3 types.

1. Unit Regression Testing
2. Regional Regression Testing
3. Full Regression Testing.



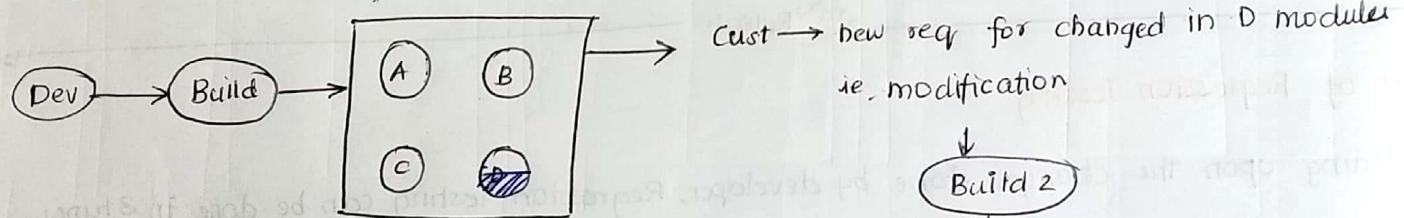
Unit Regression Testing

Testing only the changes or modifications done by the developer is called unit Regression Testing.



Regional Regression Testing

Testing the changes and impacted or affected areas is called Regional Regression Testing.



According to above eg: Dev will give Build1

and Build1 consist of A,B,C and D module.

TE will test the build1. Once Testing is completed customer will tell, you all should do changes for module D, Dev will develop the changes and give build2 to TE. Now, TE based on impact analysis

we come to know that, doing changes in mod D

is affecting X, D, C, A so, testing the changes (D)

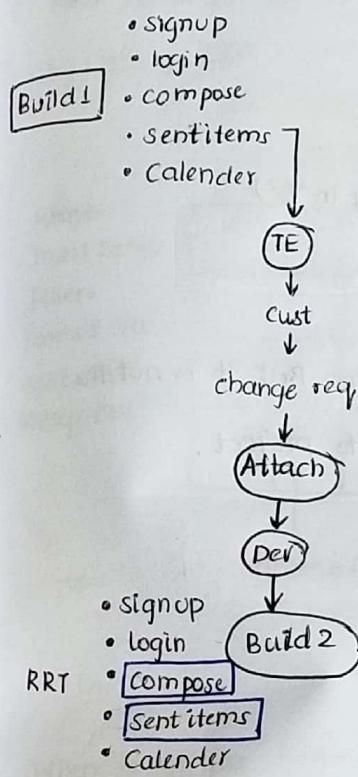
areas is called Regional Regression Testing.

only shaded area

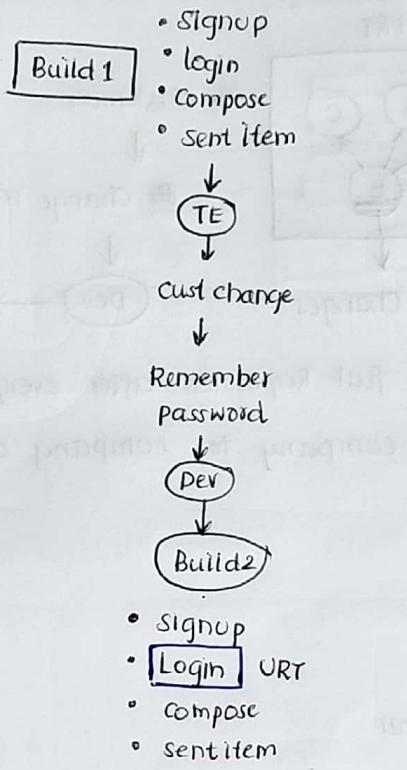
↓
Smoke, FT, IT, ST

Impact Analysis.

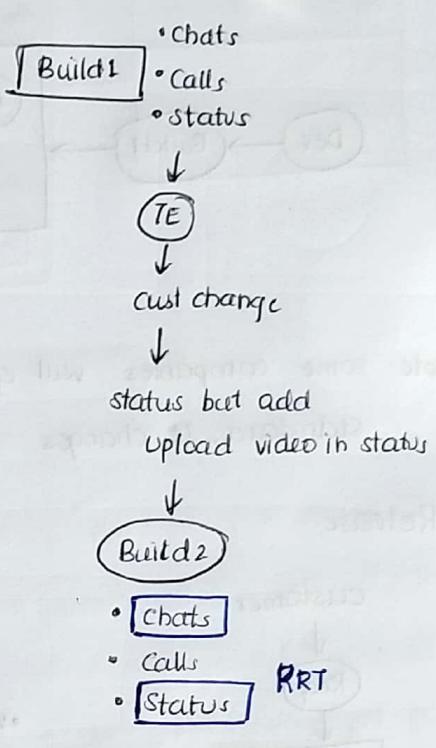
eg1: gmail



eg2: gmail



eg3: Whatsapp



What is Impact Analysis.

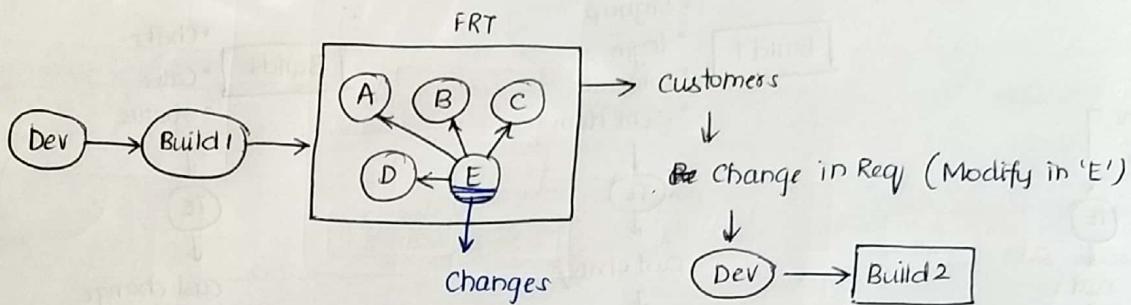
TE will interact with BA, Sr.TE, Sr.Dev, customers and a person who is having very good project knowledge, discuss about the impacted areas, gather the information, document the impacted areas. This meeting is called Impact Analysis.

How do you identify impacted areas.

- Customer when gives the requirement, customer only inform developers and TE about the impacted areas.
- When BA converts CRs to SRS, In the SRS, there will be a section called impacted areas, where the BA will mention list of affected areas.
- TE will interact with Sr. Dev where in dev will give list of impacted areas because, he is the person who is doing code changes. He will be aware of all the areas, where code is modified.
- TL, TM and Sr.TE based on their product knowledge, they will give list of impacted areas. and TE will conduct impact analysis meeting.

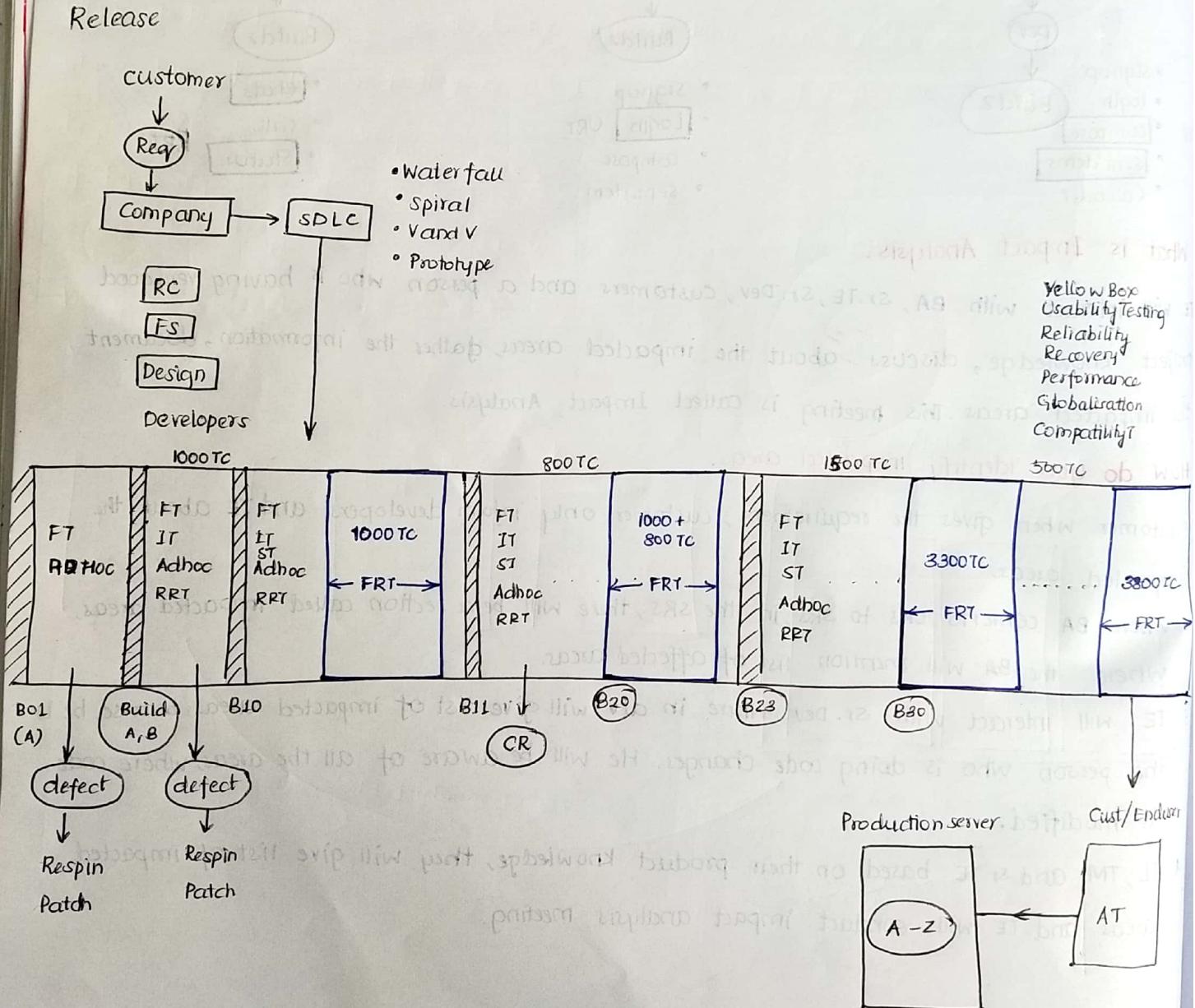
Full Regression Testing

Testing the changes and all the remaining features is called Full Regression Testing.



Note: some companies will go for full Regression after every 10 or 12 cycles. But it is not standard. It changes from company to company and project to project.

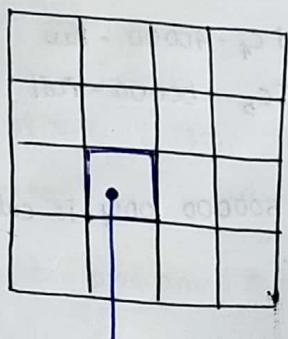
Release



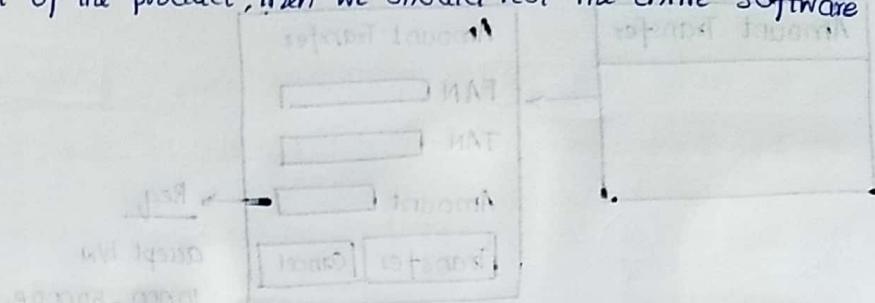
When to go for FRT?

1. When changes are more we should go for FRT.
2. When changes are done to the Root of the product, then we should test the entire software by doing FRT.

Graph
Insert Table
Filters
Format cell
Split cell
Merge cell



Changes for cell
cell (Root cause of
product)



When to go for Regression Testing?

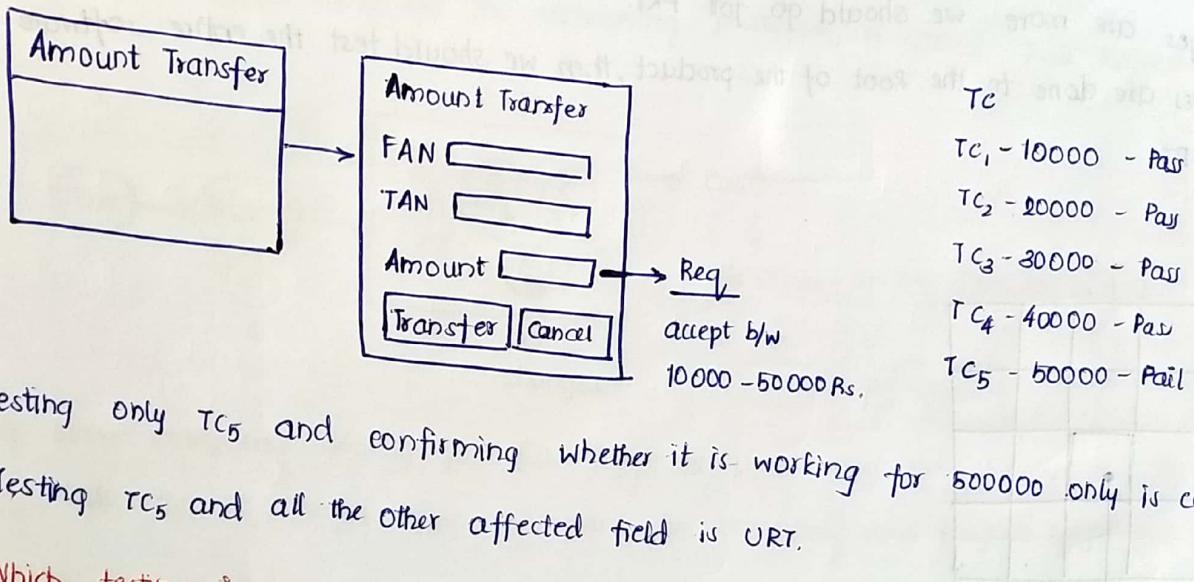
- When changes are done to the software.
- When software is changed.
- When platform is changed.

1st What is difference between Retesting and Regression Testing?

(Confirmation Testing) Retesting	Regression Testing
<ul style="list-style-type: none">• Whenever dev give the build, check whether defect is fixed or not is called Retesting / confirmation Testing.• It is done for failed TC [eg: Compose 90 TC pass 10 TC fail ↓ defect]• Retesting is planned testing (pick some TC and test)• Here we don't go for automation• Retesting takes highest priority	<ul style="list-style-type: none">• Testing the old feature to make sure that changes like adding, modifying, removing feature and fixing defects are not introducing any defect in the old feature is called Regression Testing.• It is done for pass TC.• Regression is generic testing• We go for Automation.• takes lesser priority.

Is there any difference between Retesting and Unit Regression Testing.

Yes.



Testing only TC₅ and confirming whether it is working for 500000 only is called Retesting.
Testing TC₅ and all the other affected field is URT.

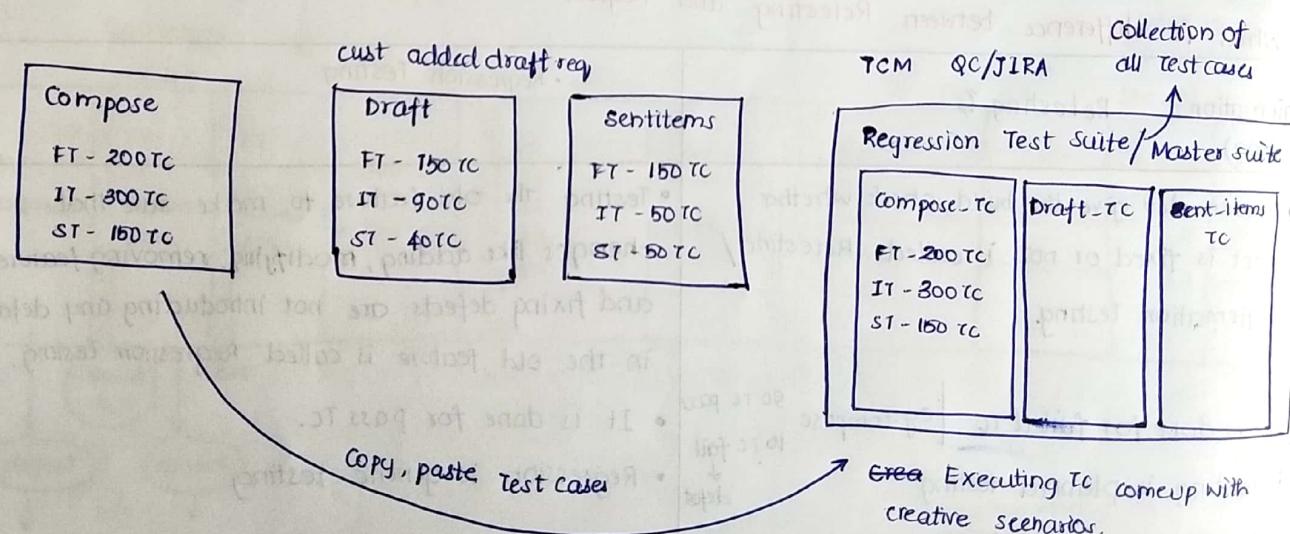
Which testing is acting as a bridge between Manual and Automation Testing? Regression Testing.
Which type of Regression testing is called Preferred type of Regression Testing?
Regional Regression Testing.

If by doing URT - we might miss some defects.

By doing FRT - we will take more time. So, we prefer RRT.

Do you write Regression Scenario's and Test Cases?

Yes, we write TC.



What are the disadvantages / drawbacks of doing Regression Testing Manually.

OR

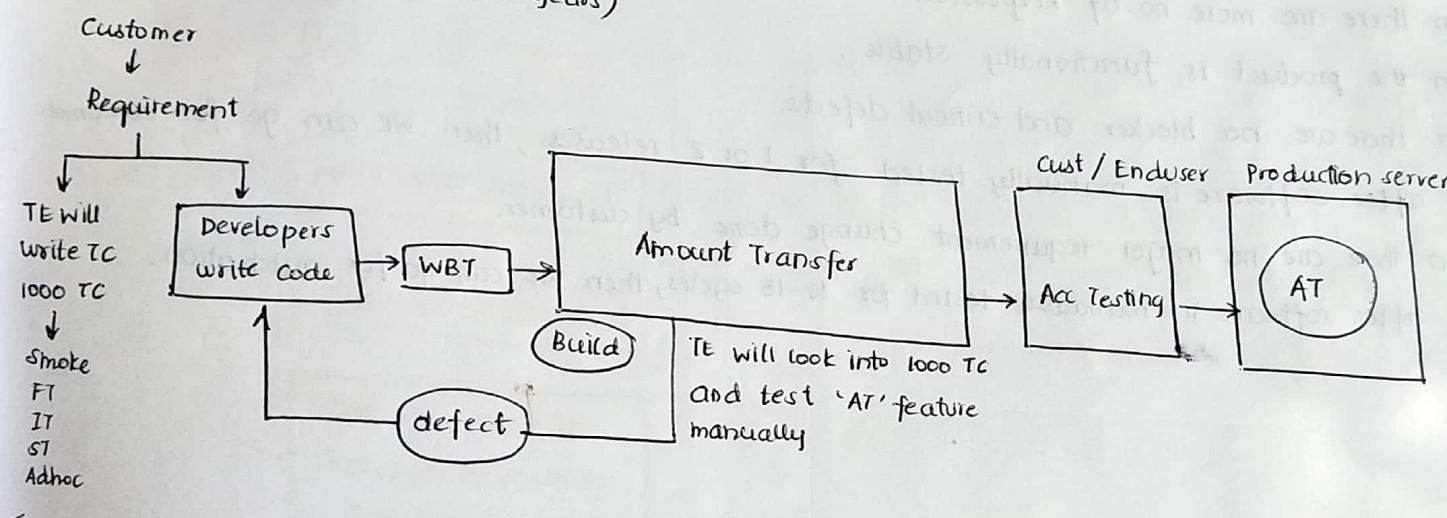
Drawbacks of Manual Testing / Black Box Testing.

1. Time taken is more
2. More resource utilization
3. Monotonous job
4. Tedious job
5. No consistency. (same TC executing for 1000 times, TE will not test 1000 times, so no consistency)
6. Less Accuracy.

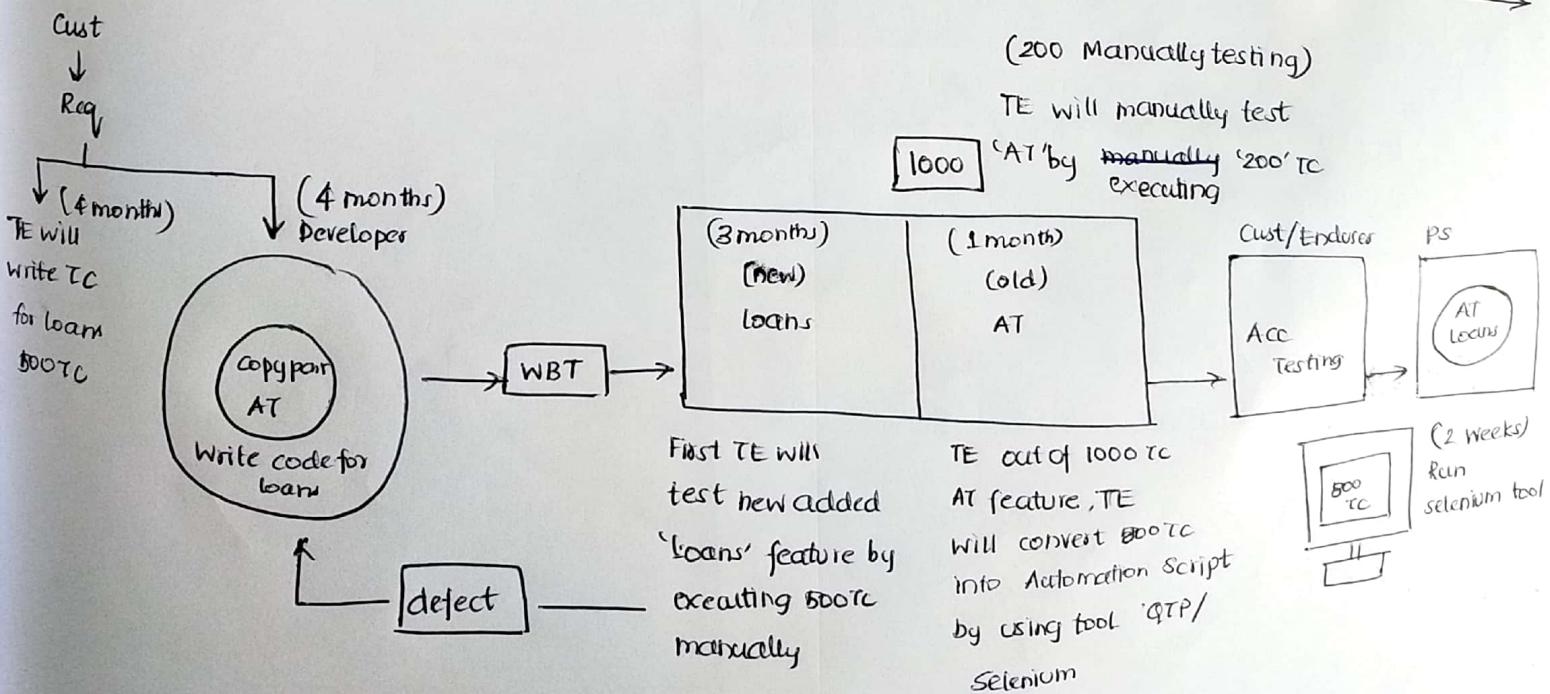
In order to overcome the drawbacks, company will go for Automation.

How Automation works?

Release 1 (Amount Transfer) (1.5 years)



Release 2 (Loans)

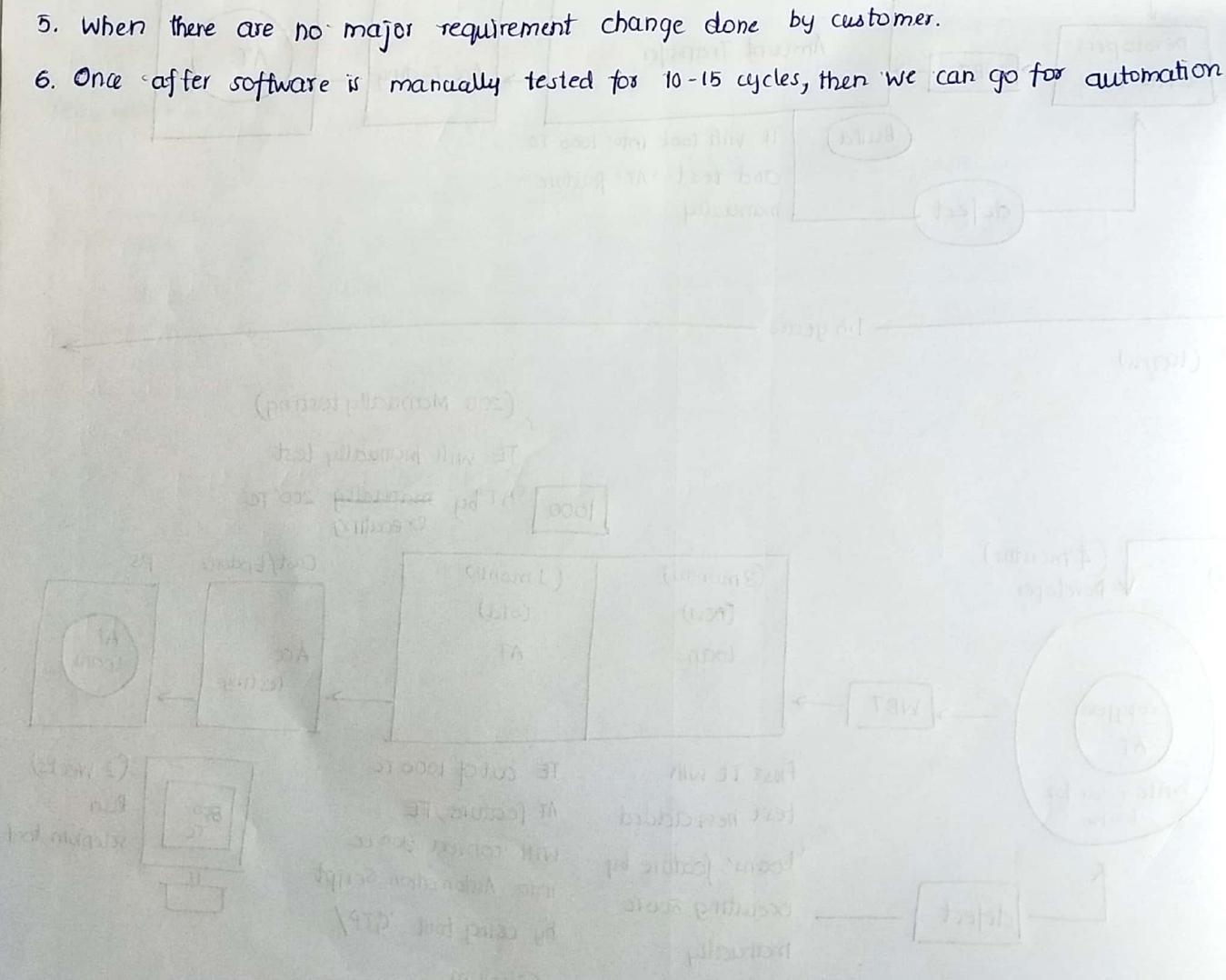


Why we should not do automation at the initial stage itself?

1. In the early stage of product development, our aim is to catch more no. of defects. So, we should do manual testing first because, Automation tool can't think by itself.
2. In the early stage of product development, more defects will be predicted. If we go for Automation, we will not be able to complete script execution.
3. In the early stage of product development, customer keeps on doing lot of requirement changes, here updating manual TC will be easy. In this stage, if you go for automation, we will be spending time only in modifying the script rather than testing the application.

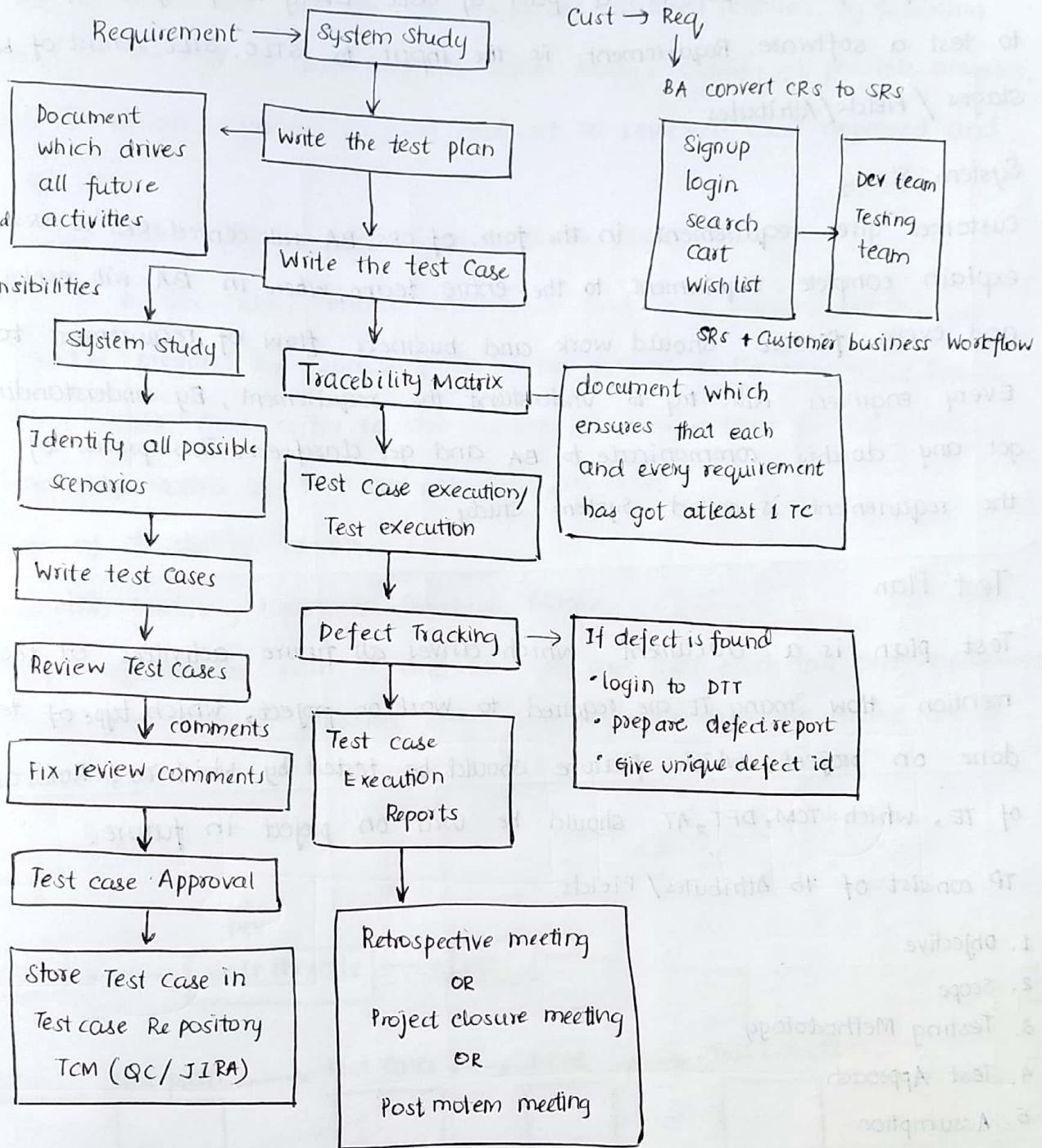
1st* When to go for Automation.

1. When there are more no. of Regression TC.
2. When the product is functionally stable.
3. When there are no blocker and critical defects.
4. Once after software is manually tested for 1 or 2 releases, then we can go for automation.
5. When there are no major requirement change done by customer.
6. Once after software is manually tested for 10-15 cycles, then we can go for automation.



Software Test Life cycle / Test Life Cycle / Software Testing Process.

- How many TE are required
- Which testing
- Which feature should be assigned to which TE
- Roles and Responsibilities
- Which TCM tool
- Which DFT
- Which AT and 15 attributes



It is a step by step procedure to test a new software. STLC stands for Software Test Life cycle. STLC is a part of SDLC. Every company will follow STLC in order to test a software. Requirement is the input to STLC. STLC consist of below mentioned stages / Fields/Attributes.

System Study

Customer gives requirement in the form of CRS. BA will convert CRS to SRS and BA will explain complete requirement to the entire team, where in BA will explain how each and every feature should work and business flow of requirement to entire team. Every engineers will try to understand the requirement. By understanding if you get any doubts communicate to BA and get clarified. This process of understanding the requirement is called system study.

Test Plan

Test plan is a document which drives all future activities of testing. Here we mention How many TE are required to work on project, which type of testing should be done on project, which feature should be tested by which TE, Roles and Responsibilities of TE, which TCM, DFT, AT should be used on project in future.

TP consist of 15 Attributes/ Fields.

1. Objective
2. Scope
3. Testing Methodology
4. Test Approach
5. Assumption
6. Risk
7. Back up plan / Mitigation plan / contingency plan
8. Roles and Responsibilities
9. Scheduling
10. Defect Tracking
11. Test environment / Test Bed
12. Entry and Exit criteria
13. Test Automation
14. Deliverables
15. Templates.

Test Case

Here TE will start to write test cases for their respective assigned features, by following procedures to write TC. For that, we will first do system study, identify all possible scenarios, write TC by apply TC design techniques in TCM and get TC reviewed and approved and store TC in TCM tool.

* Requirement Traceability Matrix (RTM) / Cross Reference Matrix (CRM)

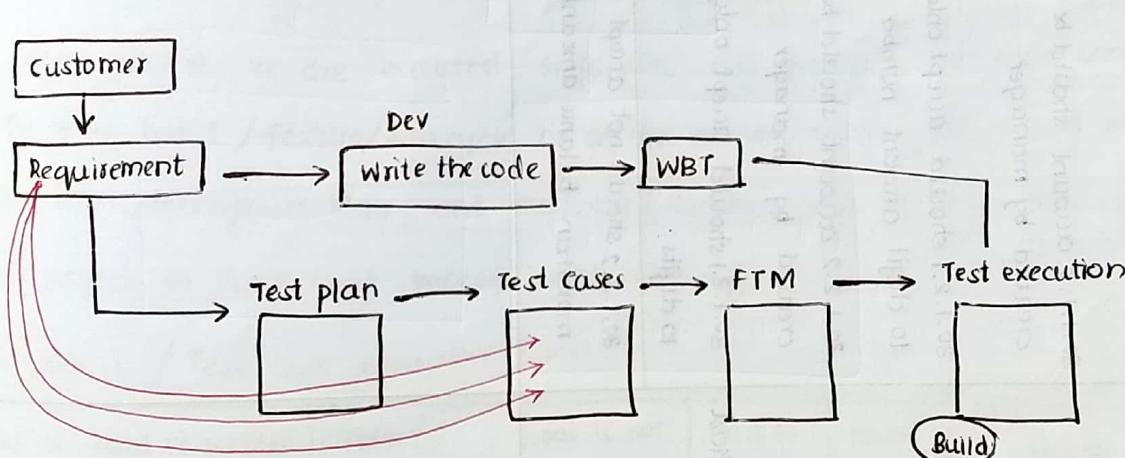
Traceability Matrix is a doc which ensures that each and every requirement has got atleast 1 TC. TL/TM prepares it, suppose if TC is not present for the requirement then TL will mark the requirement. Once after TC are written, TL will unmark the requirement.

We prepare Traceability Matrix in TCM tool, MS excel, MS word.

There are 3 types of Traceability matrix.

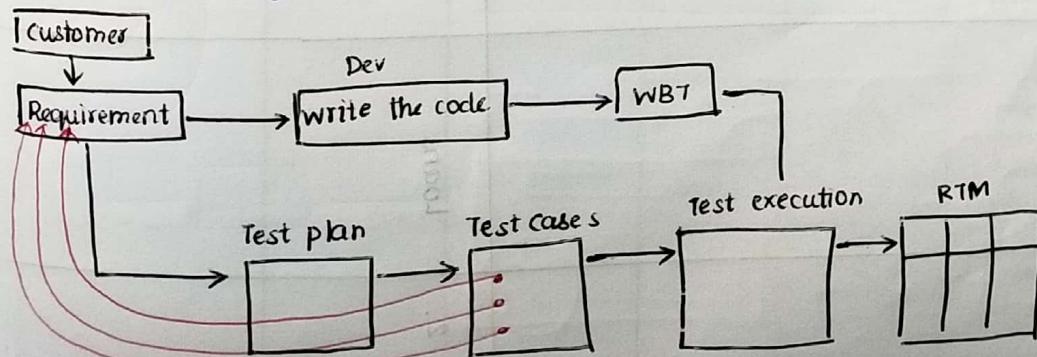
1. Forward Traceability Matrix / Horizontal Traceability Matrix.

Here we map requirements with TC and we make sure that each and every requirement has got atleast 1 TC. This is prepared before test execution. Here we ensure are we building product right.



2. Backward Traceability Matrix / Vertical Traceability Matrix

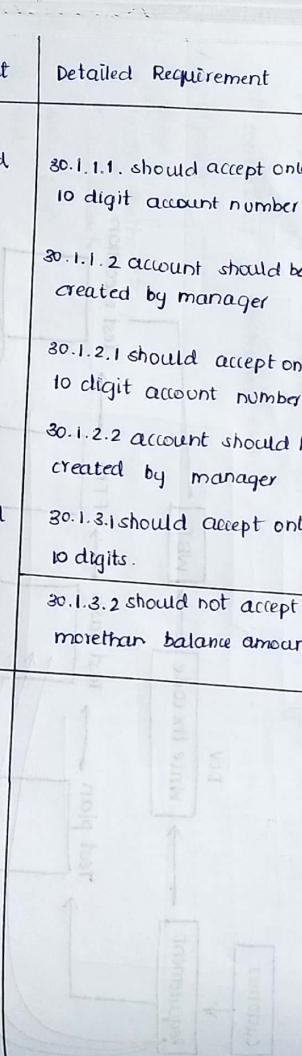
Here we map TC back to the requirement after Test execution and we make sure that product which developed, is it really developed according to requirement. Here we ensure that are we building right product.



3. Bi-directional Traceability Matrix

It is a combination of both forward and reverse traceability matrix.

Traceability Matrix Template

Sl.no	Module name	High level Requirement	Detailed Requirement	Test casename	Automation Script name	Execution type
1.	Amount Transfer	30.1.1 FAN text field 30.1.2 TAN text field 30.1.3 Amount textfield	30.1.1.1 should accept only 10 digit account number 30.1.1.2 account should be created by manager 30.1.2.1 should accept only 10 digit account number 30.1.2.2 account should be created by manager 30.1.3.1 should accept only 10 digits. 30.1.3.2 should not accept more than balance amount	CBO_AT_FAN_TC1 CBO_AT_FAN_TC2 CBO_AT_TAN_TC3	CBO_AT_FAN_SCRIPT1 CBO_AT_FAN_SCRIPT2	Automation Automation Manual
2.	Loans					

2. Test coverage will be good
3. We will get to know which should be executed manually and automation.
4. Suddenly if customer change the requirement, we should do traceability matrix.

Test Case Execution

By this time developer will give build or software to TE and TE will look into TC and test the build or s/w. Where in TE will do Smoke, FT, IT, ST, Adhoc, Regression etc by looking into the TCs. Since testing is done by looking into TC, this is called as Test execution / Test case execution.

Defect Tracking

While testing s/w if TE finds any defect, then TE will immediately login to DTT, prepare defect report by giving unique defect ID. and TE will communicate defect to developers. Developer will fix the defect and send the new build to TE. and TE will retest the defect. If defect is really fixed, then TE will close the defect. Suppose if, defect is not fixed, then TE will reopen the defect. This process is called defect tracking.

Test case execution Report / Test case summary Report

Once after all the TC are executed, SR.TE / TL will generate Test case execution report. for every build / feature / test cycle or at the end of release. This report will be sent to testing team, development team and customer (customer who works on time and material build project or time and money based company).

Build Report. / Test Case execution Report.

Module Step no.	Total TC Written Testcase name	Total TC executed	Total TC not executed	No.of TC pass	No.of TC fail	Pass %	Fail %

Retrospective Meeting / Project closure meeting

It is a meeting conducted by TM on the last day of the release. TM will invite all the TE who has worked on the project and discussed about the achievements and mistakes while working on the project.

Achievements	Mistakes.
<ol style="list-style-type: none">1. Test cases were reviewed by peers and I got many inputs because of this my test coverage was good.2. We did Brainstroming meeting which helped us to improved scenario coverage.3. We did impact analysis meeting which helped to identify impacted areas.4. Dev fixed blocker fixed - very fast, because of this I was able to complete testing in the planned time.5. We swapped modules between the TE and did adhoc testing which helped us to find more no.of defects.	<ol style="list-style-type: none">1. Dev didn't fix blocker defects ontime which delayed our testing.2. There was no proper communication b/w TM.3. Requirements were not clear.4. TC Review didn't happen properly.

? What are the levels of testing?

unit testing



Integration Testing



System Testing



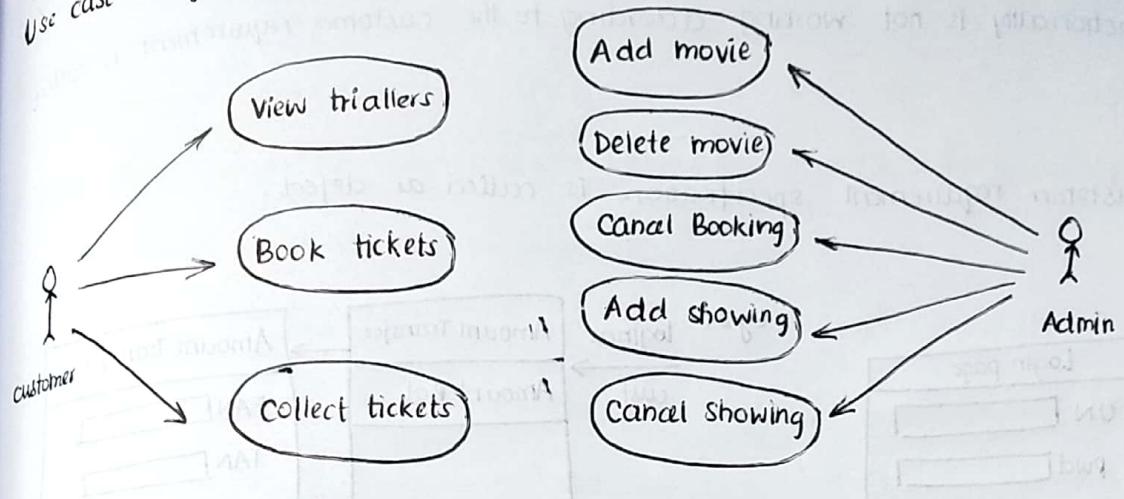
Acceptance testing.

? What are the types of Requirement collections?

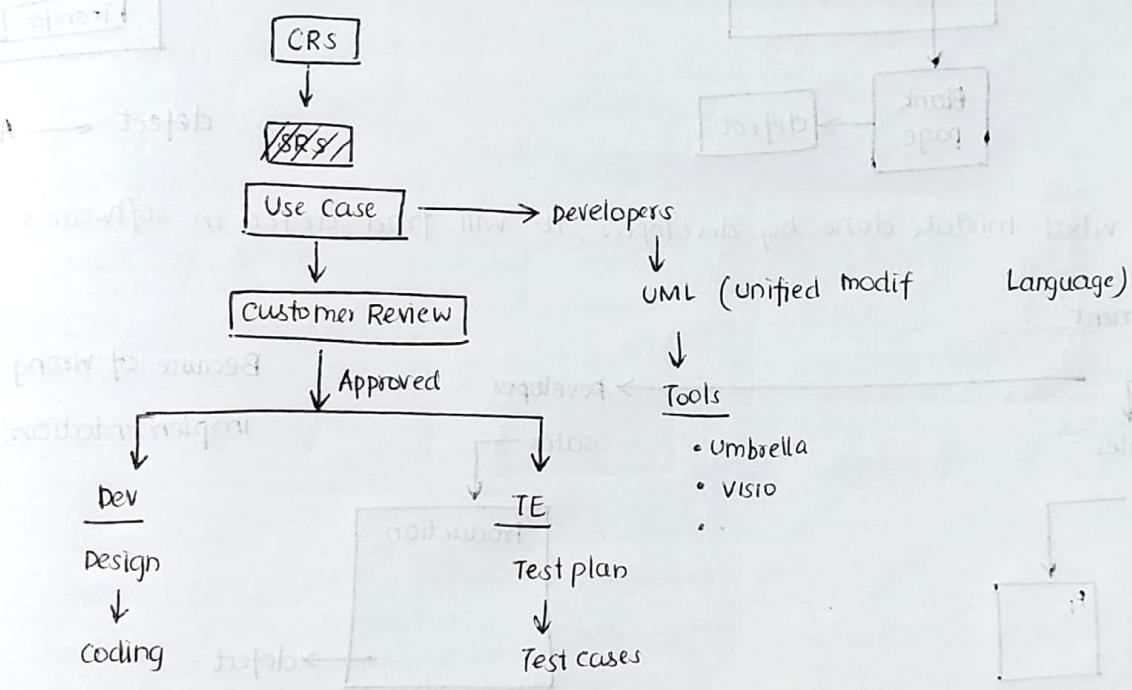
- 1) CRs
- 2) SRs
- 3) FS
- 4) Exploratory Testing.
- 5) Use cases.

- It is the pictorial representation of the requirements how end user interacts with the application.

Use case diagram for Book my show



Who will prepare Use case?



What is difference between use case and prototype.

Use case

- Use case is a pictorial representation of the requirement
- Here we get to know how end user interacts with the application.

Prototype

- It is the exact image of the application.
- Here we don't get to know how end user interacts with the application.

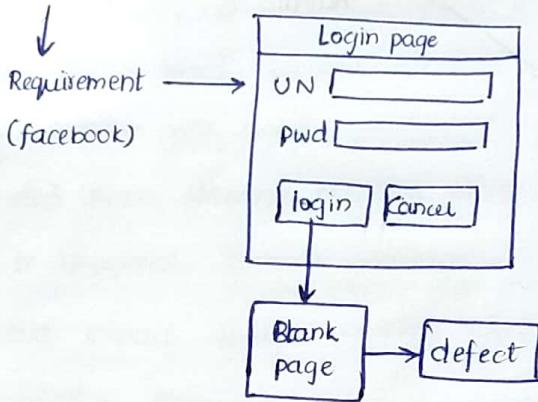
Defect.

If a feature or functionality is not working according to the customer requirement as defect

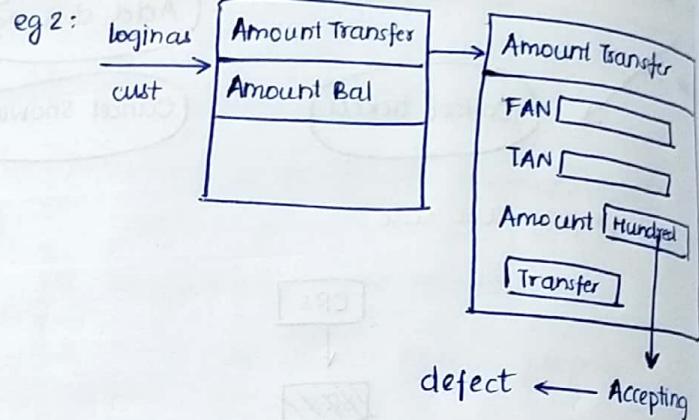
OR

Deviation from the customer requirement specification is called as defect.

eg: 1 Customer

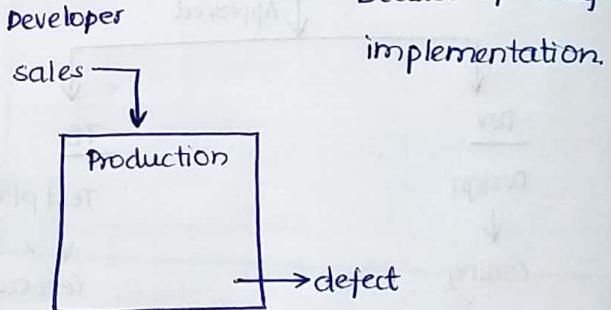
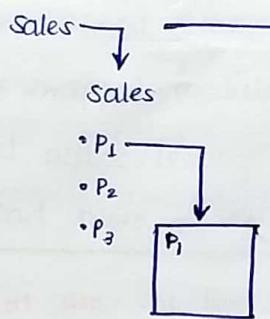


eg: 2:

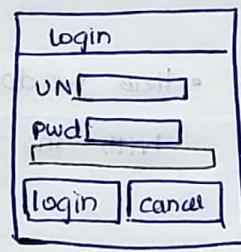


? Because of what mistake done by developers, TE will find defect in software?

1) Requirement

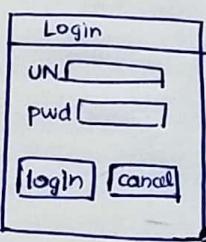


2) Because of missing implementation.



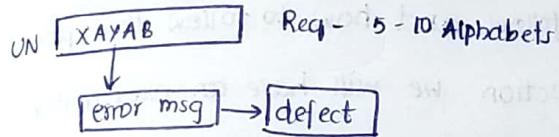
3) Because of Extra implementation.

Requirement → Developer



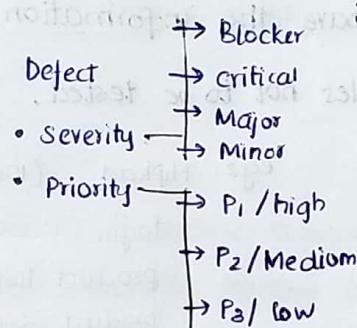
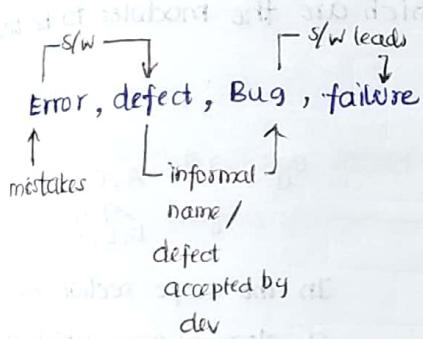
What is the defect between Error, Bug, defect, failure.

1. Mistake done in the code, because of this we will not be able to compile the code or run the code is called error.
2. Error found in the software or application is called as application defect.



3. Bug is an informal name given to defect. OR defect accepted by developer is called Bug.

4. Many bugs/defects present in the software leads to failure. Failure is a word used by End users.



What is severity?

Impact of defect on the customer business workflow.

1. Blocker / Shows stopper
2. critical
3. Major
4. Minor

Test Plan

Test plan is a document which talks about future testing activities. (OR) Test plan is a document which derives future testing activities.

Test plan will be prepared by TL, with the help of TE, the TP will be reviewed by Test Manager [It can also be reviewed by project manager, scrum master, customer. For 100%, The contribution of TL for preparing the TP will be 60%, TE contribution will be 20%, TM will be 20%. Each and every company will prepare test plan before starting the project. In Test plan, we have 15 Attributes.

1. Objective
2. Scope
3. Test Methodology
4. Test Approach
5. Assumption
6. Risk
7. Back up plan
8. Defect Tracking
9. Entry and Exit criteria
10. Test Bed / Test Environment

6. Risk
7. Back up plan
8. Roles and Responsibilities
9. Scheduling
10. Defect tracking.
11. Test bed
12. Entry and exit criteria
13. Test Automation
14. Deliverables
15. Templates.

Objective

In this section we talk about the main aim of preparing the Test plan and testing project. The word aim means which process and procedure we are going to follow give a quality software to the customer. [In this section we will have a clear explanation of which process we are going to follow and how to follow that process for eg: If the process is Agile, then in this section we will have an explanation how to follow agile.]

Scope

In this section, we will have the information like which are the modules to be tested and which are the modules not to be tested.

eg: Gmail

- 1. Login
 - 2. Compose
 - 3. Sent items
 - 4. Inbox
 - 5. Help
- } should be tested → not to be tested

Customer will give the requirements for login, compose, sentitems, inbox and help features, Dev will develop all these features, The help feature will be developed and tested by Content developers or technical writers.

It is already tested by content writers, again it will not be tested by TE.

feature. After purchasing the cart feature, It will be integrated to the flipkart project, since the cart feature is already in the market, so we will not test that in the flipkart project. We might just test the integration between cart and other features.

eg2: Flipkart (10m)

- login
- gm
- product list
- product details
- payment option
- Cart

Flipkart is a customer who will give requirements for developing the modules like login, product list, product details, payment option, Cart. To develop and test all these modules, customer will also give the deadline of 10 months and the budget will be of 10 Lakh, company will develop and will test the login, product list, product details, payment option in 9 months. Now we have only

1 month. for the developing and testing of cart feature. The dev

will give the estimation of 2 months for the development of cart feature, if we go with this strategy, we may be in delay in releasing the software. Now the company will contact Amazon and will try to purchase the cart

eg3: 1st A,B,C
2nd D,E,F

In the scope section, we a clear picture which are the 1st release which are features to be tested, in 2nd release which are the features to be tested, if project is in Regression stage which are the features to be tested and not to be tested.

Testing Methodology.

In this section, we will get an information like what type of application we are going to test and what type of testing we are going to perform in the application. In this section, all the testing terminologies will be documented. Which will help the developers and the management team so, that they will get an idea

e.g.: Project name : Shaadhi.com

Type of application : web based application

Types of Testing performed on this app : Smoke, ST, FT, IT, Adhoc, Regression, compatibility, Globalisation, web security, performance recovery, Usability testing.

e.g.: Project name : Print

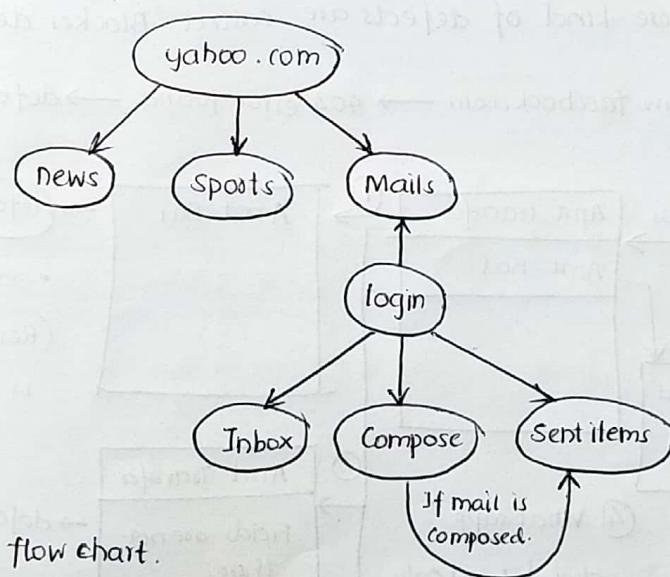
Type of application : stand alone application

Type of testing performed : Smoke, FT, IT, ST, adhoc, Regression, recovery, compatibility, reliability, installation testing and uninstallation.

Test Approach

In this section, we will get the information on how to test the application. Some of the approaches for testing the application is testing the application only by using scenarios, testing the application by using only TC, testing the application by using TS and TC, testing the application by using flow chart.

e.g. for flowchart. (10%)



Advantages of flowchart.

1. Writing flowchart is easy.
2. Flowcharts are easy to understand.
3. Flowcharts will have a good coverage.
4. The time taken to write the flowchart and test the application by referring the ~~app~~ flowchart is less.

Disadvantage for flowchart.

Changes are there we might miss some navigation flow.

Assumption.

While preparing the test plan, the TL will be having some set of assumptions, some of the assumptions are:

1. Assumption from resource point of view.
2. Assumption from Technical resource point of view (Project Management tool, DTT)
3. We assume that we get a proper support from the development team.
4. We assume we get a proper Knowledge Transfer (KT) section.
5. We assume that we get a proper supporting document for testing.
6. We assume we get a proper response from the development team for any of the requirement changes.

Risk.

If assumptions fails we face risk.

Back up plan.

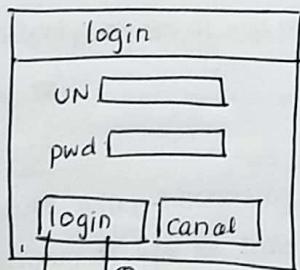
Continuation of defect.....

Blocker

Assume that there is a defect in the application and because of this defect, TE will be able to continue the testing for the respective module. And TE will be completely blocked in testing that module those kind of defects are called Blocker defect.

i) gmail.

www.gmail.com

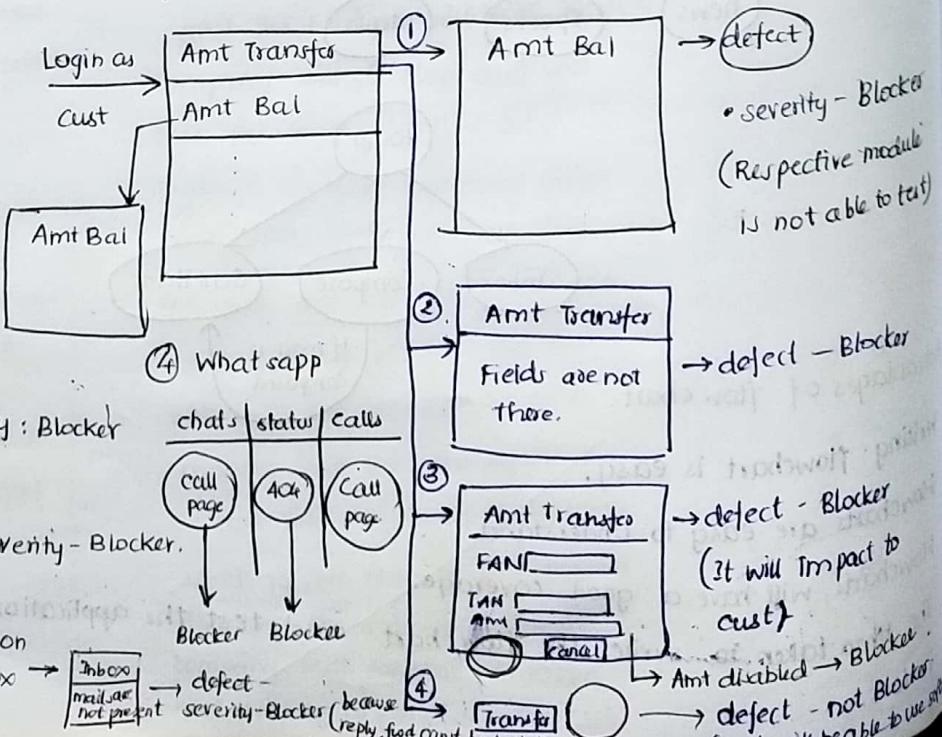


①

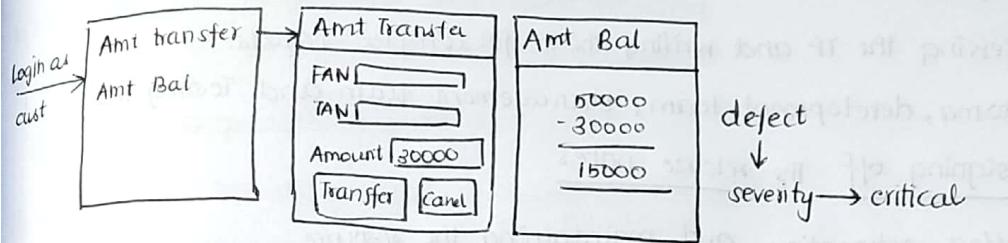
Blank Page → defect → severity: Blocker

② Invalid pwd → defect → severity - Blocker.

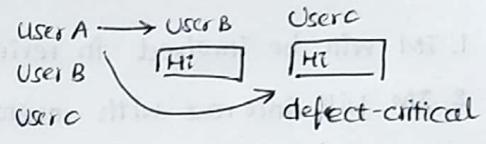
③ login as userA → Compose a mail and send to B → logout, → click on inbox → inbox mail not present → defect - severity - Blocker (because reply, fwd can't be done)



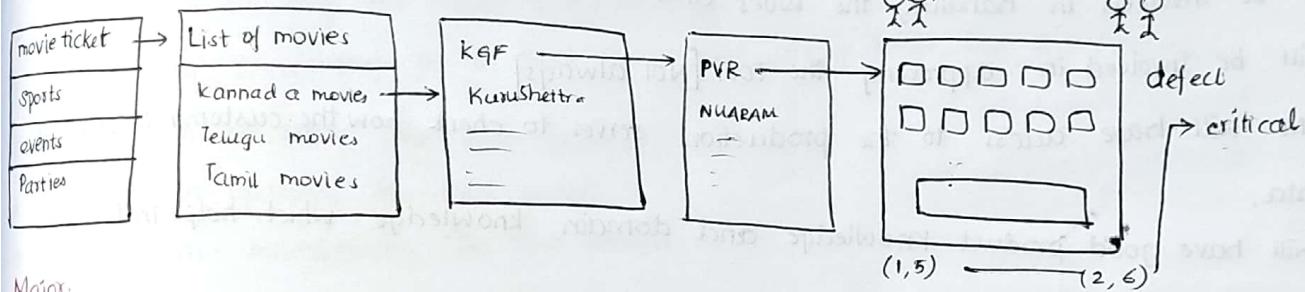
Critical
 Assume that there is a defect in the application and we are 100% sure that this defect will affect the customer business workflow, but here TE will not be blocked, we can still continue the testing, these kind of defects are called critical defects.



WhatsApp



Book my show

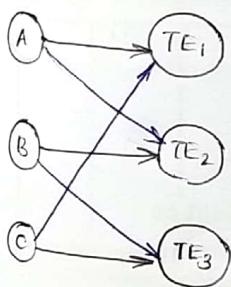


Major

Assume that there is a defect in the application and we really don't know how this defect is going to impact the customer business workflow. These kind of defects are called Major defects.

Continuation of Test plan.

Back up plan / Mitigation Plan / Contingency Plan.



In order to avoid the risk from 100% to 20%, we will be having the backup plan, all the Back up Plans will be mentioned in this section. In the section called as assumptions, TL will document all the assumptions with respect to testing activities. For each and every assumption what are the risk will be documented in risk section.

To overcome each and every risk, the backup plan will be mentioned in the

Back up plan section. With the help of back up plan section, we can reduce the risk from 100% to 20%.

e.g. Example for Assumption Risk and mitigation plan.

In the above diagram there are 3 modules i.e., A, B and C. Each module is assigned for the respected TE, i.e., (A) will be tested by TE₁, (B) will be tested by TE₂, (C) will be tested by TE₃. In this scenario, the assumption is all the TE will be in the company till the end of the project. The assumption might fail i.e., if any 1 TE quits the job, then that respected module will not be tested completely, because of this we might face the risk. To overcome this risk, the backup plan is one module will be assigned for 2 TE. i.e., For TE₁, (A) is a primary

module and (C) module is 2° module. For TE_2 , (A) is 2° and (B) is 1° and for TE_3 , (C) is 1°, (B) is 2°. This will be the Back UP plan.

Roles and Responsibilities.

Roles and Responsibilities of TM

1. TM will be involved in reviewing the TP and writing the TP for complex projects.
2. TM will interact with customer, development team, Management team and Testing team.
3. TM will be involved in signing off the release notes.
4. TM will be involved in effort estimation and maintaining the resource.
5. He will be involved in handling the issues and escalation.
6. TM will be involved in approving the TCs. [Not always]
7. The TM will have access to the production server to check how the customer is creating the data.
8. TM will have good product knowledge and domain knowledge which helps in to handle the project.

Roles and Responsibilities of TL

1. TL will be involved in writing and reviewing the TP.
2. TL will interact with dev team, testing team, management team and customers.
3. TL will be involved in assigning the task to the TE.
4. TL will be involved in tracking the activities of the TE and will make sure the work is completed or not.
5. TL will be involved in handling TC review.
6. TL will be involved in approving the TC.
7. TL will be involved in testing complex features.
8. TL will be involved in consolidating the reports.
9. TL will be involved in preparing Traceability matrix template.

Roles and Responsibilities of TE / What are the characteristics of TE.

1. Involved in system study
2. Identifying all possible scenarios
3. Conducting Brain stroming meeting and updating the scenarios.
4. Converting TS to TC by using TC design technique in TC template.
5. Involved in reviewing TC.
6. Giving review comments and fixing the review comments.
7. Involved in executing the TC.

6. Involved in identifying the defects and communicating defects to dev by using DTT.
7. Involved in tracking the defect.
8. Involved in selecting the TC for Regression Testing.
9. Involved in updating the TC, whenever req are getting changed.
10. Involved in conducting Bug triage Meeting.
11. Involved in performing Traceability matrix.

Roles and Responsibilities of AT.

1. Involved in installing and uninstalling the build.
2. Involved in selecting the TE for automation
3. Involved in selecting the Atool. (not always)
4. Involved in converting TC to Test scripts by using tools like QTP/Selenium.
5. Involved in executing the Test script.
6. Involved in updating the test scripts.
7. Involved in maintaining the test scripts.

Scheduling.

System Study		Identify scenarios	Build day	
		Write the TC	↓	Execute the TC
		Review the TC		To review the defects
				1 st Release
		23 Jan 2019	30 Jan 2019	2 Feb 2019
		20 Jan 2019	25 Jan 2019	
		10 Jan 2019		
		1 Jan 2019		

Scheduling is a section which will give you the information, when to start and stop the testing activities of the project. With the help of scheduling we will get a clarity, when to start the testing activities. With the help of this, we will also get a clear picture which says when we will get the build and when we are supposed to release a build.

Note: The dates in the scheduling section are the rough estimation. It might change when we start the project.

Defect Tracking.

Whenever we get a build, as a TE we will start executing the build. While executing the build ie Test cases if we find any defects, we will communicate to the developer. i.e., we will log the defect to the dev, dev will fix the defect. As a TE, we will retest the fixed defect and will get a confirmation whether it is really fixed or not.

Procedure for defect tracking.

In this section we will get an idea, which says which type of DTR should we use.

Processor = i5 from intel

RAM = 4GB

Harddisk = 500GB

Software configuration

Server software configuration.

OS = Linux

Web server = Tomcat / Jetty / Putty.

App server = Web logics / Web sphere

Database =

Software configuration of client s/m.

OS = Win7, Win8, Win8.1, Win10

Browsers = IE, Google chrome, Opera

versions = IE10, IE11, Chrome 56, 57, 58, Firefox 60, 63, 71.

Entry and Exit criteria.

With the help of this section, we will come to know when to start and when to end, i.e. FT, IT, ST. This is the key element for test plan. We refer this, to make sure that when to start the testing activities.

Entry and Exit criteria for FT.

Entry criteria for FT - 1. Application should be done with WBT., ~~Functional~~ Functional scenarios and TC should be ready.

3. Resource should be ready.

4. Application should be installed in testing server

Exit criteria of FT.

1. Pass % should be 80%.

2. The count of blocker defect should be 0.

3. The count of critical defects can be 15.

4. The count of major defect can be 50.

5. The count of minor defects should be 100.

These counts are not standard. It will be varying from proj to project and comp to comp

Entry and exit criteria of IT.

entry criteria for IT.

1. It should satisfy the exit criteria of FT.
2. Integration scenarios and integration test case should be ready.

Exit criteria for IT.

1. It should satisfy the exit criteria of FT.
2. The pass % should be 90-95.
3. We should not have any blocker defects.
4. The count of critical defects should be 0.
5. The count of major and minor defects should be less than the exit criteria of FT.
6. The extent of

Entry and exit criteria of ST.

Entry criteria of ST.

1. It should satisfy the exit criteria of IT.
2. System Scenarios and system test case should be ready.

Exit criteria of ST

1. It should satisfy the exit criteria of FT, IT
2. The pass % should be 95-100.
3. We should not have any blocker defects.
4. The count of critical defects ~~and~~ should be 0.
5. The count of major and minor defects should be less than 15 and 20 respectively.

These standards are not stable, it might change once when we start working in the project, and these standards will be varying from company to company and project to project.

Test Automation.

With the help of this section we will get the information which is mentioned below.

1. When to start automation.
2. Which are the features to be automated.
3. Which are the features not to be automated.
4. Which automation tool we should use.
5. Which automation framework should we use, i.e.
 - 5.1. Keyword Driven.
 - 5.2. Data Driven
 - 5.3. Hybrid Framework.

6. How to select the features that has to be automated.

e.g. Automating help link is not required because it is not frequently used, by the user.

The module which is not stable, we can't automate it.

Note: The modules which are stable can be promoted for automation.

Deliverables.

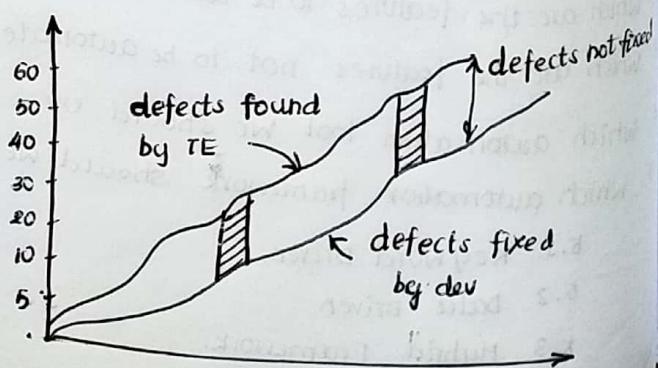
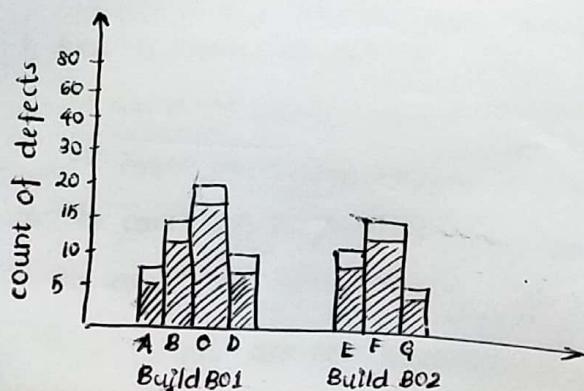
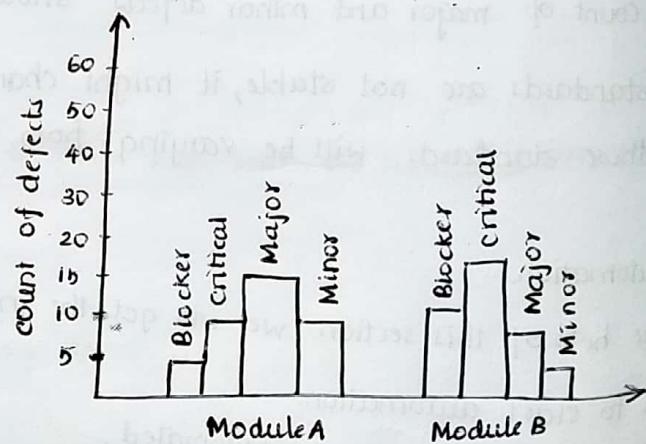
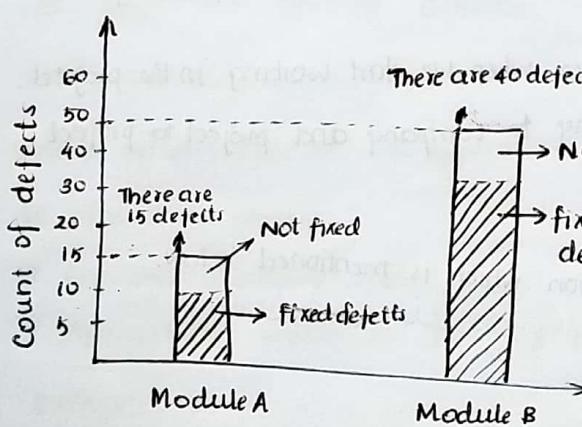
In this section we will have the information which talks about the output of the outcome of the testing team. This section will give a clear instructions which says, which are the documents to be submitted to the customer at the time of releasing the software.

1. Test Cases
2. Review Reports.
3. Execution reports.
4. Traceability matrix report.
5. Defect tracking report.
6. Test scripts.
7. Test plan.
8. Release notes.
9. Graph matrix

Graph

At the time of working in project or testing the project, TL will prepare some set of graph which talks about the defects.

fig. Defect distribution Graph



Metrics
These are the set of parameters which defines the testing quality on the product for every releases.

Test Eng name	Blocker		Critical		Major		Minor	
	fixed	Not fixed	fixed	Not fixed	fixed	Not fixed	fixed	Not fixed
Pradeep	10	5	15	5	30	10	50	10
shankar	15	10	20	10	40	20	70	20
shashi	20	5	20	10	50	10	30	10

Test eng name	Blocker		Critical		Major		Minor	
	found	fixed	found	fixed	found	fixed	found	fixed
Pradeep	10	5	15	5	30	10	50	10
shankar	15	10	20	10	40	20	70	20
shashi	20	5	20	10	50	10	30	10

Release Notes.

It is a document which is prepared by the TL / TM / BA, This is a document which will be given to customer at the time of releasing the software. The release notes will be once reviewed by the Test Manager, and will be signed by TM. In release notes, we will have the below mentioned data.

1. How many defects are fixed and not fixed.
2. Which are the newly added features.
3. Which are the removed feature
4. Which are the deleted features.
5. In which all platforms the application is tested and not tested.
6. The version of the software.
7. Steps to install the software.
8. Which are the open and pending defects.

P.T.O (after gmail IT TC)

? Alpha Testing is done on which site ? company site
 ? Beta Testing is done on which site ? customer / end users. site (public / private)

Templates

- In this section we will have some set of templates which will be used by TE at the time of prg
1. Test plan template
 2. Test case template
 3. Defect report template
 4. Traceability matrix template
 5. Test case execution template
 6. Review Template.

CBA- Test plan

Testplan History				
Authorname	Reviewername (TM/TL)	Approved by	Comments	Approved date
Ramesh	Suresh	Girish	Login Home Amt Transfr Amt Bal	1 Jan 2019
Loki	Venki	Pinki	Loans Insurance	10 Jan 2019

Page 2	
Contents	
1. Objective Page 3
2. Scope Page 5
.....
15. Templates Page 90

Page 3

Reference

- CRS → Link URL
- SRS → Link URL
- FS → Link URL

? Tomorrow is the release, Today you will find a Blocker defect; As a TE will you release the s/w to the customer?

As a TE I'm not supposed to decide whether to release the s/w or not.. Management team will decide whether to release the s/w or not. I will explain the blocker defect to the management team. The management team will communicate to client. But as a TE, I would like to give a quality s/w to customer, so I will support for not releasing the s/w.

Performance Testing [Bench mark Testing] [Bottle Neck Testing] / Base line / spike testing

Testing the stability and response time of an application by applying load is called as performance testing / Bench mark testing / Bottle neck testing.

Stability

The ability to withstand the designed no. of users is called as stability.

Eg: customer^{BA/Sr.Dev/Sr.TE} will tell the application should be used by ten thousand people, checking for 10,000 people and make sure the application is stable for 10,000 users.

Response time

The time taken to send the request to the server, run the request program in the server, receive the response from the server is called as response time.

T_1 = time taken to send the request to the server

T_2 = time taken to run the requested program in server.

T_3 = time taken to receive response from the server.

$$\text{Response time } R_T = T_1 + T_2 + T_3.$$

Load

No. of users using the application.

Eg: customer says the application should be used by 10,000 person, when 10,000 people are trying to login at a time, the home page should be displayed within 1 sec. Testing the above statement on the application is called as performance testing

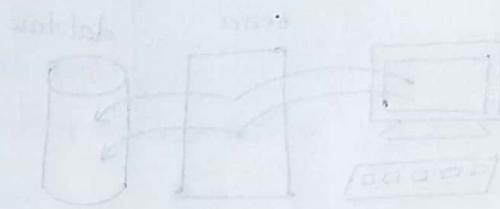
Types of performance Testing.

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

Load Testing

Testing the stability and response time of an application by applying the load which is less than or equal to the designed no.of users. is called as Load testing.

1,000	1 sec
3,000	1 sec
5,000	1 sec
7,000	1 sec
8,000	2 sec
9,000	2 sec
10,000	2 sec



The requirement says when 10,000 users are trying to login at a time, the home page should be displayed within 2 sec.

Stress Testing

Testing the stability and response time of an application by applying the load which is more than the designed no.of users.

$$\text{Req} = 10,000 - 2 \text{sec}$$

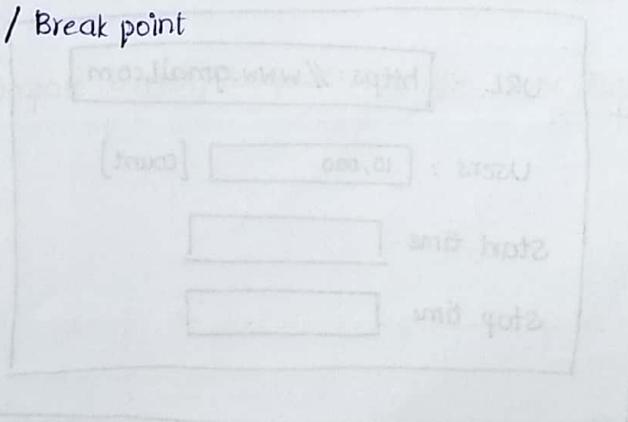
10,000	2 sec
12,000	2 sec
13,000	2.5 sec
15,000	3 sec
18,000	4 sec
20,000	5 sec

Scalability Testing

Testing the stability and response time of an application by applying the load which is more than the designed no.of users, here we will check in which stage the application will be crashed./ Break point

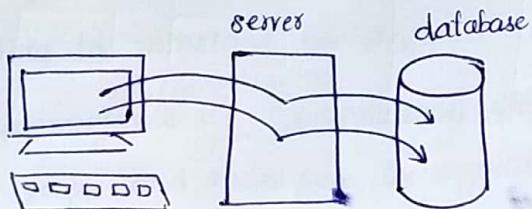
$$\text{Req} = 10,000 - 2 \text{sec}$$

20,000	6 sec
30,000	10 sec
50,000	1 min
70,000	3 min
90,000	10 min
1,00,000	0 404 - Blank page.



Volume Testing

Testing the stability and response time of an application by transferring huge volume of data is called as volume testing. It is mainly done to check the capacity of the database.



Soak Testing / Endurance Testing

Testing the stability and response time of an application by applying load continuously for a particular period of time is called as soak testing. eg: Login to facebook and not login out or not triggered out... / login to gmail and not login out. for a longer duration

Tools in performance testing.

1. Load Runner
2. J meter
3. NEO Load
4. Rational Performance test - IBM
5. silk performance testing tool
6. QA Load

Eg for performance Testing Tool.

A screenshot of a software interface for a performance testing tool. At the top, there is a toolbar with five buttons: 'Run', 'stop', 'start Recording', 'stop recording', and 'Pause'. Below the toolbar, there is a configuration panel. The 'URL' field contains 'https://www.gmail.com'. The 'Users' field is set to '10,000 [count]'. There are also fields for 'start time' and 'stop time', both of which are currently empty.

1. click on start recording
2. The recording will get started. [It will capture each and every action what we perform on the application].
3. Open the browser, enter the URL, login to an application.
4. Click on stop recording
5. The tool will automatically write the code for the action.
6. Enter URL in the tool and count of user and click on run button
7. After few durations click on stop button
8. Check the start time and stop time

What is the difference between smoke and soak testing.

Smoke

Testing the basic and critical features of an application before doing thorough or rigorous testing is called smoke testing.

• Smoke is a functional testing.

• We perform smoke testing as soon as we get a build.

Soak.

Testing the stability and response time of an application by applying load continuously for a particular period of time is called as soak testing / endurance testing.

- Soak is a non-functional testing.
- We perform soak testing once we are done with smoke, FT, IT, ST.

Will you write test cases for performance testing.

Here, we write test scenarios but scenarios itself will be referred as test cases.

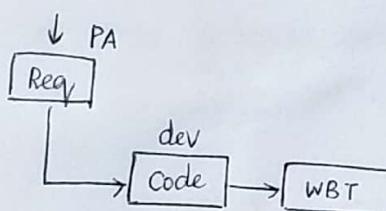
[high level test case].

Eg:- To check that when 10,000 users are trying to login, the home page should be displayed within 2 sec.

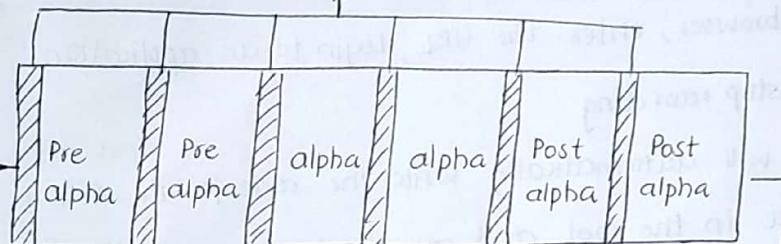
• To check that when user clicks on compose button, compose page should be displayed within 1 sec.

Alpha Beta and gamma testing.

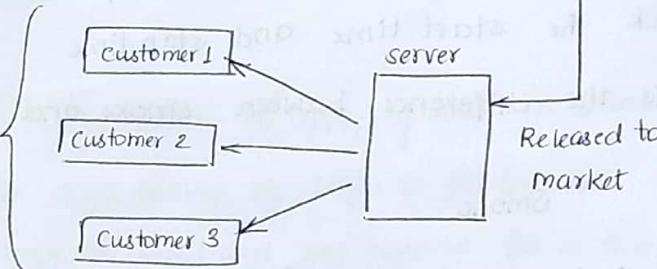
Customer



Smoke Testing



Beta
Testing



Alpha, Beta and gamma are the terminologies which are used by product based company

Alpha Testing

It is a testing which is done by the TE before releasing the software or product to the market.

In alpha we have 3 types.

- 1) Pre-alpha
- 2) Alpha
- 3) Post-alpha.

Pre-alpha

It is a testing which is done by TE on all the features at high level and will check whether the product is ready for next level of testing.

Alpha

Here the TE will do thorough and rigorous testing. where they will do both +ve and -ve testing.

Post alpha

In this stage the requirement will be freezed, the developers will be involved in fixing the defects and TE will be re-testing the defects.

- Note
1. Once the product is done with post alpha it will be released to the market.
2. We can perform prealpha, alpha and post alpha for n no. of cycles. [It will be guided by PA]
3. If we are in the stage of alpha, we are not supposed to go back to the pre-alpha stage.

Beta Testing.

This is a testing which is done by end users / cust / TE. Here the customers will download the free version or trial version of the product which was released by the company. If the customer faces any problem, he will send a feedback to the company, the company will cross check the feedback, develop it and test the feedback and again a new free version or trial version will be released to the market. The customer will again download the new free version, if he is comfortable he will purchase the product.

This is called as Beta testing.

Gamma Testing.

This is a testing which is done after Beta testing in the customer place at the time of releasing the software to customer. OR It is the third stage of software testing which is done after ~~B~~ testing before commercial release of the software to the customer.

? Why we should do Beta testing?

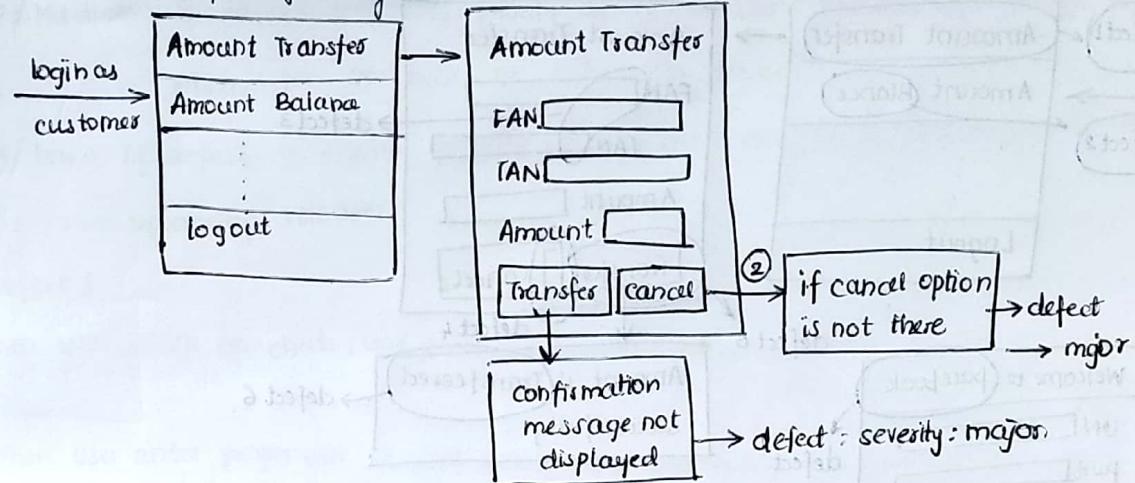
1. To check how the customer is using the application. we should do Beta testing
2. To check in which platform the customer is using the application we should do BT.
3. The application will be tested by the TE by their point of view. but the customer, he might use the application from his point of view. To know that scenario we will do Beta testing.

? What is the difference between Beta and acceptance Testing

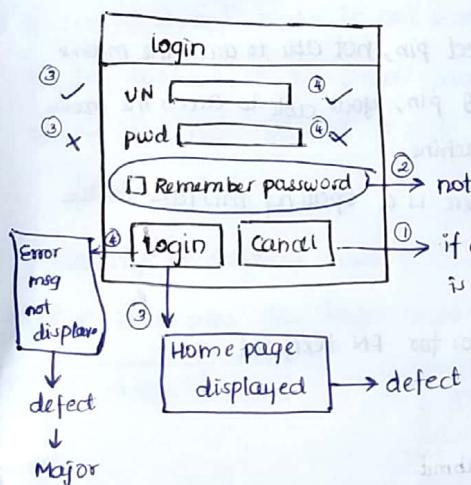
Beta Testing	Acceptance Testing
<ol style="list-style-type: none">1. It is a testing which is followed by product based companies.2. Company will take the ownership of product3. The company will not be having control on the product ie, we will not come to know for how many users we have to develop the application, and on which platform we have developed the application.	<ol style="list-style-type: none">1. It is a testing which is followed by service based companies.2. Customer will take ownership of product3. Customer will be having control on the product where, he will have an idea for how many users and on which platform we have to develop the application.

Continuation of Major - egs.

Eg:

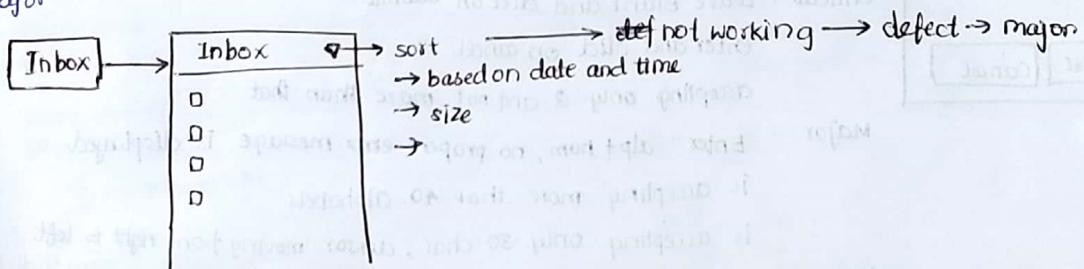


When user recharges a mobile, mobile is recharged successfully, balance is displayed properly, but message not send to the user. → defect → major.



In WhatsApp, user is not able to send 1 of the smiley, defect → major.

- ① able to send 1 of the smiley, defect → major.
- ② not able to send any smiley → defect → blocker
- ③ 'not mate' is not working → defect → major
- ④ able to start the msg → defect → major.

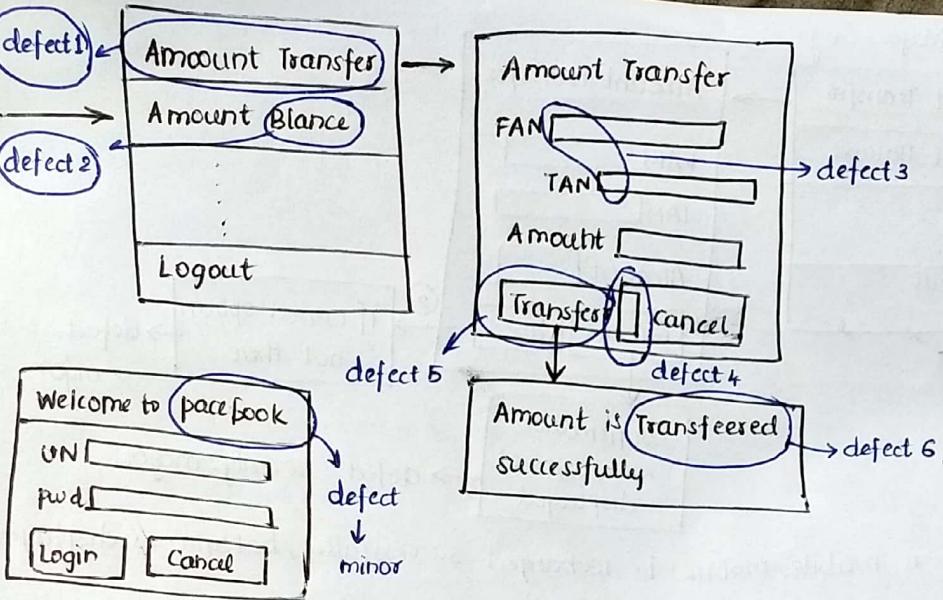


Minor

Assume that there is defect in the application and you are 100% sure that this defect will never affect the customer business. These kind of defects are called minor defects.

like spelling mistake, alignment problem, object overlapping, all the cosmetic defect / usability defects.





Identify blocker, critical, major, minor defects (4) for ATM machine

blocker - Take ATM card, insert to ATM machine, enter correct pin, not able to access the machine.

Critical - Take ATM card, insert to ATM machine, enter wrong pin, you're able to access the machine.

Major - User is not able to get the receipt in ATM machine.

Minor - In ATM machine receipt is generated but there is a spelling mistake in the receipt.

should accept
b/w 2-40 char

gmail

* FN [text input field]
Submit Cancel

Tell the blocker, Critical Major, minor for FN textfield

blocker : FN text field is disabled.

Critical : Leave Blank and click on submit.
Enter and click on cancel
accepting only 3 and not more than that

Major : Enter alp+num, no proper error message is displayed
is accepting more than 40 alphabets.
is accepting only 30 char., cursor moving from right to left

Minor : Is not showing place holder
showing text area field
First Name → FN.
* in blue colour.

★ Priority

Importance given to fix the defect, is called as priority/ how soon the defect needs to be fixed by the developer is called as priority. There are 3 types of priority.

1. P1 / High: If defect is having priority as P1, then dev should fix the defect immediately.

P2: / Medium: If defect is having priority as P2 / medium, then dev can fix the defect within few test cycles or within the release.

P3 / low: If defect is having priority as P3, then dev can fix the defect in next upcoming release.

Defect 1

When user click on chats, 404 error displayed. Defect : severity : Blocker priority: P1.

2) When user enter proper user UN and wrong pwd into gmail, home page is displayed. Defect : severity : critical . priority : P1.

3) In whatsapp user is not able to delete the chat. [critical, P1]

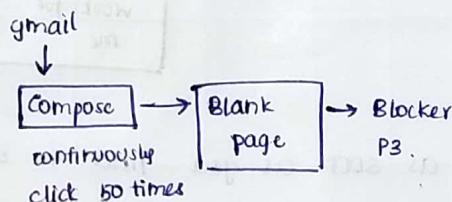
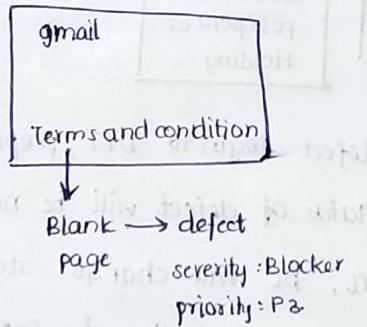
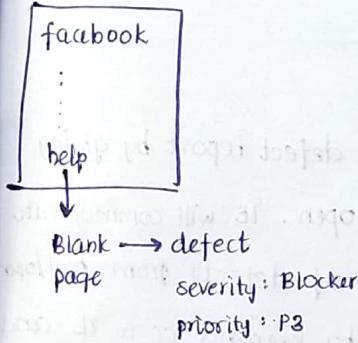
4) In whatsapp mute is not working [Major, P2]

5) In flipkart, in the orders page at the bottom of the page [Minor, P3]

6) In ola, user booked a cab, for some location , later it is showing some other location [critical, P1]

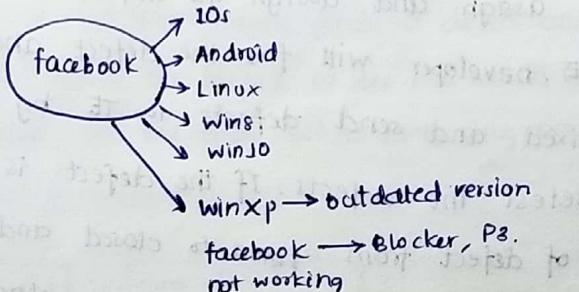
Identify 15 defects and give proper severity and priority.

* Give examples for high severity and low priority. [Blocker / critical , P3].

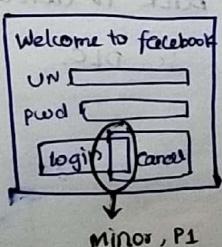
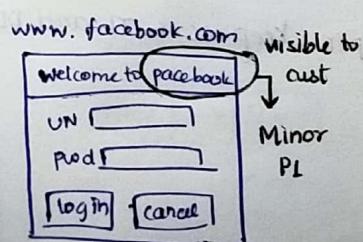


Whatsapp
↓
50 times install and uninstall and pg is going to blank → Blocker, P3

Whatsapp
↓
upload 100 photos & videos is uploaded
↓
blank → blocker, P3



Give examples for low severity and high priority. [minor , P1]



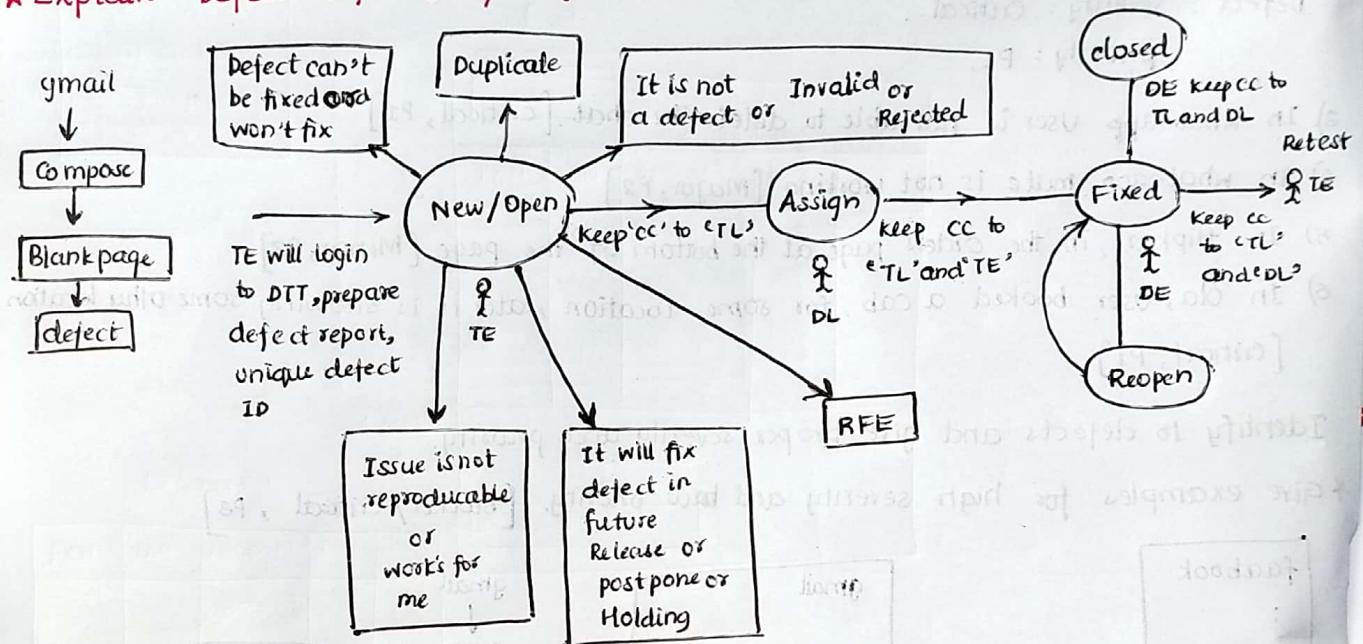
Get me eg for high severity and low priority (1), low severity and high priority (1) high severity and high priority (1), low severity and low priority (1)

Who will give severity and priority.

TE will give severity and priority, but priority can be changed / given by developer, BA, customer, PM.

Developer will always fix the defect which is having highest priority rather than high severity.

* Explain Defect Life Cycle / Bug life cycle.



TE as soon as you find the defect login to DTT, prepare defect report by giving unique defect ID, By default status of defect will be new/open. TE will communicate defect to DL by keeping CC to TL. DL will change status of defects from new/open to assign and assign the defect to respective developer by keeping CC to TL and TE. developer will fix the defect and change status of defect from assign to fixed and send defects to TE by keeping CC to TL and DL. TE will retest the defects. If the defect is really fixed, then TE will change status of defect from fixed to closed and send a mail to DE by keeping CC to TL and DL stating defect is closed.

Suppose if defect is not fixed then TE will change status of defect from fix to reopen and send the defect back to developer keep CC to TL and DL stating defect is reopened. This process is DLC.

Defect Life Cycle

1. New / Open
2. Assign
3. Fixed
4. Closed
5. Reopen
6. It is not defect
7. Duplicate
8. Defect can't be fixed.
9. Issue is not reproducible
10. Fix the defect in future release
11. RFE

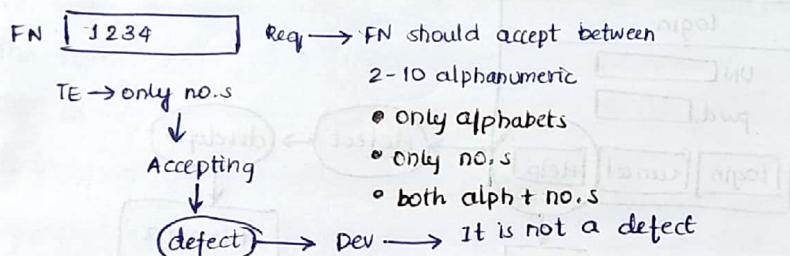
What is the difference between New / Open?

TE when you find defect for the first time, then we give the status as new.

When developer starts to work on the defect, then status should be given as open.

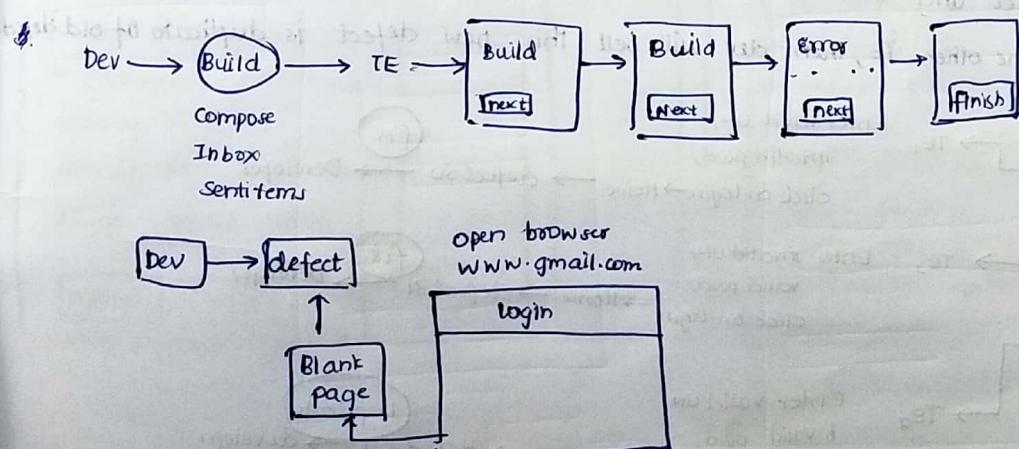
* Why developer will tell, it is not a defect or invalid or rejected?

1. Because of misunderstanding of the requirement



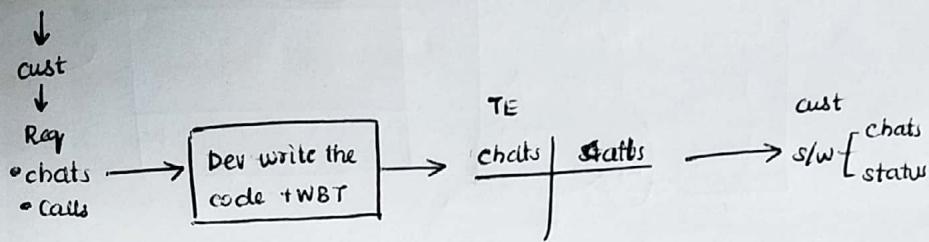
2. Because of build / software is wrongly installed

TE will install the build wrongly and start to test the software, finds a defect and communicate to developer. Now, developer will reject the defect or dev will tell it is not a defect because, there is no problem in the code, where in testing team has wrongly installed the software.



3. Because of referring old requirement

Release 1 (1 year)



Release 2

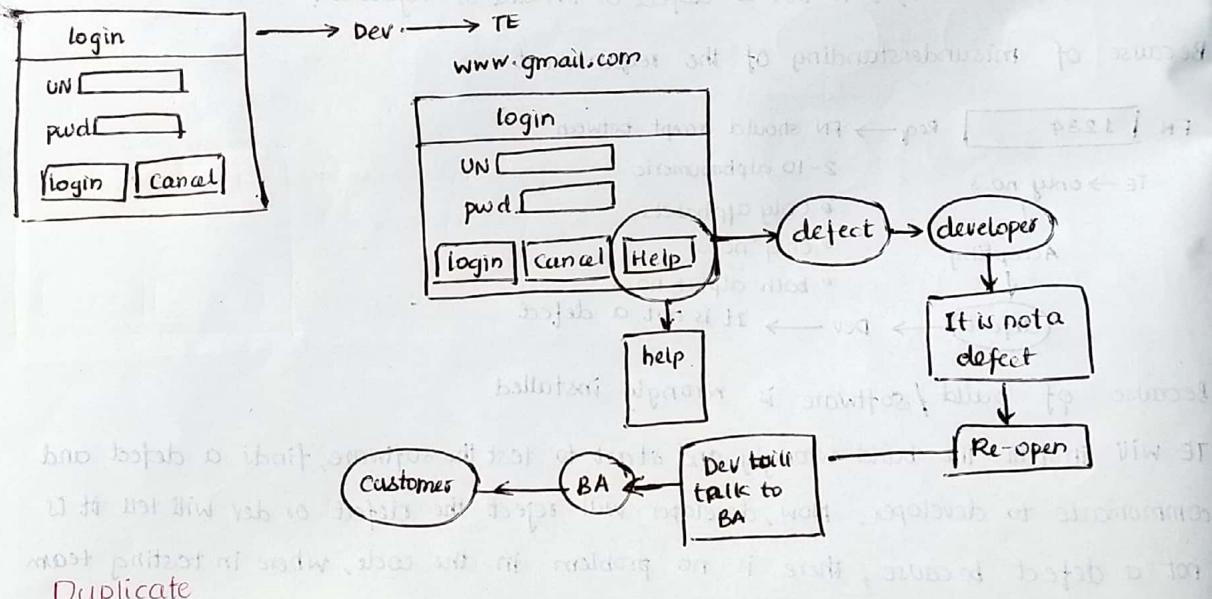
Requirement



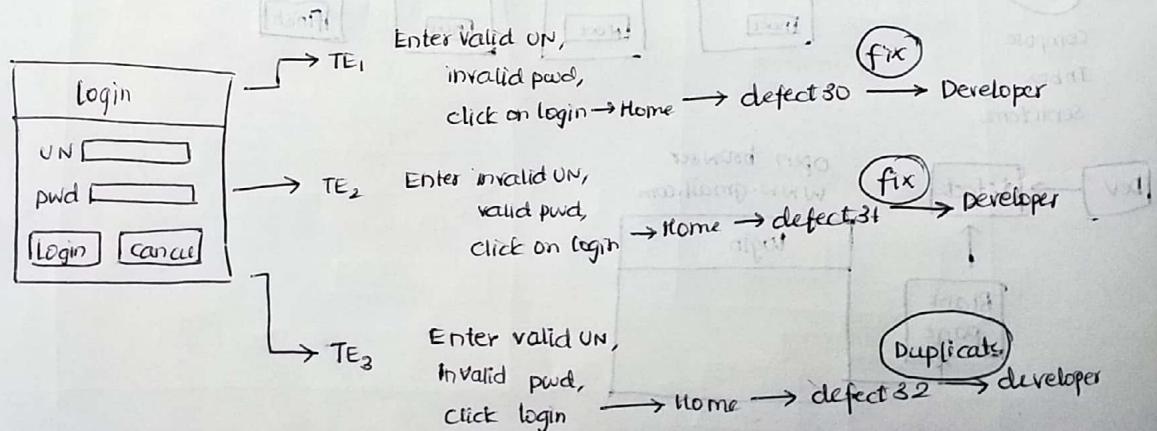
4. When developer adds extra feature.

Cust

Requirement



TE find the defect and send it to developer and here the same defect is already tracked by some other TE, then dev will tell this new defect is duplicate of old defect



Why developer will tell it is a duplicate?

To reduce defect count.

To avoid the duplicate effort.

Why TE find duplicate defects?

1. Because of testing common feature.

2. Assume that 1 of the TE has worked for 3 yrs and tested the project and found around 50 defects and quits the job, New TE joins the company and test the same project and find the same defect to developer, now developer will tell these defects are duplicate of old defects.

If TE finds a new defect and already same defect is present with status as new, then we should not track the defect.

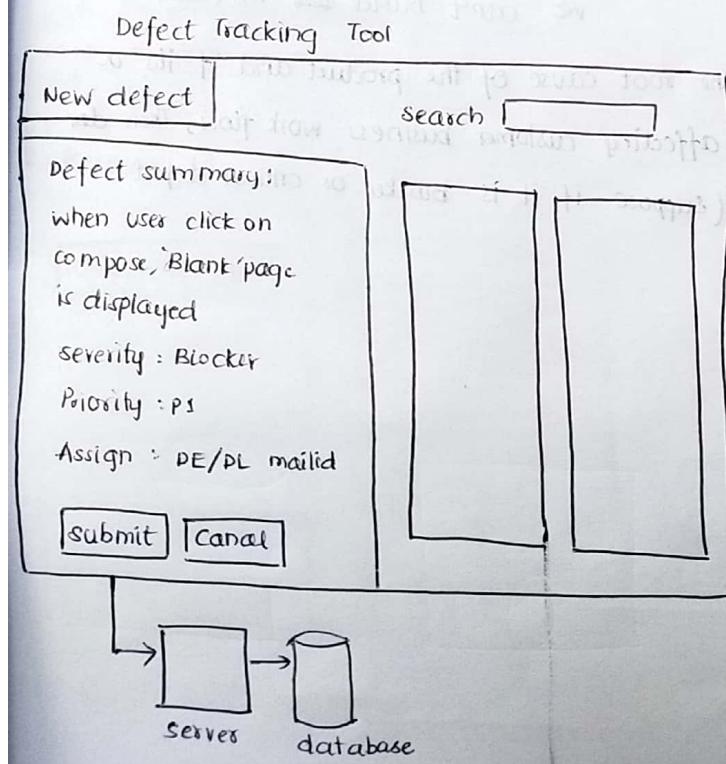
If TE finds a defect and already same defect is there with the status as fixed, then we should re-open the defect.

If TE finds a defect and already same defect is present with status as closed, then we can track the defect.

How to avoid duplicate?

1. TE when he finds defects and communicate to DL, keep cc for TL and TE working on the same project and same team.

2. TE as soon as you find defect, login to defect tracking tool, search for duplicate defect, by using defect id or keywords with wild card, it will display matching defects, if already same defect is present, then don't track the defect, if same defect is not present and this is how we can avoid duplicate.



TE → defect → login to DTT → search

too, Search

compose, Blank *

It will display list of all
or defect ID
matching details

Click on compose, 'Blank' Page	Blocker	P1	New

3. TE When you find defect, before you track it, crosscheck with team members, and sometime developers, so that we can avoid duplicate defects.

Defect can't be fixed or Won't fix.

1. If technology itself is not supporting

It means the programming language which is used to develop the software is not having solution or capacity to fix the problem.

Requirement
Transaction displayed in excel

Transactions				
FAN	TAN	Amount	Date	Time

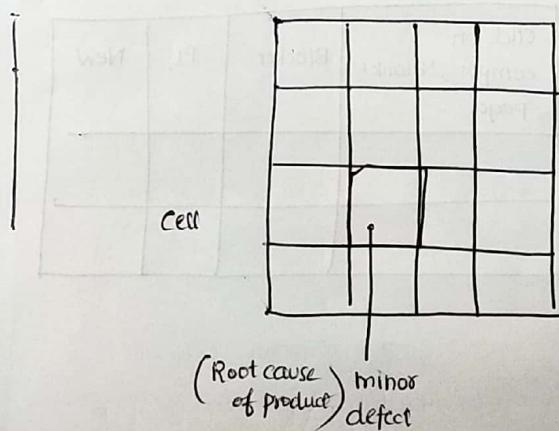
FAN	TAN	Amount	Date	Time

1 lakh

search
Download to Excel

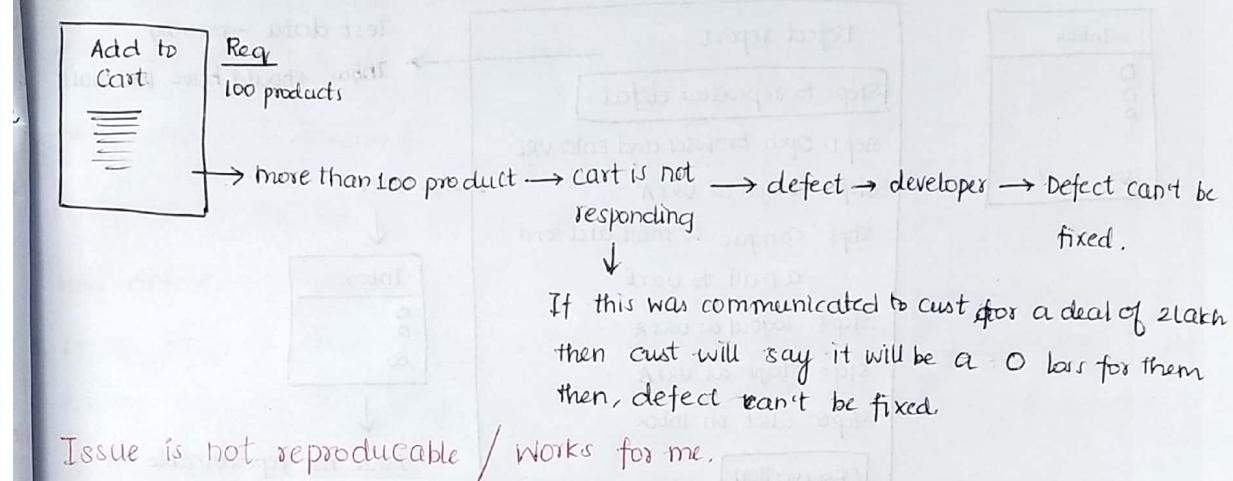
64,000 Transaction → defect → developer → can't be fixed because of Technology problem and by using this Java code, we can't build ~~MS~~ MS excel.

2. When TE finds the defect in the root cause of the product and if its a minor defects and if it is not affecting customer business work flow, then dev will tell defect can't be fixed. (suppose if it is blocker or critical defect developer should fix the defect.)



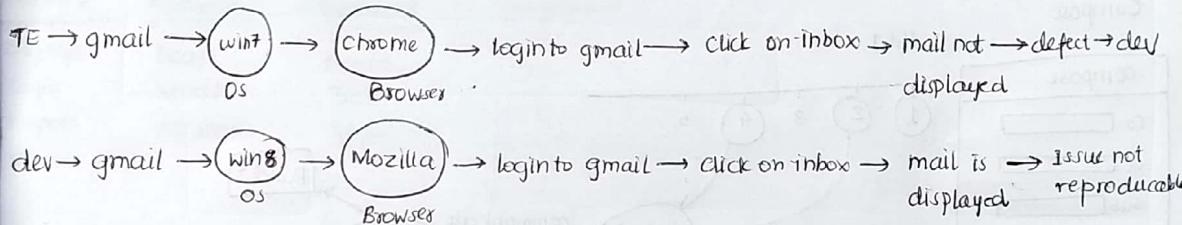
3. When cost of fixing defect is more than the cost of defect, then dev will tell defect can't be fixed.

cost of defect : loss occurred to the customer because of defect



? What do you mean by issue not reproducible / works for me?

TE finds the defect and send it to developer and developer is not able to see the defect and everything works fine for developer. In this case, dev will tell issue not reproducible.

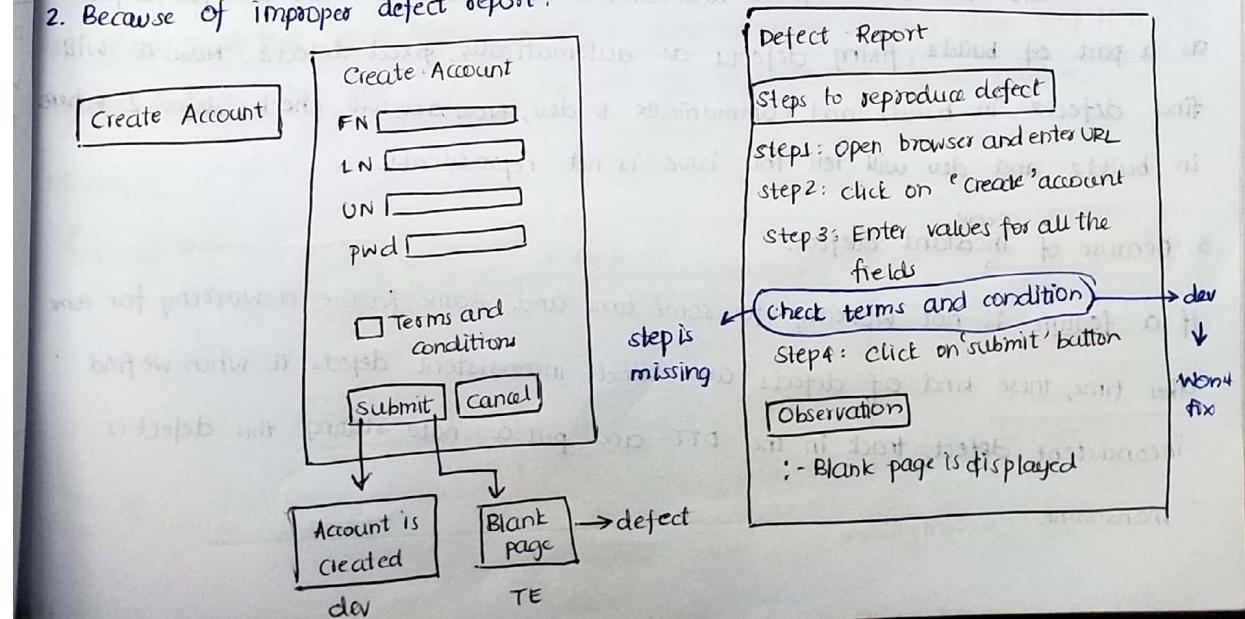


? Why TE finds not reproducible defects?

1. Because of platform mismatch

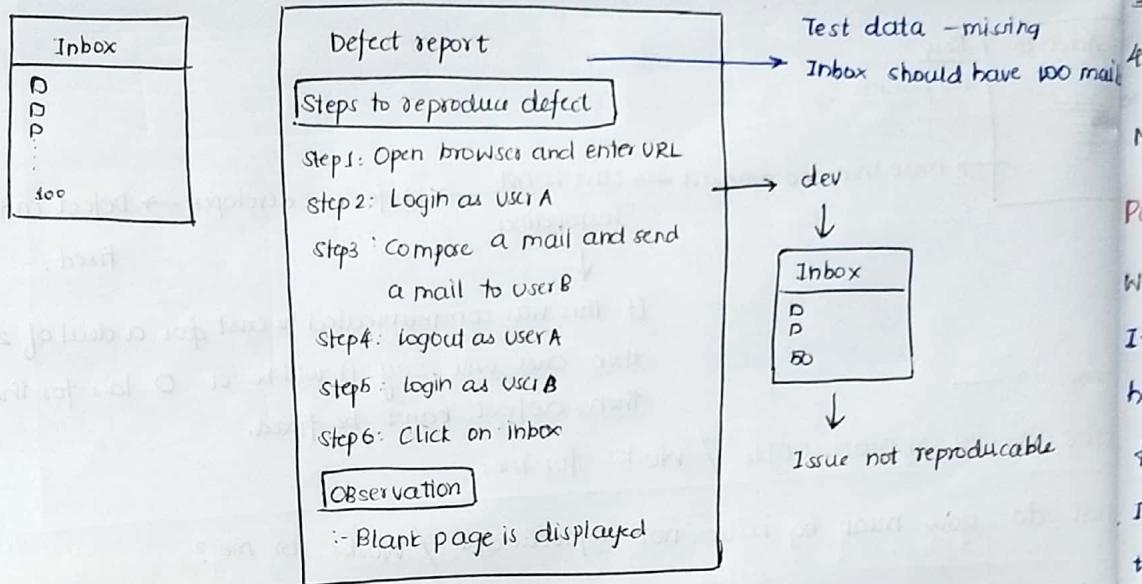
- a. OS mismatch
- b. Browser mismatch
- c. version mismatch
- d. Browser settings mismatch

2. Because of improper defect report



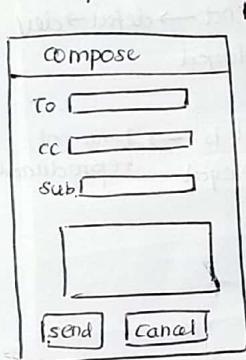
3. Because of data mismatch

gmail → Login as userA, compose a mail and send a mail and send to userB, logout as userA, login as userB, click on inbox

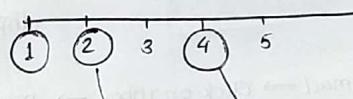


4. Because of Build mismatch.

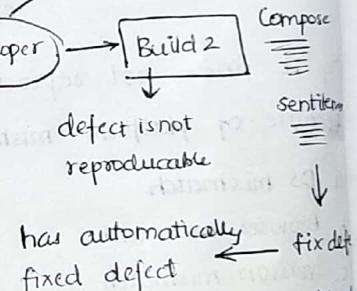
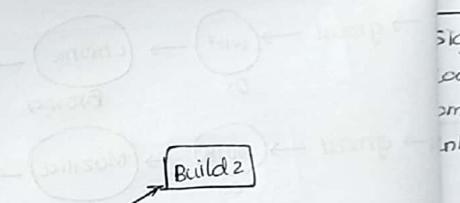
Compose



Build 1



communicate



Developer will give build1, and build1 consist of 2 defect (defect1 and 2), TE will st to test build1 and find defect1 and communicate to dev and dev will fix, defect as a part of build2, fixing defect1 as automatically, fixed defect2. Now, TE will find defect2 in build1 and communicate to dev, Now dev will check, defect2 in build2 and dev will tell this issue is not reproducible.

5. Because of inconsistent defect.

If a feature is not working for some time and same feature is working for other time, these kind of defects are called inconsistent defects. TE when we find inconsistent defect, track in the DTT and put a note stating this defect is inconsistent.

Flipkart

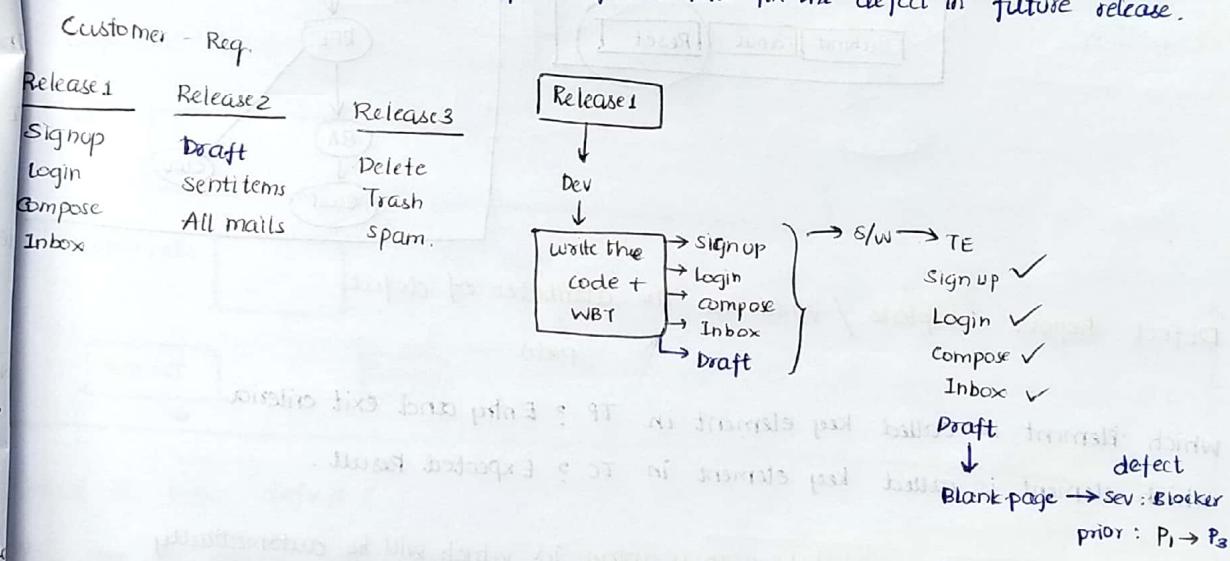
- 1 → Add product to cart → check, in case product not reproducible → defect
- 2 → Add product to cart → product is displayed
- 3 → Add product to cart → product is not displayed
- 4 → Add product to cart → product is displayed.

Note: defect is inconsistent.

Postpone or fix the defect in future Release or Holding.

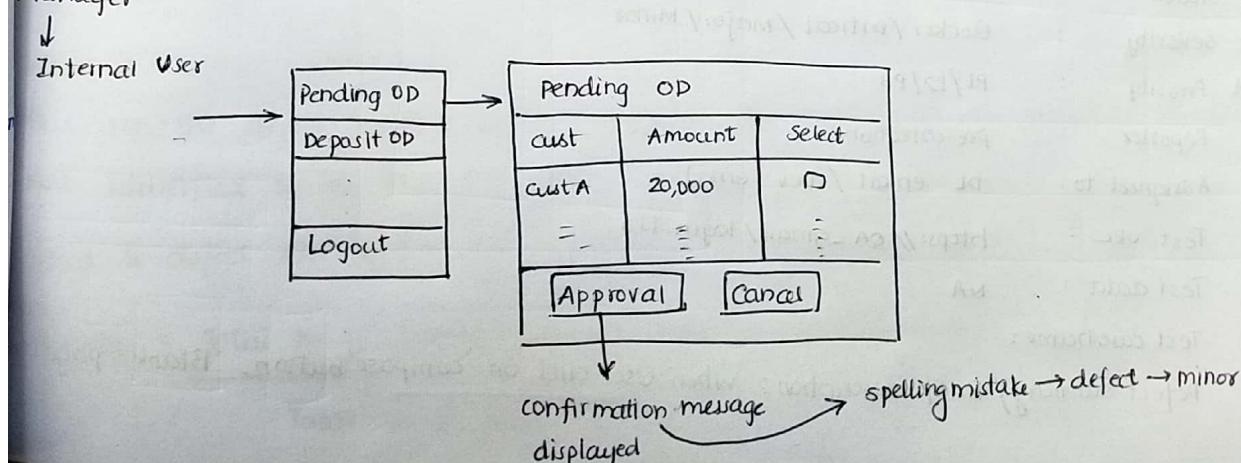
Why defect gets postponed?

1. If TE finds a minor defect at the end of the release and the developer is not having sufficient time to fix the defect, then dev will tell he will fix the defect in future release.
2. If TE finds the defect in a feature which is not required for the customer in the current release then, developer says he will fix the defect in future release.



TE finds a defect in a feature which is exposed to the internal users and if it is a minor defect, then dev will tell, he will fix the defect in future release.

Manager



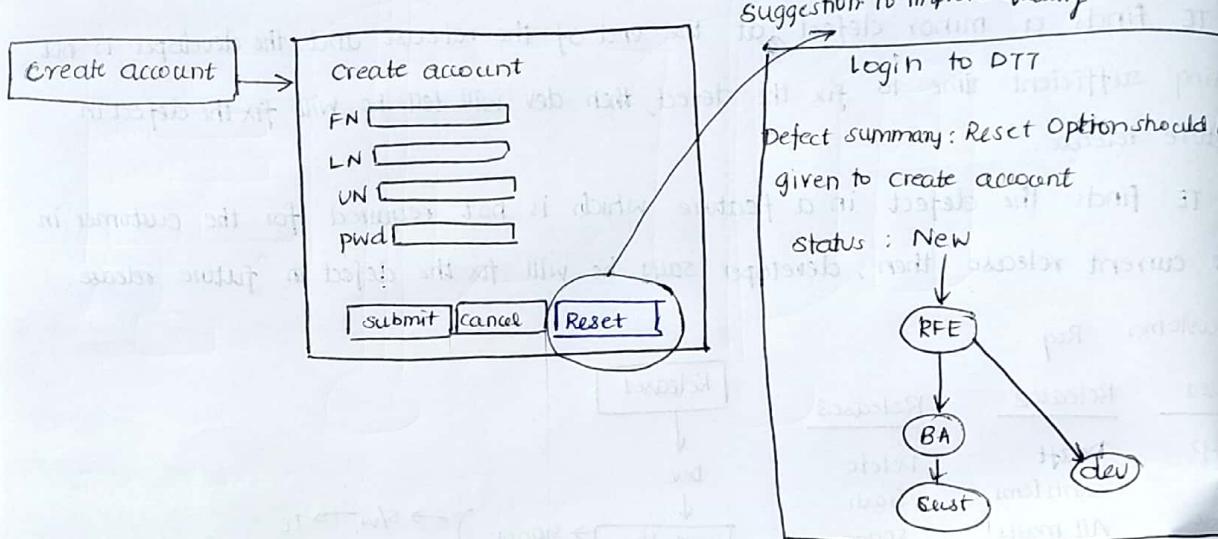
4. When TE finds a defect in a feature, where in cust is planning to do lot of requirement changes in the same feature and in customer can remove the feature as well. In this case, dev will tell, he will fix the defect in future release.

RFE [Request for enhancement]

TE finds a defect in the software and if this defect is not part of the requirement, then, dev will tell it is an RFE.

Who will give RFE?

RFE can be given by TE, dev, BA, PM and customers



Defect Report Template / What are the attributes of defects
(fields)

Which element is called key element in TP? Entry and exit criteria

Which element is called key element in TC? Expected Result.

* Defect id : 4741 → defect ID is unique id which will be automatically generated by 'DTT'.

Project name : Gmail

Release name : TIGER

Requirement no : 31.4.1

Module name : Compose

Build ID / NO : Build B033

* Status : New/Open, Assign, Fixed, Closed

* Severity : Blocker / Critical / Major / Minor

* Priority : P1 / P2 / P3

Reporter : pre-condition

Assigned to : DL email / Dev email

Test URL : <https://qa-gmail/login.tld>

Test data : NA

Test case name :

Defect summary / Brief description : When user click on 'compose' button, 'Blank' page

Detailed Description.

OS : Win 10

Browser : IE

Browser version : IE9

Steps to reproduce defect

Step 1 : Open browser and enter URL

Step 2 : Enter 'UN' and 'PWD' and Click on 'Login' button.

Step 3 : Click on 'compose' button

Observation

'Blank' page should be displayed.

ATTACH Screenshot

It is a report to show for developer that defect is really present in application

Expected Result :

compose page should be displayed

cc

Comments

Submit

Cancel

What is Fatal defect?

This are the defect which results in complete software failure and TE can't continue the defect.

What is deferred / Trivial defect?

This are the defect which will not affect cust business workflow and these defects can be fixed in the upcoming release. These defects / minor defects is called deferred or trivial defect.

What is Latent defect?

This are the defect present in the software for some particular period of time and not identified by the TE.

What is defect density?

Defect

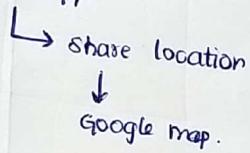
Density = Total no. of defects found - Total no. of defects fixed

Total no. of ~~executed~~ Test case executed

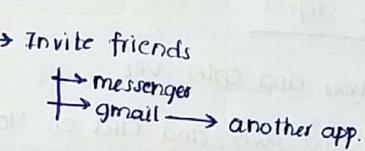
What is system Integration Testing?

Testing data flow between 2 different software and also make sure that end-to-end flow between 2 different s/w is working as expected or not. is called S/m Integration Testing.

eg1: WhatsApp



eg2: WhatsApp



★ Why does software have defect?

1. Programming errors.
2. Build not installed properly
3. Misunderstanding / miscommunication of requirement
4. Software complexity.
5. Because of CR
6. Time pressure

When to stop testing?

1. When the product is ~~not~~ functionally stable
2. When the basic functionality itself is not working [Blocker]
3. When time span is less, we should test basic and critical features and stop
4. When there is no budget.

Why should we do testing?

1. To ensure that quality in software.
2. To ensure that all the customer requirements are implemented correctly.

What are the principles of s/w testing or Manual Testing?

1. We should do early testing
2. We should not do exhaustive testing.
3. Testing is done to show that presence of defect in software.
4. We should follow pesticide paradox.
5. We should follow defect clustering.
6. Testing is context dependent. [Depending upon the type of the application and how customer uses the software].
7. Absence of error in the software doesn't mean that software is defect free.

What are the types of defects?

1. Functional defect
2. Usability defect.
3. performance defect
4. compatibility defect.
5. Database defect.
6. Web-security defect
7. Globalisation defect.

• Wrong implementation defect

• Extra implementation defect

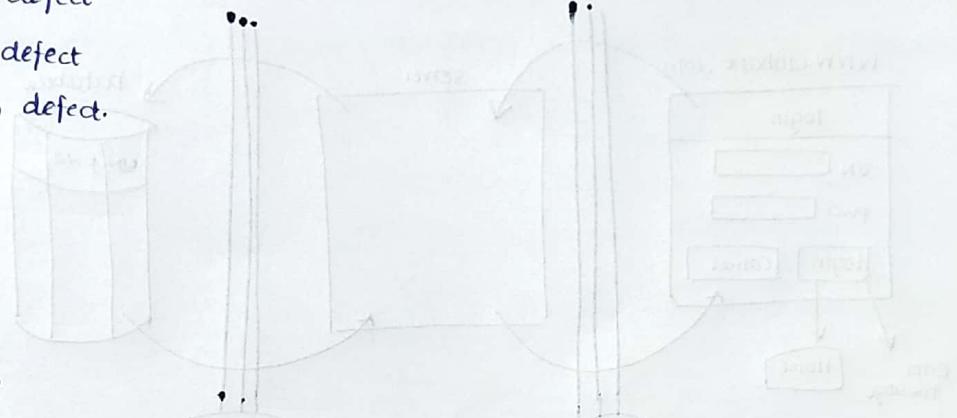
• Missing implementation defect.

✓ Blocker defect

✓ critical defect

✓ Major defect

✓ Minor defect.



What is Testware?

Collection of all the materials that is required to test the software is called Testware.

e.g.: Test case, Test scenario, TCM, TScript etc.

What is compliance Testing?

Testing the software and check whether software needs government rules, regulations policy is called compliance testing.

What is Risk based testing?

Testing the project which is under risk is called Risk-based Testing. Here TE will try to test the feature or functionality which has got highest impact and probability of failure.

What is Pilot Testing?

It is a type of testing done by set of users where in they will do trial run of the project, check the performance, give the feedback to the company before software is deployed into production server. This is Pilot testing.

(For B testing, once after sw is deployed into P.s only we will do).

What is the difference between Test strategy, Test plan and Test approach)?

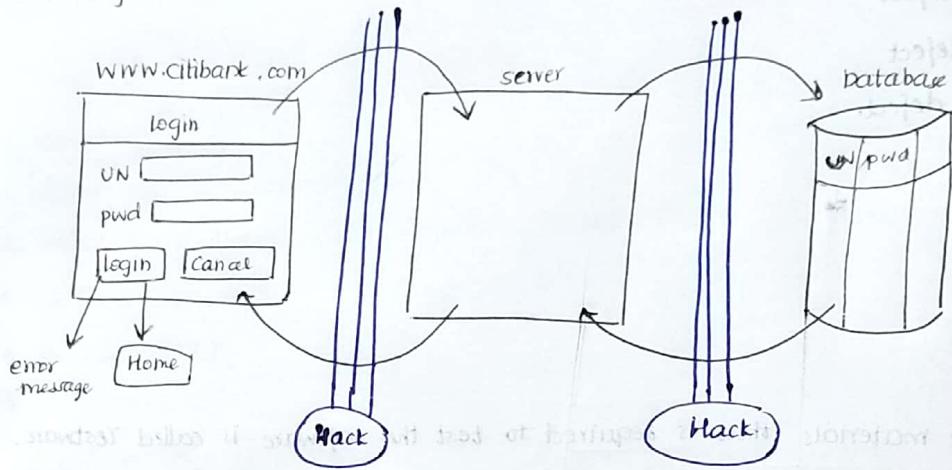
Test 'strategy' is also called as Master test plan.

Test approach is a part of Test plan.

Test strategy is a document which is prepared by the project manager on organisation level. Test strategy is a document which captures the organisation's view on how we go about testing all the projects in future.

What is Web-security testing? OR security testing or penetration testing, Testing the authentication of an application to check, how well the application is secured from unauthorised users or hackers is called web-security testing.

Testing the Authentication of an application.



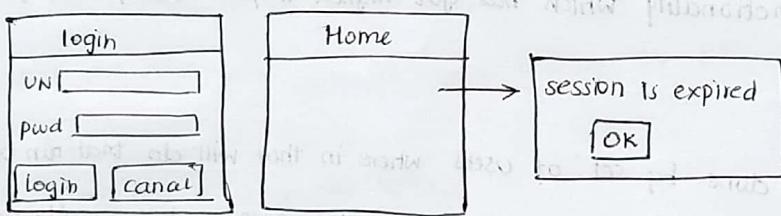
Types of Web security.

1. URL manipulation

Open browser, enter URL, click on Login, copy, paste on notepad and logout, and enter tab and copy, paste the same URL, it should ask for UN and pwd.

2. session Expiry

www.citibank.com



3. SQL injection.

It is a technique used by hackers where in they try to inject SQL commands into SQL statements through webpages.

4. Cookies based testing.

Cookies are some piece of information send from webserver to webbrowser. There are two types of cookies, i.e., session cookie [These are the cookies which are stored in memory of the browser and these cookies will be automatically deleted as soon as the session ends], and persistent cookie [These are the cookies which are stored in memory of the browser and these cookies will be deleted only when explicitly deleted].

stored in memory of the computer. These get deleted once the cookie reaches the expiry date. Or user should manually delete the cookie.

5. CSRF ATTACK [cross site request for Forgery].

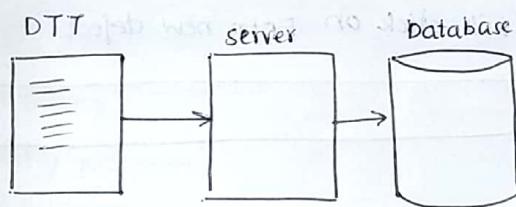
6. XSS ATTACK [cross site scripting].

Tools for doing web security

Burp Suite

What is defect tracking tool and explain how to track the defect in DTT.

DTT is a software which is used to store the defect or track the defect in centralised place and communicate defect to developer in an organised way.



- 1. Bugzilla - Free Tool /Open source
- 2. QC/AIM } Write TC
- 3. JIRA } Track defect
- 4. MANTIZ } store requirements
- 5. Bugzini } PM Tool
- 6. Bugpro } It is licensed
- 7. Bugnet
- 8. Rational Clear Quest.

How to track defect in DTT. [Bugzilla/QC].

TL/TM/IT eng will give URL, UN and pwd to TE.

Step 1: Open browser and enter URL
<https://server-Bugzilla/login.jsp>.

Welcome to Bugzilla	
UN	<input type="text"/>
pwd	<input type="password"/>
<input type="button" value="login"/>	<input type="button" value="cancel"/>

Home

Enter new defect Search
create issue defect id

Dashboard

projects
→ CRM list
→ ERP of
→ fb defects
→ gmail

Configure fields
 defect summary
 status
 desc
 priority
 module name
 defect id
 Query bugs

Step 2: TE will first search for duplicate defects by using search option (if you know defect id). If you don't know defect id, then we have to go for advance search option.

Advance search

Project name	gmail
Release name	Tiger
Module name	compose
Build no	B033
Status	New
Severity	Blocker
Priority	PL
Reporter	
Bug summary	Compose, Blank *
<input type="button" value="submit"/> <input type="button" value="cancel"/>	

List of matching defects

Defect id	Status	Severity	Bug summary
4291	New	Blocker	'Compose' page
4279	New	Blocker	'Compose' page
7160	New	Blocker	'Compose' page

steps: Suppose if there are no duplicates, then TE will click on Enter new defed.

Enter defect details.

Project name	
Release name	
Requirement no	
Module name	
Build id	
Status	
Severity	
Priority	
Reporter	
Assign to	
Test URL	
Test Data	
Testcase name	
Defect summary	

Detailed description

OS :

Browser :

Browser version :

Steps to reproduce defect

Observation

Attach Screenshot

ER

CC

submit

cancel

Reset

Bug history

Defect id field will be displayed with unique defect id