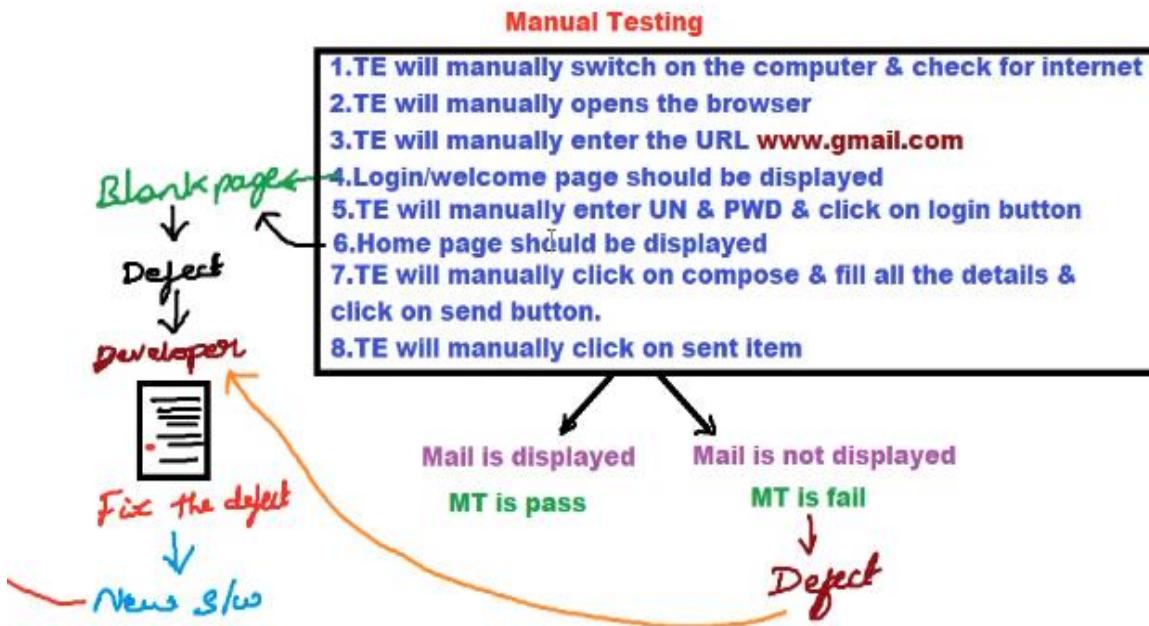
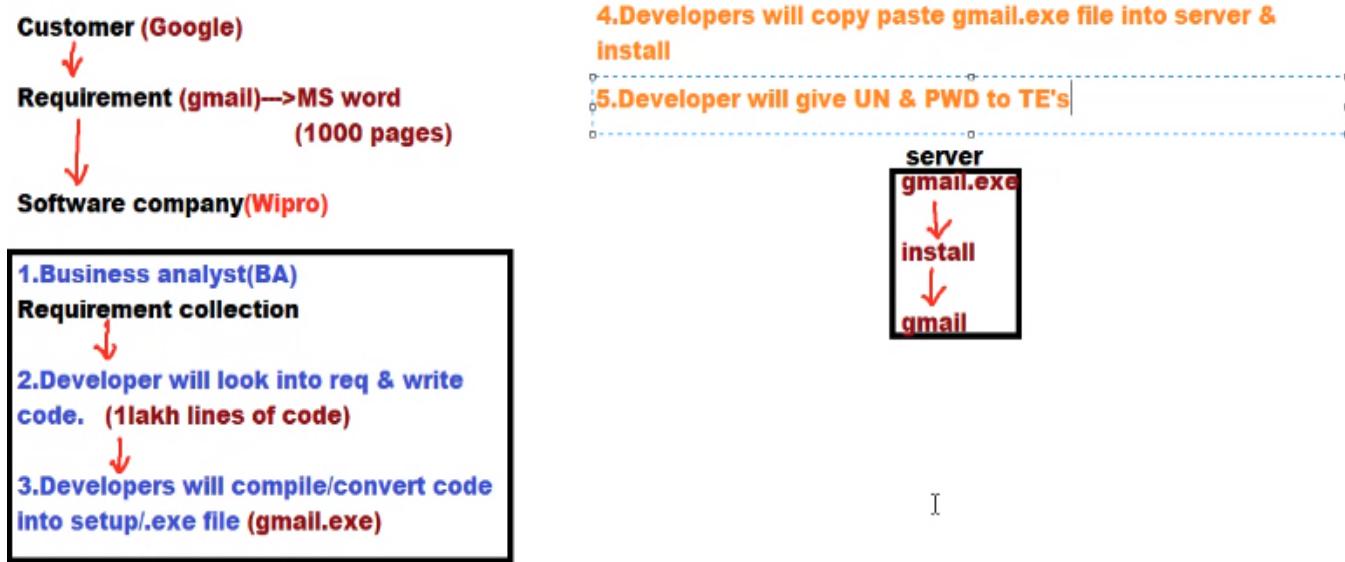


MANUAL TESTING NOTES



Note: What is the difference between Dev & TE?

TE will find the defect & communicate to developer. Developer will go to the code & fix the defect & give new s/w to TE.
TE/Developer will uninstall old s/w & install new s/w & TE will test for the defect manually.

Drawbacks of Manual Testing:

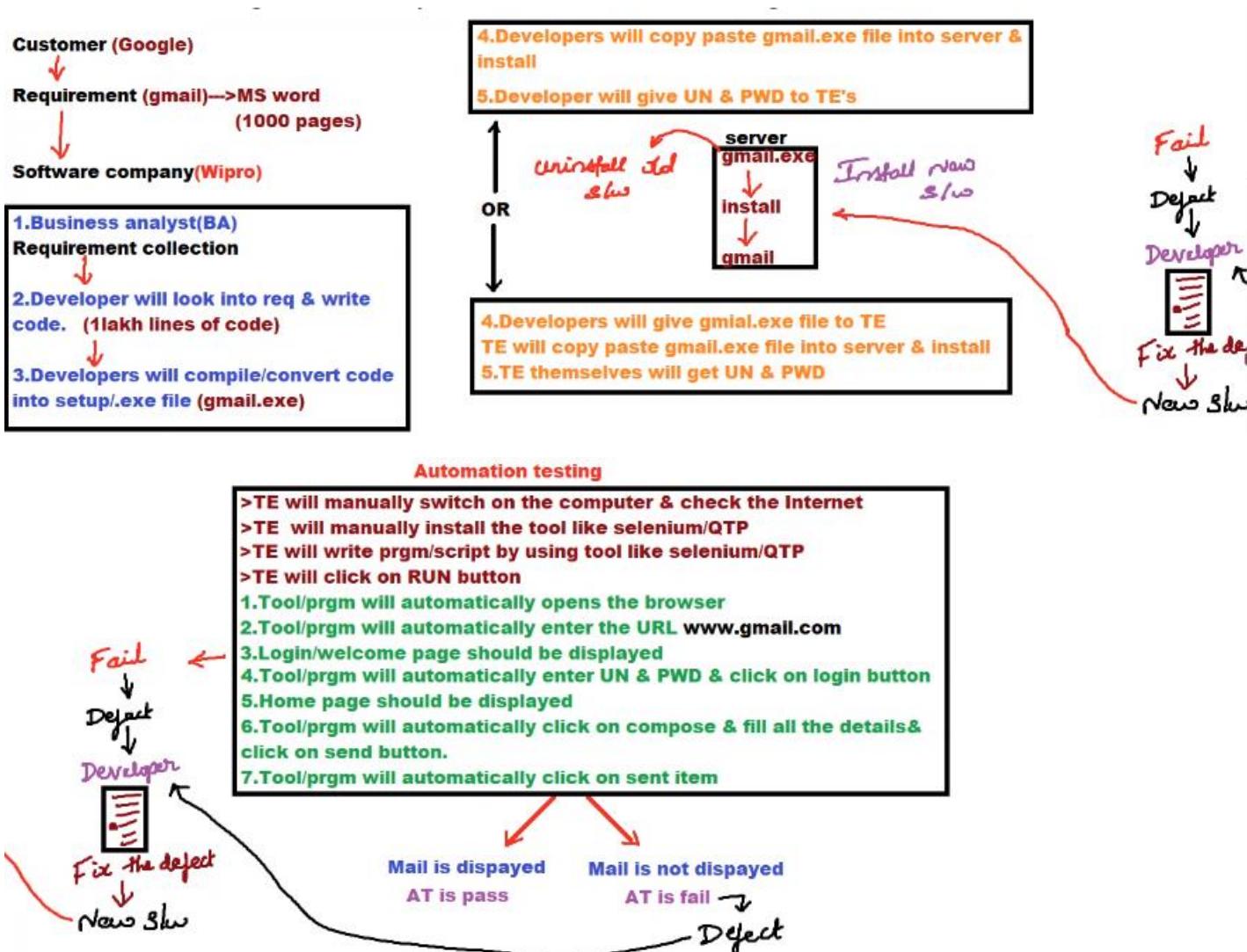
1. Time takes is more.
2. Resource utilisation is more.
3. It is a tedious and monotonous job.
4. There will be no consistency in testing.

Advantages of Manual Testing:

1. Quality will be good.
2. Programming knowledge is not required.

Note: In order to overcome the drawbacks of manual testing, company will go for Automation.

Every company when they get new project, first they will do manual testing to make sure that the quality will be good, then only they will go for Automation.

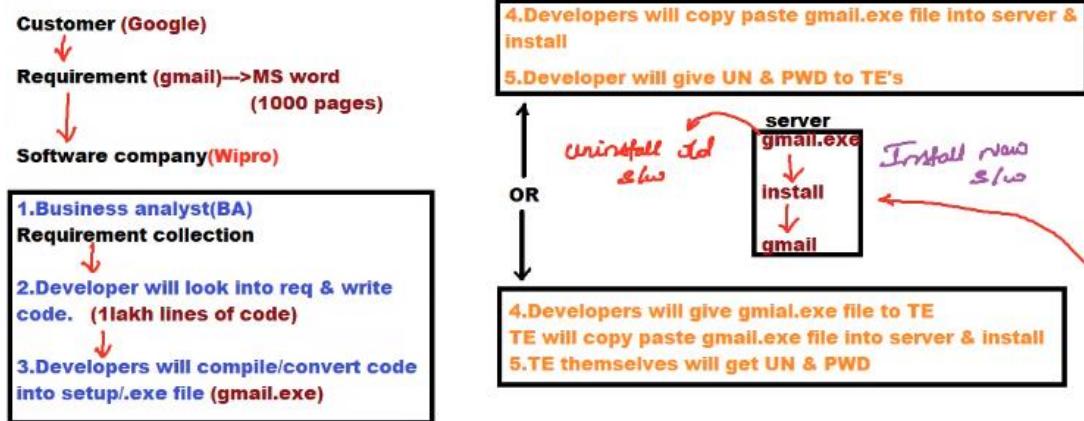


Automation Testing:

Test Engineer will write program/script by using tools like Selenium/QTP and run the program against the application or software. Tool or program will automatically test the application and gives result as pass/fail this is called as Automation Testing.

Manual Testing:

Testing the software or Application repeatedly or again and again manually in order to find defects in the software according to customer requirement is called as Manual Testing.



Contents of manual testing

SDLC (Software development life cycle)

- 1. waterfall model
- 2. spiral model
- 3. v & v model
- 4. prototype mode
- 5. customised model
- 6. hybrid model
- 7 agile model

Software testing

White box testing

- 1. Path testing
- 2. Conditional testing
- 3. Loop testing
- 4. WBT from memory point of view
- 5. WBT from performance point of view

Black box testing

- 1. Functionality Testing
- 2. Integration testing
- 3. System testing
- 4. Acceptance testing
- 5. Performance Testing
- 6. Smoke Testing
- 7. Adhoc Testing
- 8. Accessibility Testing
- 9. Compatibility Testing
- 10. Globalisation Testing
- 11. Reliability Testing
- 12. Regression Testing
- 13. Usability Testing

→ Test case

→ STLC (Software Test life cycle)

→ Test plan

→ Defect life cycle/Defect tracking

QTP - quick test professional

SDLC(Software Development Life Cycle)

It is a step by step procedure or a standard procedure to start a new project or to develop a new software.

When does SDLC start? OR when we follow SDLC?

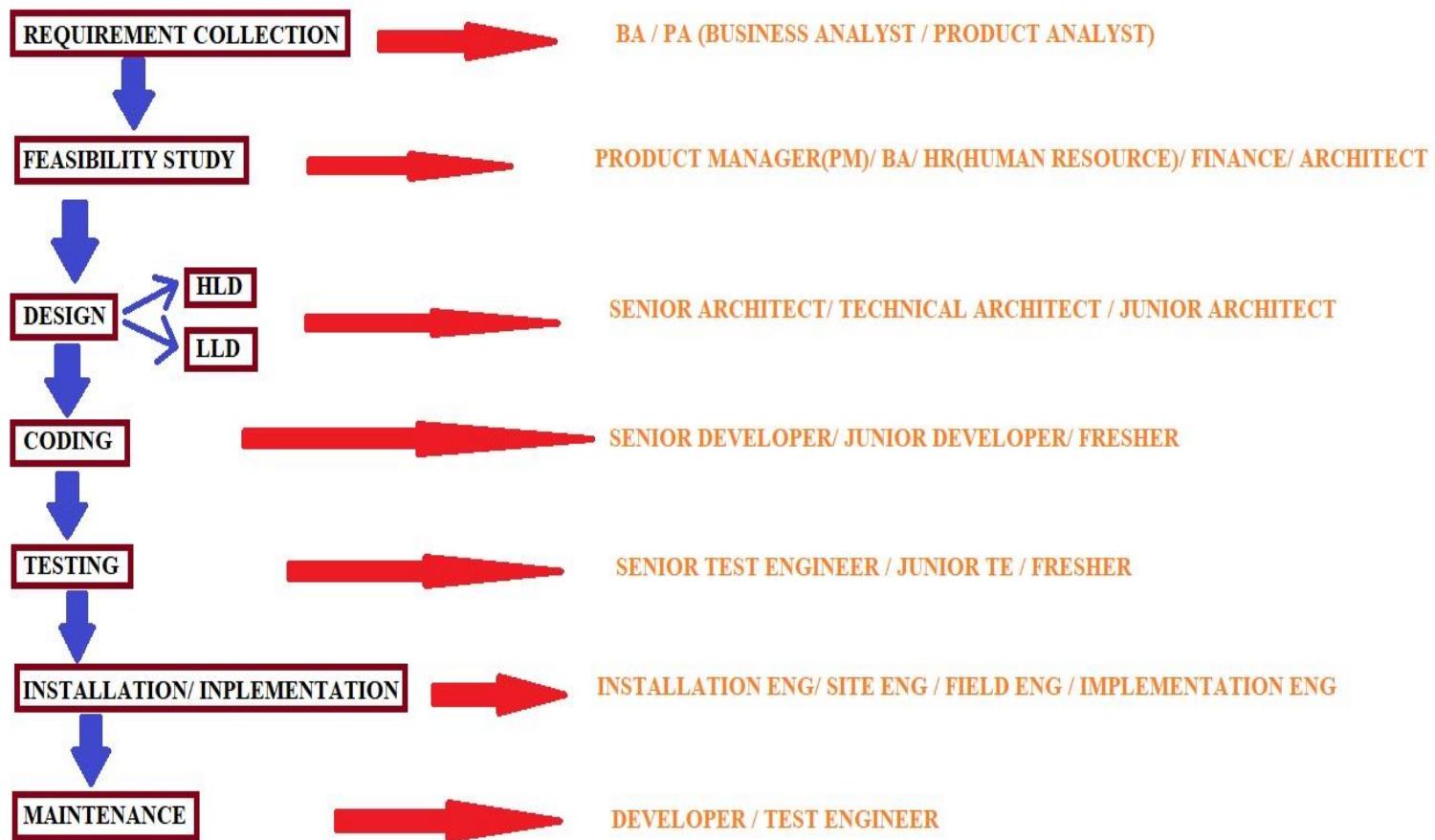
When a person or a company wants to start with a new project they have to start with SDLC.

Why we should follow SDLC ? or What will happen if you didn't follow SDLC ?

- > We will not get to know how many engineers are required to complete the project.
- > We will not get to know how much cost is required to complete the project.
- > There will be no supporting requirement documentation.
- > There will be delay in releasing software to the customer.

Flow
What?
When?
Why?
Satges?
Types?

STAGES OF SDLC



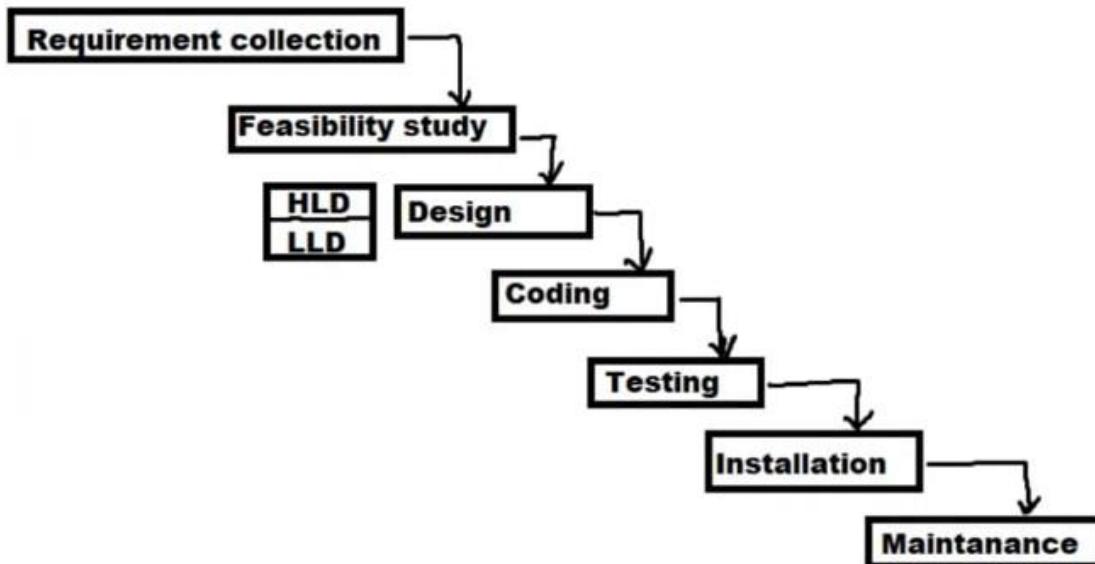
Types of SDLC:

- >Waterfall Model
- >Spiral Model
- >V&V Model
- >Prototype Model
- >Customised Model
- >Hibrid Model
- >Agile Model

Waterfall Model

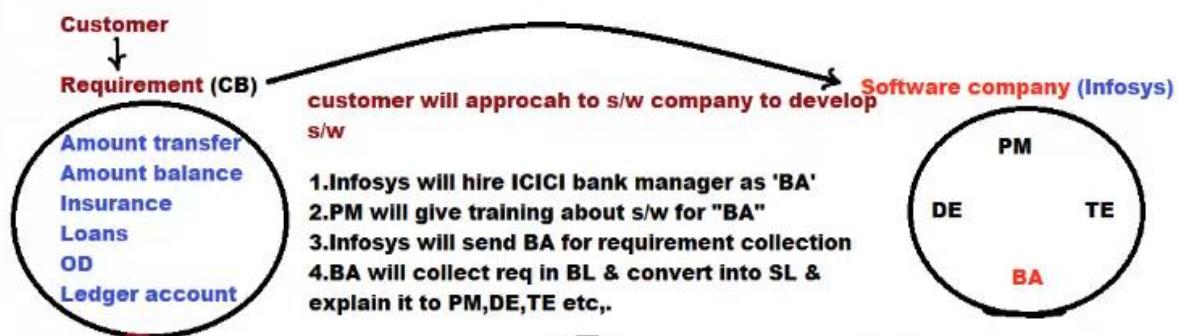
- >It is a step by step procedure or a standard procedure to develop a new software.
- >It is also called as traditional model or sequential model or basic model.

Stages of waterfall model:

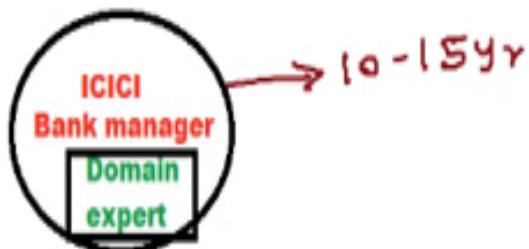


Requirement Collection:

- > It is a collection of the requirement from the customer place. It is done by Business Analyst or Product Analyst.
 - > BA from the company will go to customer place collect requirement in Business Language(BL) & come back to company, convert BL to Software Language(SL), explain it to the Project Manager(PM), Developers, Test Engineers, Architect,etc., this process is called as Requirement Collection.
- BA will act as bridge between customer and company.



BUSINESS LANGUAGE

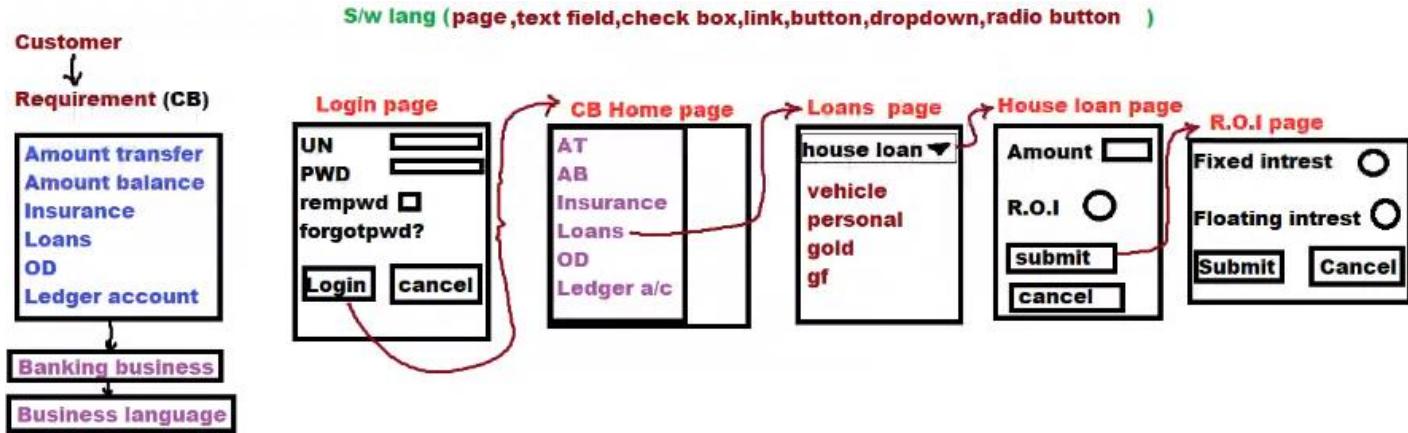


Who can become BA ?

- > **Domain Expert:** If a person is having around 15-20 years exp in same domain & has got very good knowledge on the domain is called as Domain Expert.
- > **Sr. Developer:** A person who is working on same project for around 8-10 years exp & has got very good knowledge on the product.
- > **Sr Test Engineer:** A person who is working on the same project for around 6-8 years exp & has got very good knowledge on the project.

Note: 70-80% of critical & complex projects BA will be present.

20-30% of small & simple projects BA will not be present in that case Sr. Developer or Sr Test Engineer will play role of BA.



Feasibility Study:

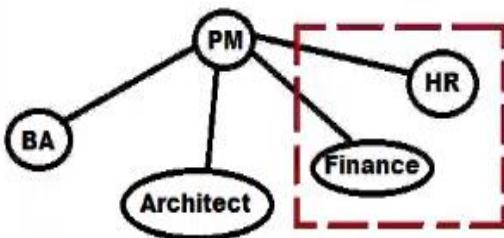
- > Once after Requirement Collection is done then we go for Feasibility Study.

It is done by a team which consists of Project Manager, Business Analyst, Architect, Finance Team and HR Team.

This is the stage where company will decide to take up the project or not & if they take up the project then company will check whether they have sufficient resources, sufficient technology & sufficient lab setup & this is the stage where company will get to know that if they take up the project do they get profit or not.

This process is called as Feasibility Study.

Role of Project Manager:

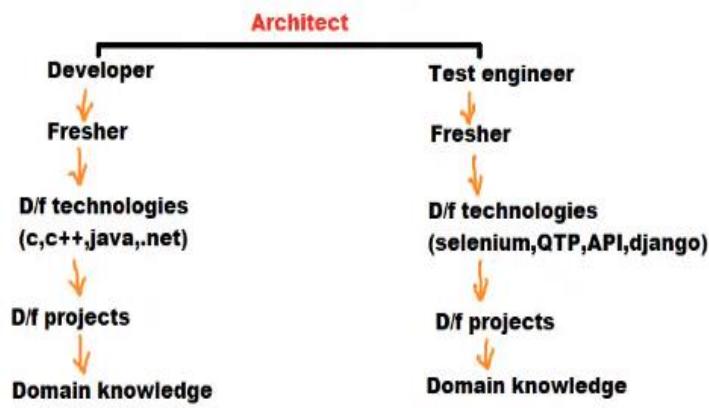


- > PM will interact with architect, BA, Finance team and HR team & he will decide whether to take up the project or not. Here Finance team and HR team can be optional.

Role of Architect:

> Architect will always think from technical point of view & he will decide technically is it possible to implement the project or not.

If it is possible which kind of technology should be used, this decision will be taken by architect.



Role of BA:

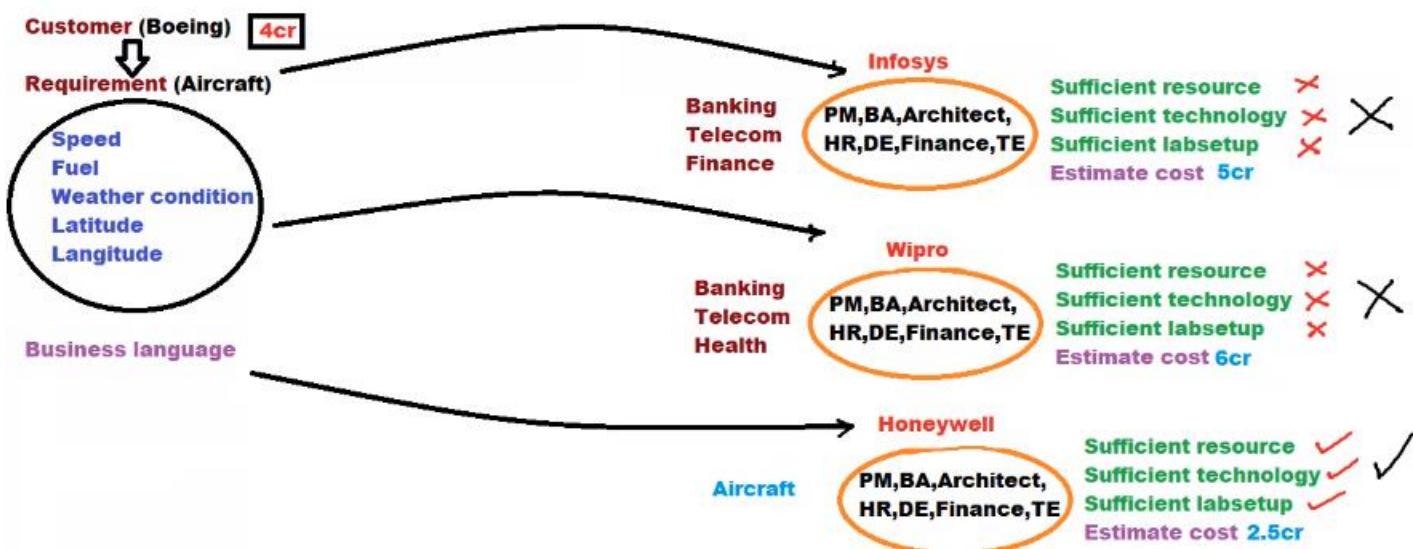
> He will be involved in Requirement Collection.

Role of Finance Team:

> Finance team will always think from money point of view & they will check what is operational cost required to work on the project & if they invest do they profit or not.

Role of HR:

> HR will always think from hiring/resources point of view.



Design:

> Once after Feasibility Study is done then we go for Design.

It is done by Architect or Sr. Architect or Technical Architect.

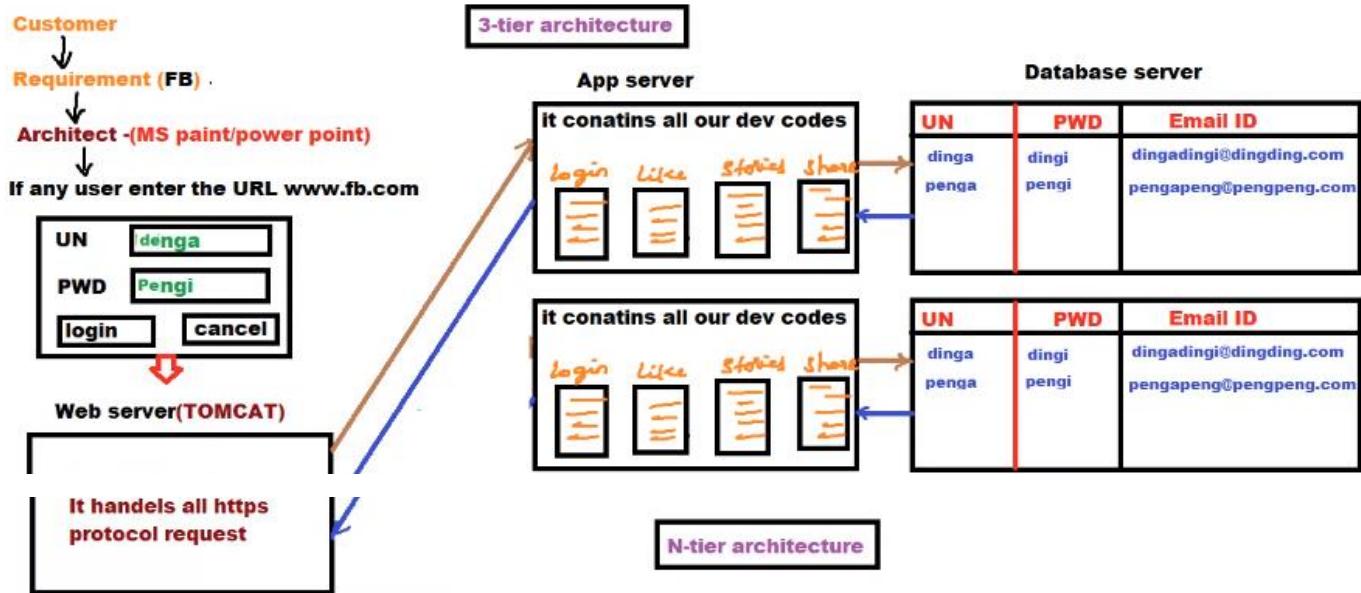
There are 2 types of Design:

- > High Level Design or External Design.
- > Low Level Design or Internal Design.

High Level Design:

> Example 1:

Designing the architecture of the s/w or project is called as High Level design.



> If there are 1 web server, 1 app server and 1 database server then it is called as 3-Tier Architecture.

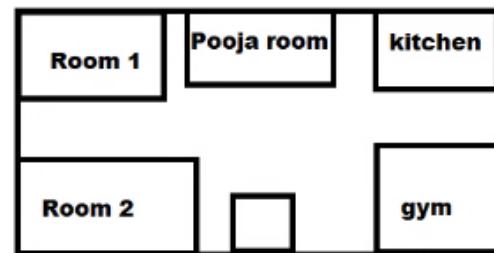
> If there are 1 web server, multiple app server and multiple database server then it is called as N-Tier Architecture.

WS: It will handle all https protocols request

AS: It contains all developer codes

DBS: It contains all the data of users

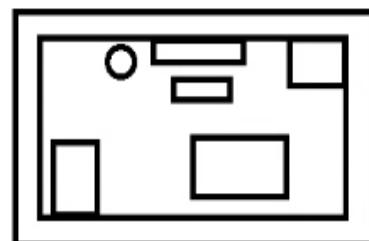
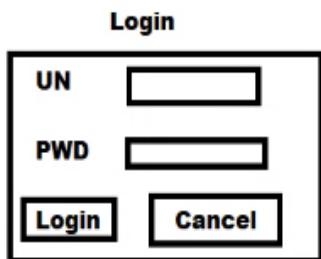
If you take any application, designing the main modules of the application is called as HLD.



Low Level Design:

> Example 1:

If you consider any application, designing each and every individual module in detail is called as Low Level Design.



Coding:

> Once after Design is done we go for coding.

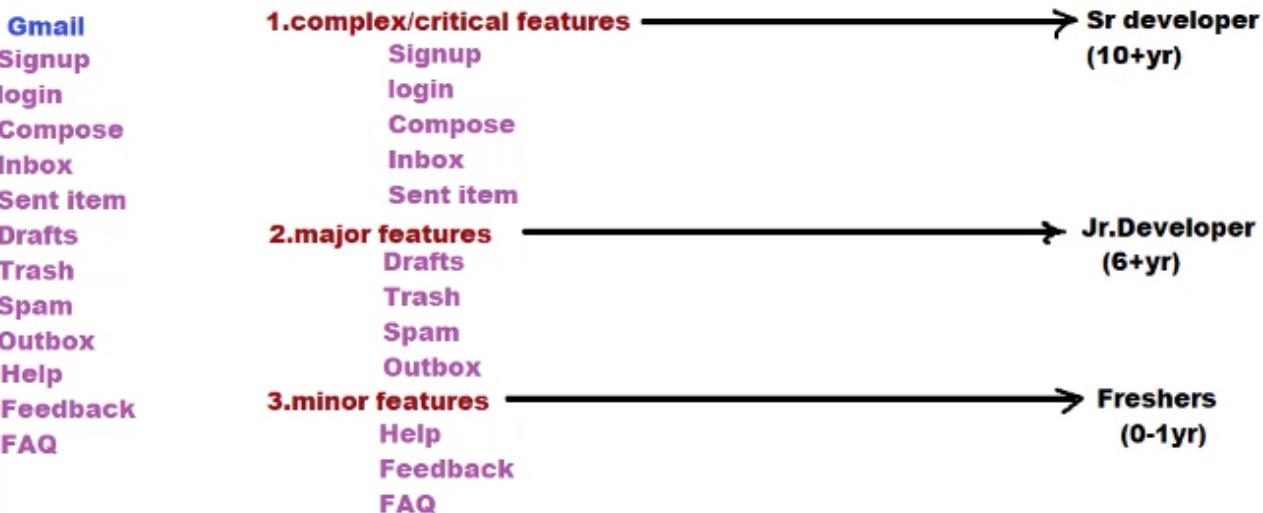
It is done by Sr.Developer, Jr.Developer & Fresher by looking into LLD & requirement.

Modules/features

1.complex/critical features

2.major features

3.minor features



Testing:

> Once after Coding is done, Developers will give s/w to TE wherein all the TE(Sr, Jr & Fresher) will start to test the s/w by entering all possible inputs into s/w to find defects in the s/w according to customer requirement is called as Testing.

Customer

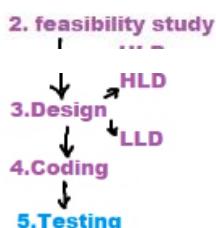


If any user opens browser & enter URL
www.cb.com

Login page

UN	<input type="text"/>
PWD	<input type="text"/>
<input type="button" value="Login"/>	<input type="button" value="Cancel"/>

- It should accept 2-8 characters
- it should accept min 5 char & include atleast 1 special char



Login page

UN	<input type="text"/>
PWD	<input type="text"/>
<input type="button" value="Login"/>	<input type="button" value="Cancel"/>

AB ✓
AB12 ✓
A*B2 ✗ → defect
*#/0.-?!@!68 ✗
AB* ✗
A1208C ✗
*!# ✗
001863421AZ ✗
! ? ✓

Note: In earlier days in waterfall model testing was done by Developer, but nowadays if you go to any company, if the company is following waterfall model, testing is done by TE only.

Character

A-Z,a-z,0-9,alphanumeric values(Ab12),special characters(!@#\$%^&*)(`{})

Why developer should not involve in Testing? or What are the drawbacks if developer involve in testing?

- > Chances are there developer may utilize all the testing time in coding.
- > Developer will be more concerned on developing the s/w rather than testing the s/w.
- > Developer will be over-confident.
- > Developer will always write the code & test it from positive point of view rather than testing from negative point of view.
- > While testing the s/w if developer finds any defect, he will try to hide the defect than fixing the defect.
- > Total time taken is more.
- > S/w quality will not be good.

Installation/Implementation:

- > Once after Testing is done, if the s/w quality is good, then we go for Installation/Implementation.
- It is done by IT Engineer or Site Engineer or Installation Engineer.

Installation/Implementation:

- > Once after Testing is done, if the s/w quality is good, then we go for Installation/Implementation.
- It is done by IT Engineer or Site Engineer or Installation Engineer.



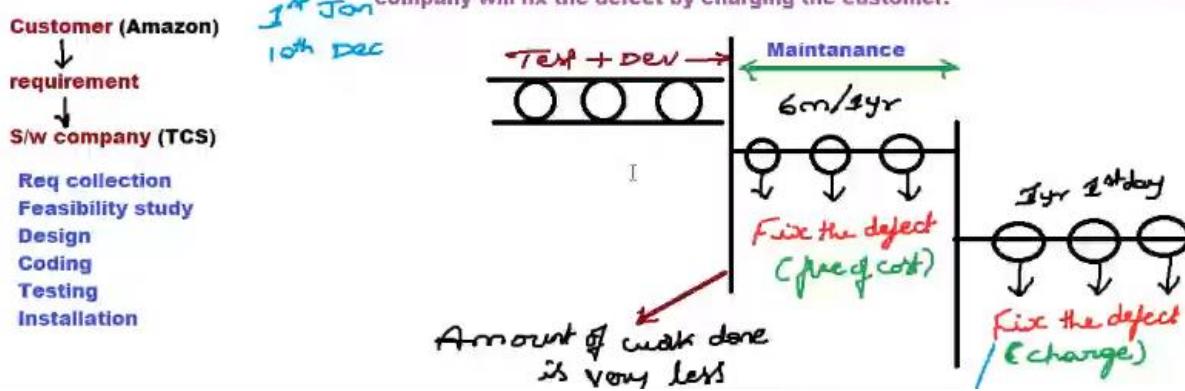
IT Engineer from the company will go to customer place, setup the environment & install the s/w & he will give demo to customer so that customer can use the s/w & run the business.

- > In earlier days Sr. Developer was involved in installation.
- > In the absence of Sr. Developer company under went loss.
- > Hiring separate engineer --> "IT Engineer/Site Engineer/Field Engineer".
- > Sr. Developer will train them 'how to install the s/w'.

Maintainance:

- > Once after s/w is installed customer will start using the s/w for business purpose. While using the s/w if the customer finds any defect, company will fix the defect without charging the customer & this may go upto 6 months or 1 year & it depends on the agreement between customer & the company.

This period is called as Maintainance Period. Once after maintainance period is completed, if the customer find any defect, company will fix the defect by charging the customer.

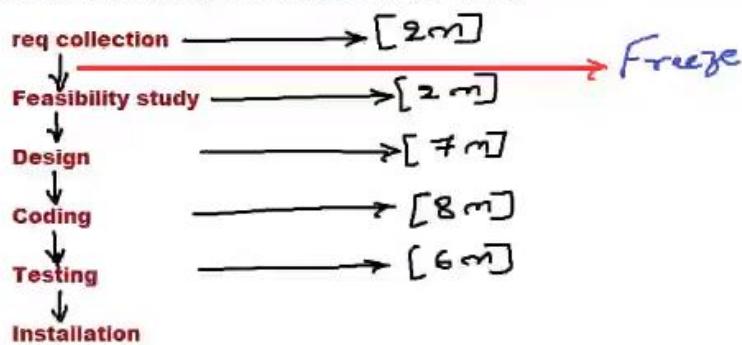


Who will be involved in maintainance?

The same developer & test engineer who will have built & tested the product will be involved in maintenance but the team size will be small, because the amount of work done will be less, i.e., here they might add/modify/remove the features requested by the customer & also they might fix the defect found by the customer.

Why the model is called as Waterfall Model?

Here backtracking is not possible, i.e., once after feasibility study is completed, requirements will be freezed, we cannot go back & change the requirement, so the model is called as waterfall model.



Why requirements keep changing? or Why customer will keep on changing the requirement?

- > Because of competition in the market.
- > As and when the business changes customer keeps on changing the requirement.
- > As and when technology changes software must be changed.

Advantages of Waterfall Model:

- > It is simple to adopt.
- > Initial investment is less.
- > We can expect a stable product at the very end because requirements are freezed in the beginning.

Drawbacks of Waterfall Model:

- > Requirement changes are not allowed.
- > It is not a flexible model.
- > Testing is the smallest phase which is done only after coding, so if there is a defect in requirement & design it flows till the end & leads to lot of rework because requirement & design is not tested.
- > It leads to lot of rework which increases the total cost of the project.
- > Turn around time(total time) taken to deliver the product to the customer is more.

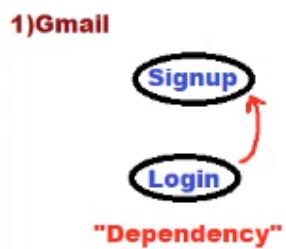
Applications of Waterfall Model:

- > Whenever we build small or simple applications.
- > Whenever we go for short term projects.
- > Whenever we are sure that customer is not going to change the requirement.

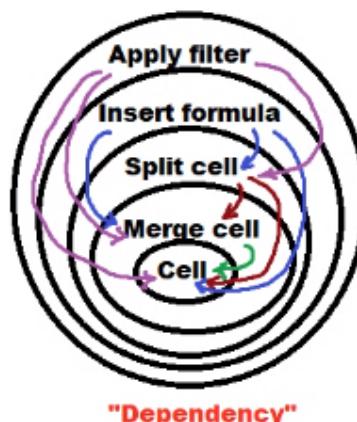
Spiral Model/Incremental Model/Iterative Model

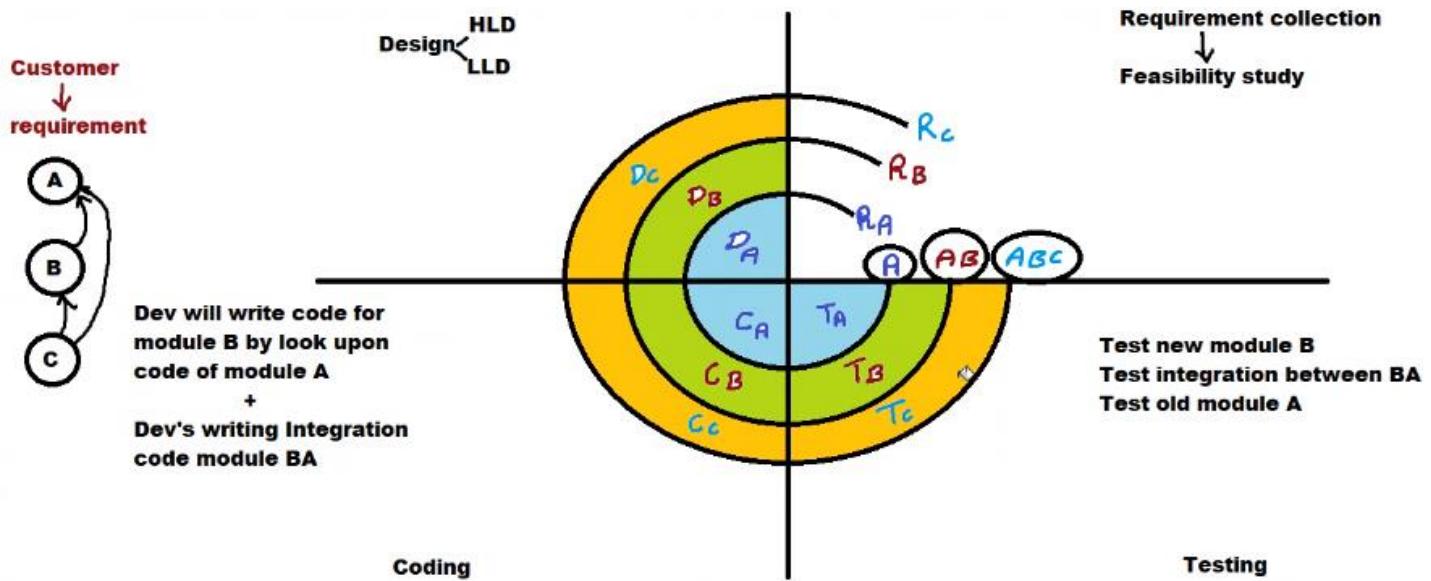
It is a step by step procedure or a standard procedure to develop a new software.

In order to overcome the drawbacks which was there in waterfall model we have come up with Spiral Model. We go for spiral model when there is dependency.



3)MS excel





- > Customer will give requirement for module 'A' in business language.
- > BA will do requirement collection for module 'A' & in the same stage feasibility study will be done.
- > Architect will do design for module 'A'(HLD & LLD)
- > Dev will do coding for module 'A' & give s/w to TE & TE will start to test module 'A' while testing module 'A' if TE finds any defect then TE will communicate to developer & developer will fix the defect & give new s.w to TE.
- > TE will uninstall old s/w & install the new s/w & test the defect first & continue testing for module 'A'. Once after testing is completed, if the s/w quality is good we will release the module 'A' to the customer.
- > When customer gives the requirement for new module 'B', BA will do the requirement collection for module 'B' & in the same stage feasibility study will be done for module 'B'.
- > Architect will do design for module 'B'(HLD & LLD).
- > Developer before doing the coding for module 'B' they take the old code of module 'A' & write the code for new module 'B'.
- > Since there is dependency from 'B' to 'A', developer will write the integration code for 'BA'.
- > Once coding is done developer will give the s/w to the TE. TE will start to test the new module 'B'. Then TE if he finds any defect in 'B' he will communicate the defect to the developer & developer fixes the defect & gives the new s.w.
- > TE will retest whether the defect is fixed in module 'B' & then TE will continue testing the moduel 'B'.
- > TE will test the integration between 'BA'.
- > TE will test old module 'A'.
- > If the product quality is good, they will release the the modules 'A' & 'B' to the customer.

Why the model is called as Iterative Model?

In every cycle same process is repeated so the model is called as Iterative Model(i.e., requirement collection, feasibility study, coding, testing....).

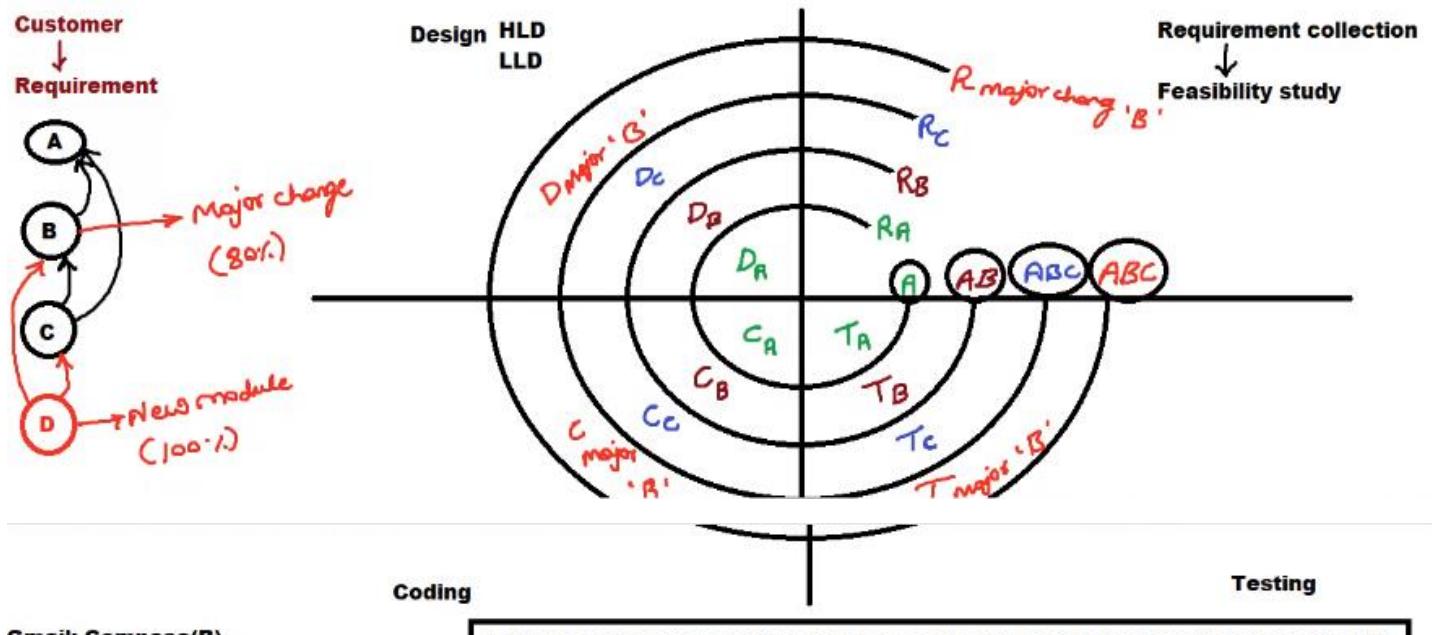
Why the model is called as Incremental Model?

In every cycle we keep on adding new module so the model is called as Incremental Model.

How to handle requirement changes?

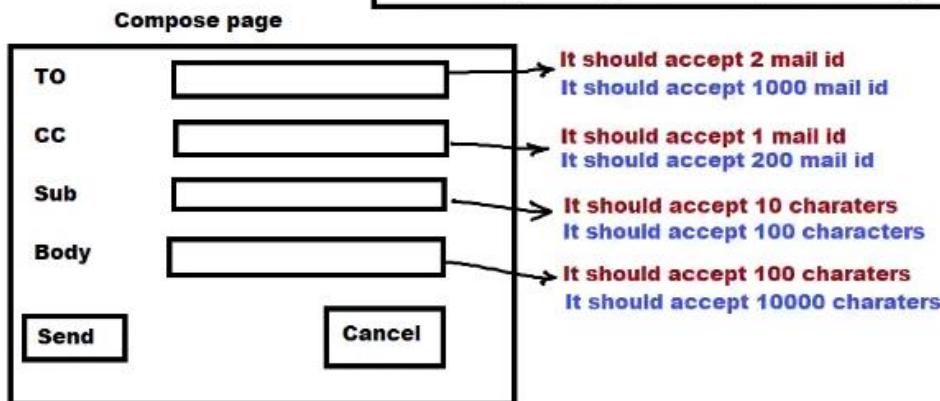
> There are 2 types of changes:

> Major Changes:

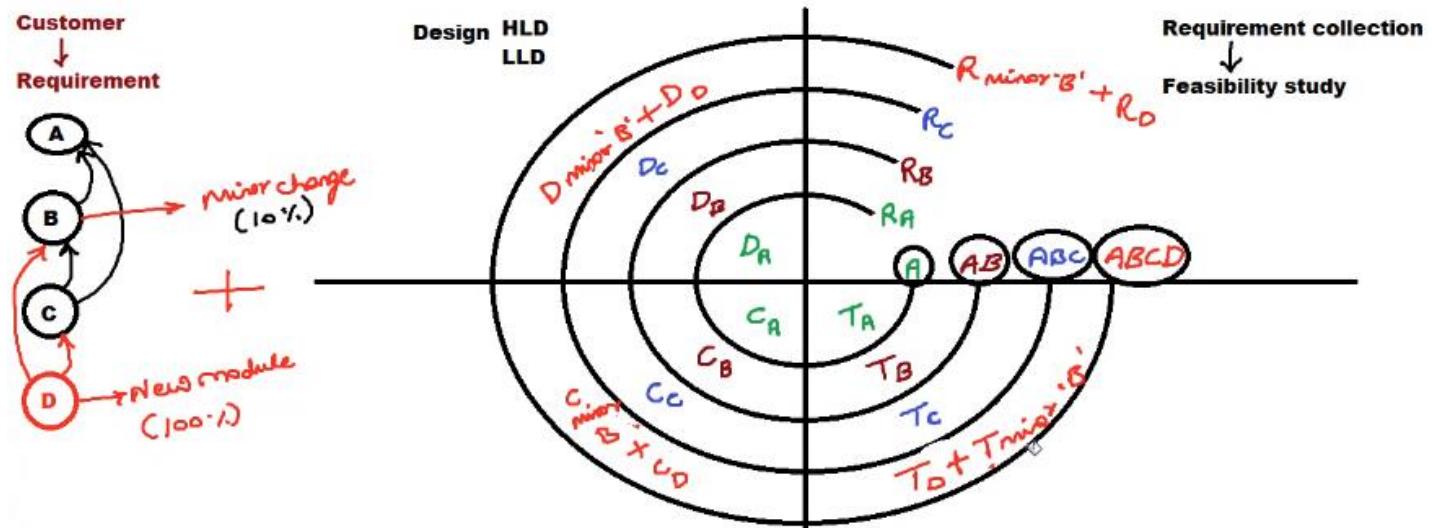


Gmail: Compose(B)

> Whenever customer gives major changes as well as new requirement, we will handle major changes first as a part of new cycle.

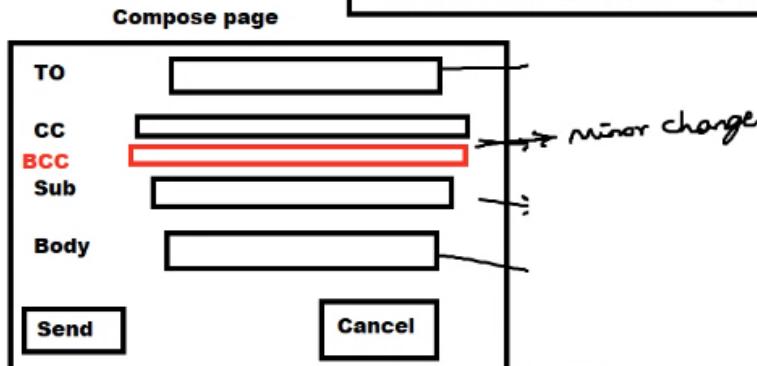


> Minor Changes:



Gmail: Compose(B)

> Whenever customer gives minor changes as well as new requirement, we will handle minor changes & new requirement as a part of same cycle.



Advantages of spiral model:

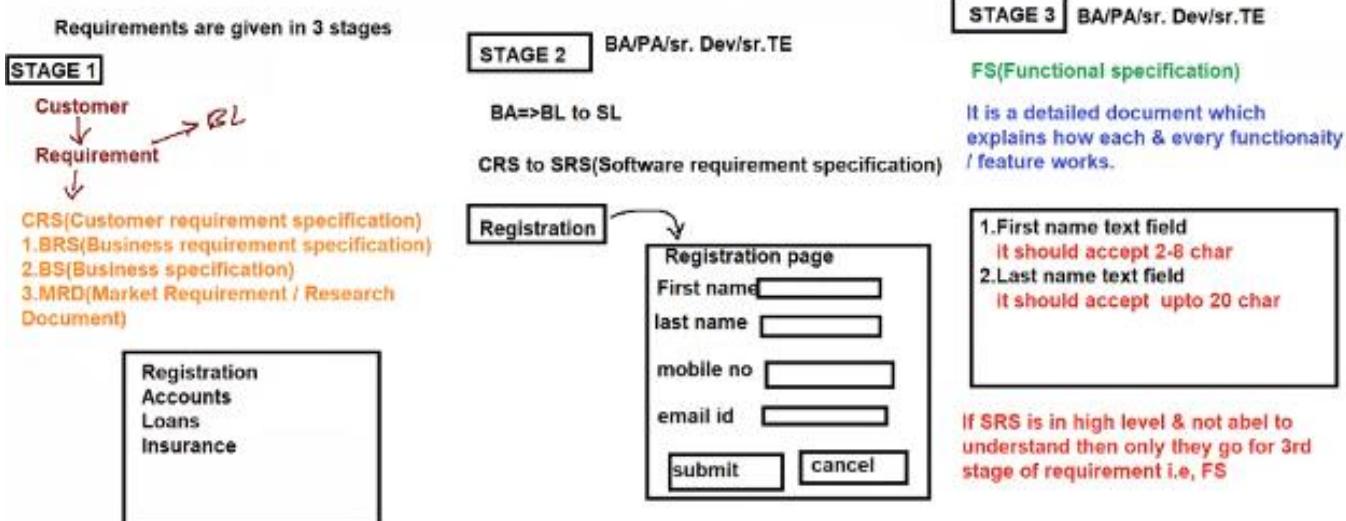
- > Requirement changes are allowed after each and every cycle.
- > Turn around time(total time) taken to deliver the product to the customer is less, because we are not developing full product at a time, but we are developing only 1 module at a time.
- > Software quality will be good.
- > It is a **Controlled Model** (here we take 1 module, develop the module, test the module, make sure that the quality is good, then only we take requirement for next module).

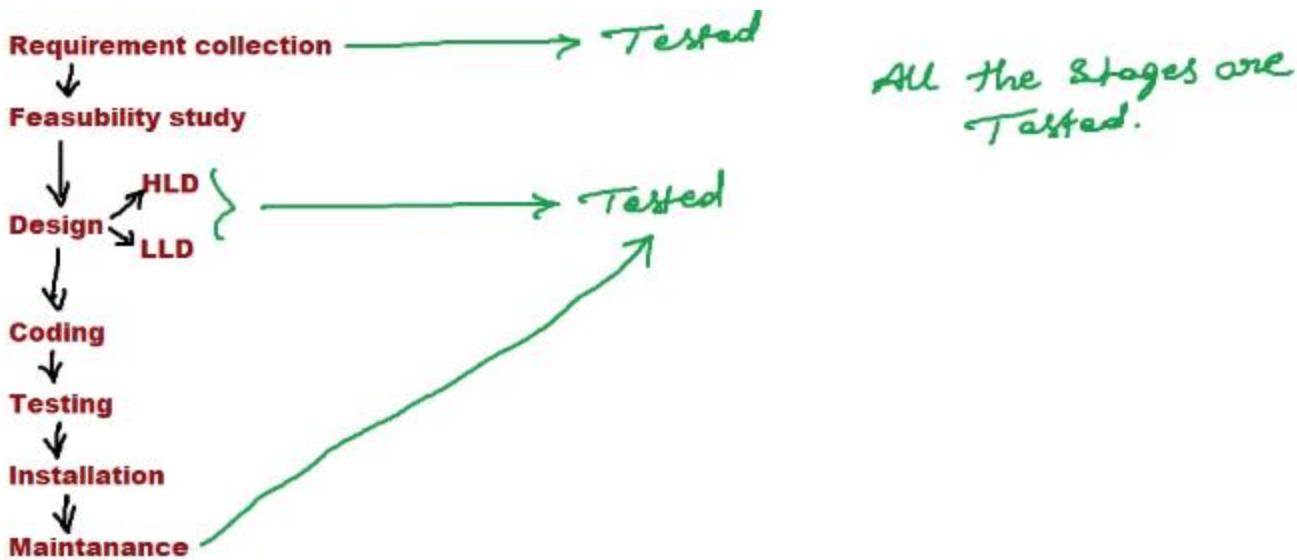
Drawbacks of spiral model:

- > Every cycle is similar to waterfall model.
- > Requirement & design is not tested.
- > Requirement changes are not allowed in between the cycle.

Applications of spiral model:

- > Whenever there is a dependency between the modules.
- > Whenever the requirements are given in stages.





Note: To overcome the drawbacks which was there in both waterfall model & spiral model we have come up with V&V Model, i.e., In waterfall model requirement changes was not allowed & in both waterfall model and spiral model requirement & design was not tested.

Verification: Verifying or testing CRS, SRS, HLD, LLD & check whether it is working according to customer requirement or not is called as Verification.

It is done by TE & it is done before s/w development.

It is also called as Static Testing.

Here we ensure that are we building product right or system right or s/w right.

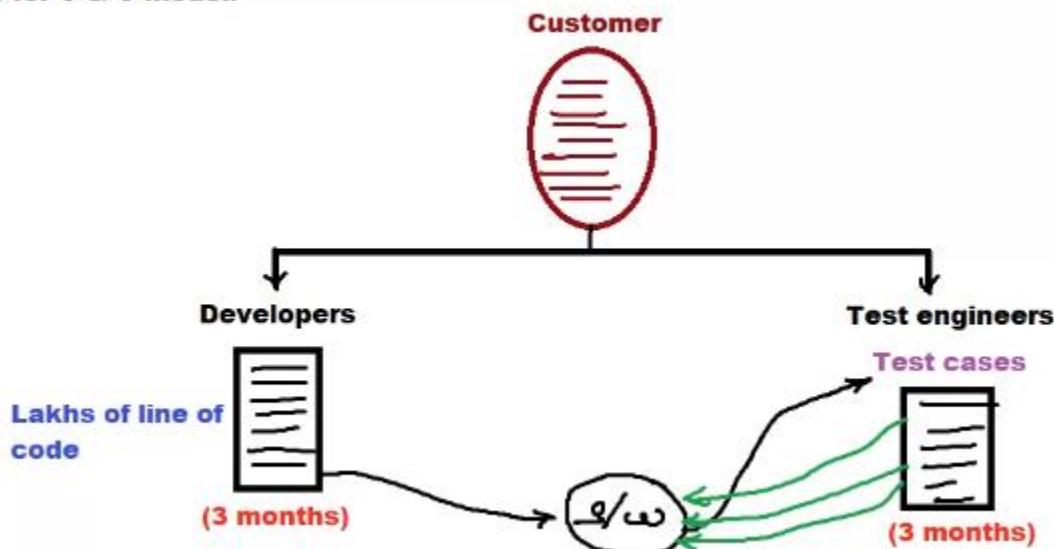
Validation: Testing the functionality of an application or s/w by executing test cases is called as Validation.

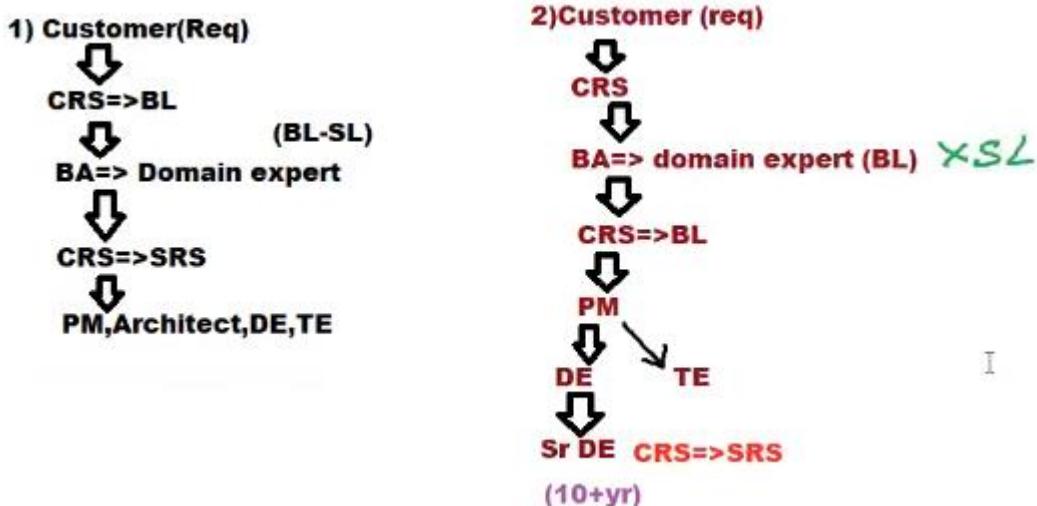
It is done by TE & it is done after s/w development.

It is also called as Dynamic Testing.

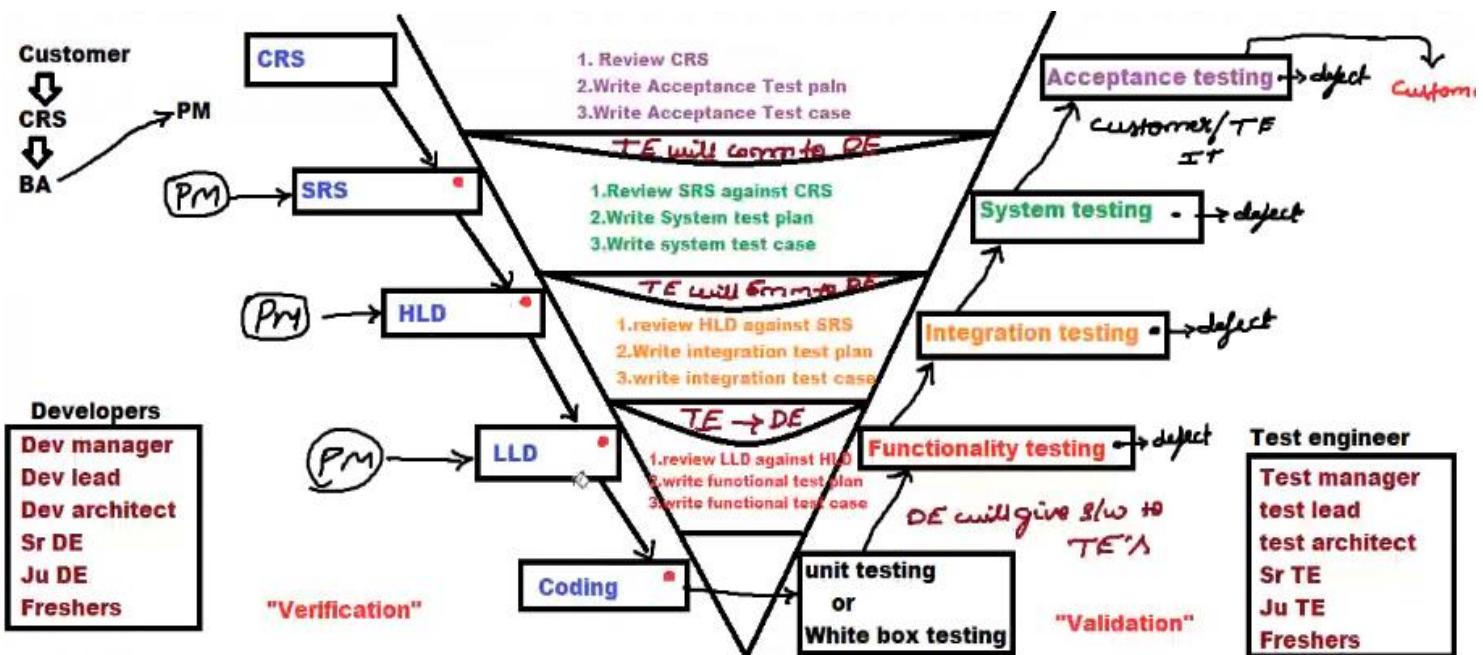
Here we ensure that are we building right product or right system or right s/w.

Example for V & V model:





V AND V MODEL



Explanation of V&V Model:

- > Customer will give requirement in the form of CRS, that will be in BL.
- > BA will go to the customer place, collect CRS & explain it to PM in BL.
- > PM will take up the CRS & give it to both developer & TE.
- > TE will review CRS, write Acceptance Test Plan & Test Case, parallelly developer will be converting CRS to SRS. While reviewing CRS if TE finds any defect [conflicting req, missing req, wrong req] he will communicate defect to the developer. Developer will cross check & confirms the defect & developer will communicate to customer. Customer will update the CRS & send updated CRS to both developer & TE. TE will review updated CRS & also update corresponding Test Plan & Test Case, parallelly developer will update defect in SRS & continue converting CRS to SRS. Once after it is done SRS will be ready.
- > PM will take up SRS & send it to both developer & TE. TE will review SRS based on CRS & also write System Test Plan & Test case, parallelly developer will be converting SRS to HLD. Once both developer & TE complete the work HLD will be ready.

> Like this flow continues till coding. Once after coding is completed developer will start testing each & every line of the code, nothing but they do WBT or Unit Testing. While doing WBT if developer finds any defect in the code he will fix the defect and WBT again. After WBT developer will give s/w to TE. TE will start with Validation activities, wherein first TE will do Functionality Testing by executing Functional Test Case, by doing this if TE finds any defect he will communicate the defect to the developer & developer will fix the defect perform WBT & give new s/w to TE. TE will uninstall the old s/w & install the new s/w test the defect & continue doing Functionality Testing. After doing Functionality Testing TE will do Integration Testing, System Testing by executing corresponding Test Cases.'

Advantages of V&V Model:

- > Testing starts in the early stage of project development i.e., from requirement collection stage itself.
- > All the stages are tested here because of this it avoids downward flow of defects.
- > Total cost will be very less.
- > Requirement changes are allowed.
- > The output is given simultaneously, because of this project gets completed very fast.

Drawbacks of V&V Model:

- > Documentation will be more (in every stage we are writing test plan & test case).
- > Managing the interaction between developer & TE is difficult.

Applications of V&V Model:

- > Whenever we go for long term projects.
- > Whenever we build complex applications or large applications.
- > Whenever customer is expecting high quality product within short span of time.

Review CRS

When we review CRS we might find defects like
1.Conflicting requirement
2.Missing requirement
3.Wrong requirement

1.Conflicting requirement

- 1.
- 2.
- 3.

Confusion

page 30:when enduser apply for loan loan manager has not having authority to approve the loan.

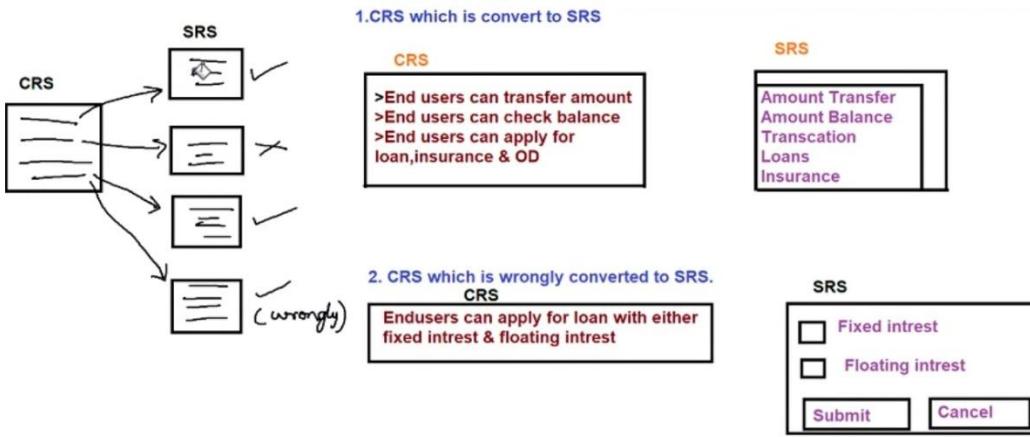
page 60:when endusers apply for loan if loan amout is
a)<20000 loan manager should approve
b)>20000 & <50000 branch manager should approve
c)>50000 regional manager should approve

2.Missing requirement

- 1.endusers can apply for loan
Approval is not given ✗
- 2.endusers can apply for insurance
Approval is not given ✗
- 3.endusers can apply for OD
Approval is not given ✗

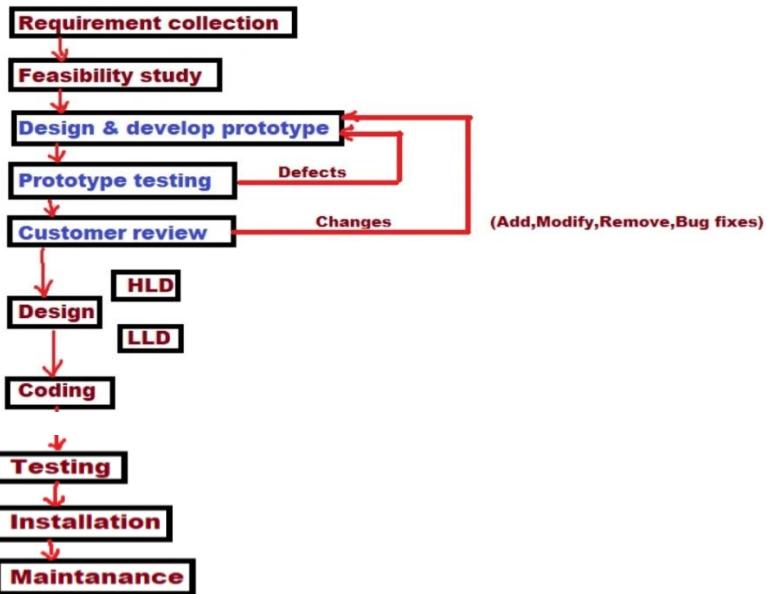
3.WRONG REQUIREMENT

- 1.Endusers can apply for loan by selecting fixed intrest & we charge 10% for fixed intrest
- 2.Endusers can apply for loan by selecting floating intrest & we charge 10% for floating intrest



Prototype Model/Dummy model

Diagram for Prototype Model:



Example stating why or when we go for Prototype Model?

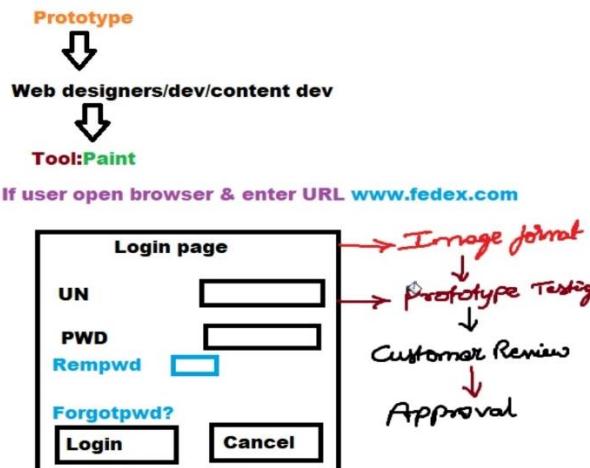
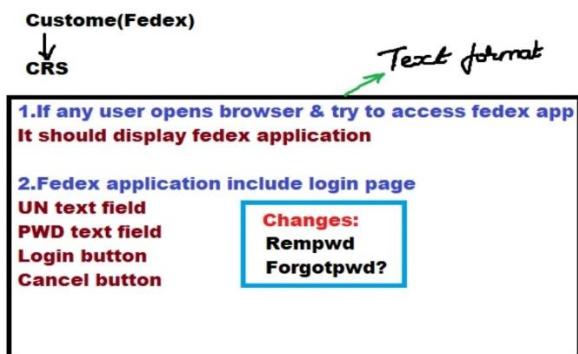


- > According to above example customer Fedex rejected the s/w because of misunderstanding & miscommunication.
- > In order to overcome this kind of problem company should follow Prototype Model.

Prototype testing?

It is a dummy model prepared by the web designer or content developer or developer, wherein they convert **text format** into **image format** by using tools like Adobe Photoshop or MS Paint or Picassa or Web HTML.

EXAMPLE of PROTOTYPE MODEL



What is the difference between Prototype Testing & Actual Testing ?

> Prototype Testing:

It is the testing done to check whether customer given requirement is present in prototype or dummy page is called Prototype Testing.

or

It is the testing done to check whether all the text format is converted to image format is called as Prototype Testing.

> Actual Testing:

Testing the functionality of an application by entering all possible inputs into the s/w to find defects in the s/w according to customer requirement is called as Actual Testing/Final Testing.

Advantages of Prototype Model:

- > There will be improved communication between customer, development & testing team.
- > Initially itself customer will get to know what is the outcome of the product.
- > Initially itself developer will get to know what exactly they have to develop.
- > Customer can request for changes in the beginning itself i.e., requirement changes are allowed & also easy to handle it.
- > We can set high expectation for the customer.

Drawbacks of Prototype Model:

- > Actual development of the application or s/w starts very late because they will be busy in designing & developing prototype.
- > There is investment needed just to build the prototype.

Applications of Prototype Model:

- > When customer is new to business.
- > When customer is not aware of full requirements.

Customized Model/Derived Model

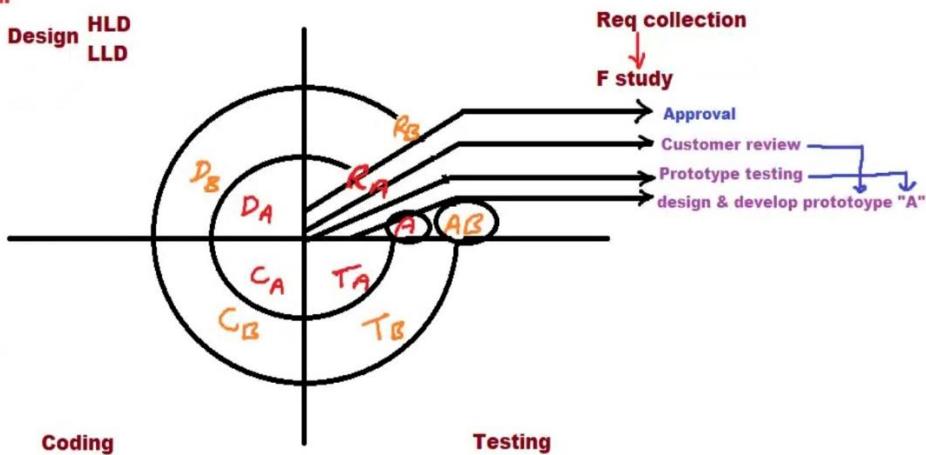
Here we take Basic Model or any single model & we change the model as per the customer requirement & company standards, that changed model is called as Customized Model/Derived Model.



Hybrid Model

The process of merging or combining more than 1 model into a single model is called as Hybrid Model.

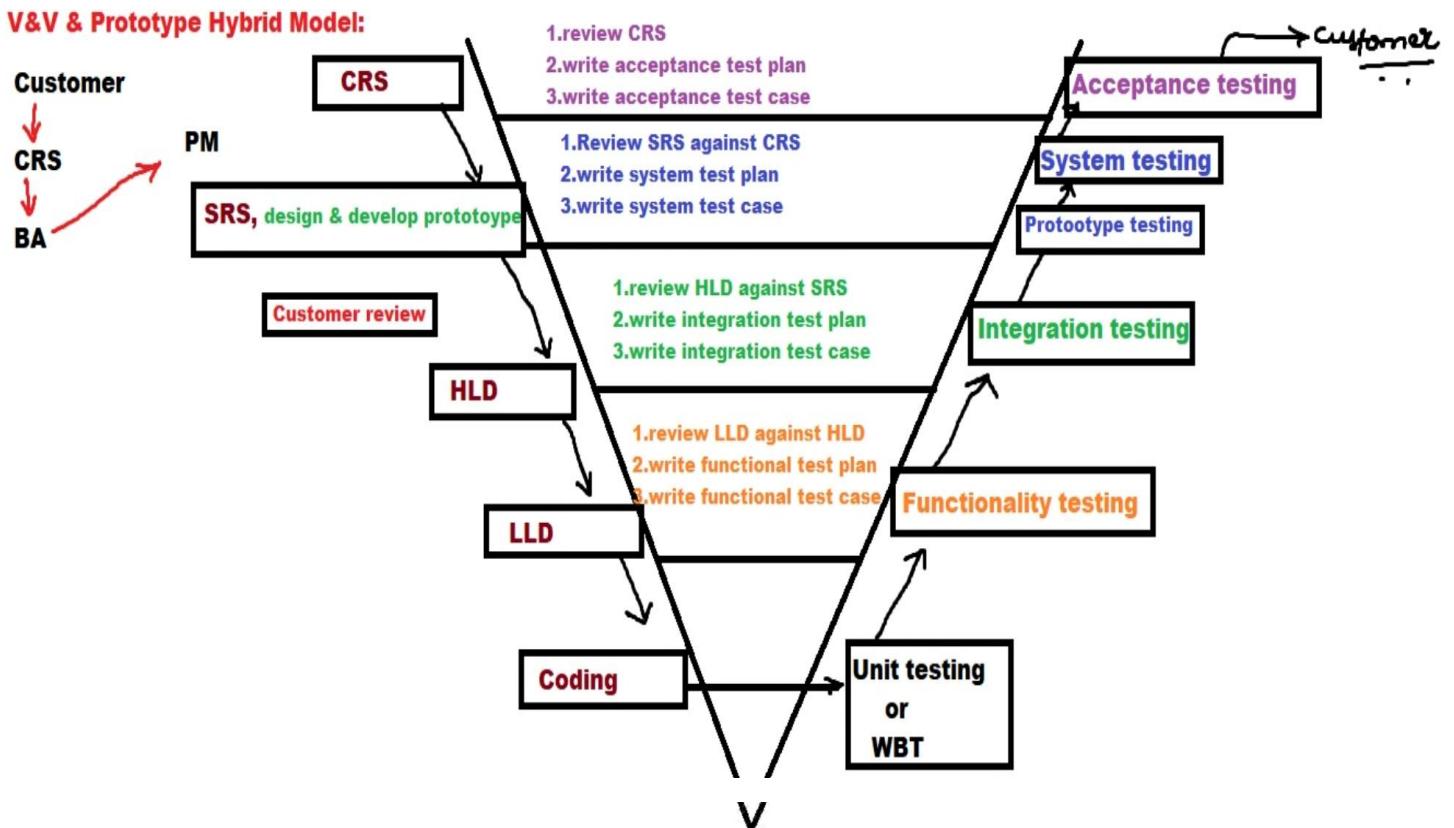
Spiral & Prototype Hybrid Model:



Application:

- > Case 1
 - > When there is a dependency between the modules
 - > When customer is new to the business

- > Case 2
 - > When requirement are given in stages
 - > When customer is not aware of full requirement



Application:

> Case 1

> When we go for long term projects

> When customer is new to the business

> Case 2

> When customer is not aware of full requirement

> When customer is expecting high quality product within short span of time

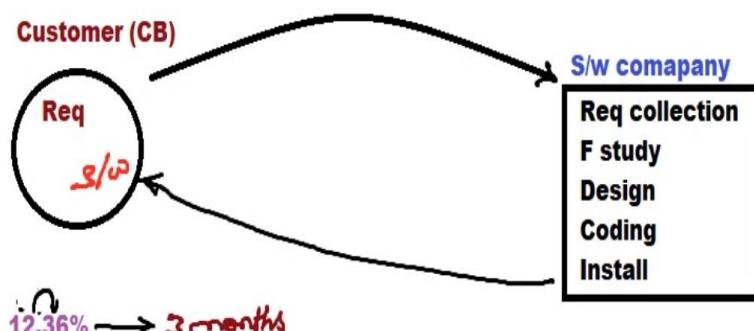


Software Testing

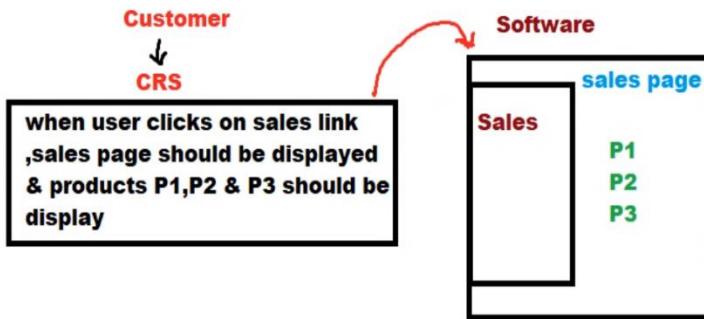
The process of identifying defects in the s/w is called as Software Testing.

or

Verifying the functionality of an application against requirement specification is called as Software Testing.

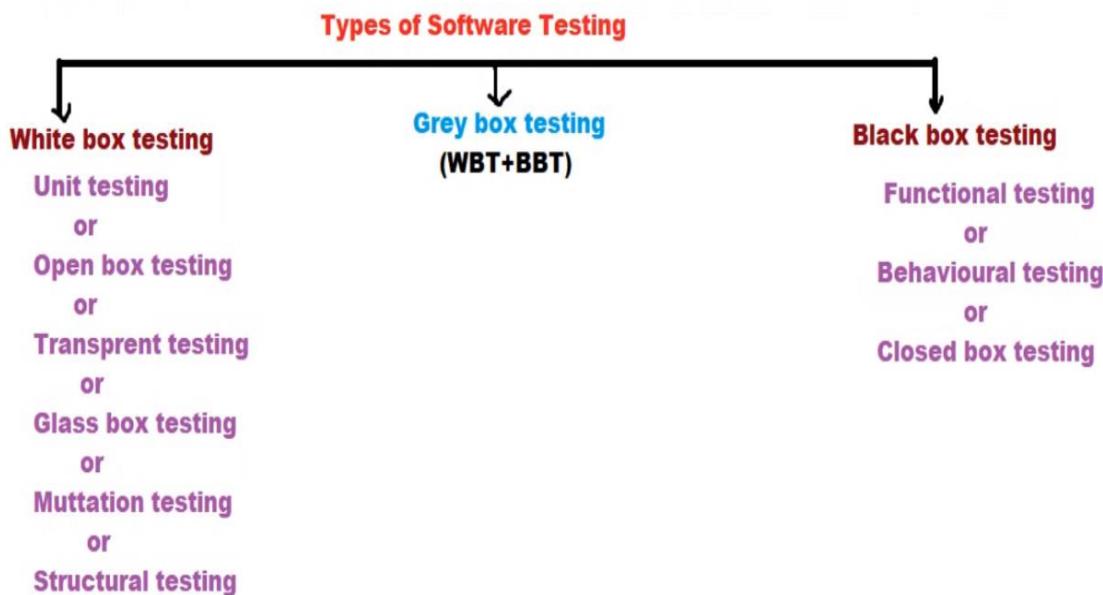


123.6%



Why we should do Software Testing?

- > Every s/w is developed to support the business, if there are any defects in the s/w it affects the business, so before we use s/w for business we have to find all the defects & fix all the defects, i.e., s/w must be well tested.
- > To improve the quality of the product
- > To check whether the product is working according to customer requirements.



what are the ways of s/w testing?

Manual & automation

what are the types of s/w testing?

WBT & BBT

White Box Testing:

Testing each & every line of the code is called as White Box Testing.

It is done by developers.

Since the code is visible it is called as Open Box Testing.

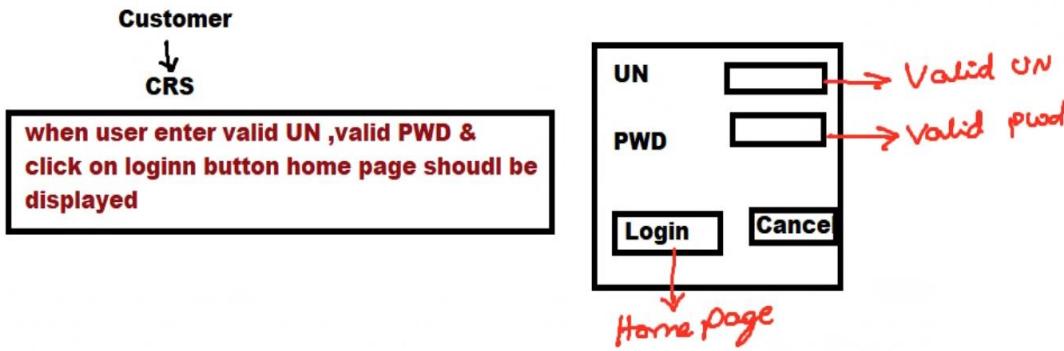
The smallest unit in the s/w is 1 line of code it is called as Unit Testing.

Black Box Testing:

Verifying the functionality of an application against requirement specification is called as Black Box Testing.

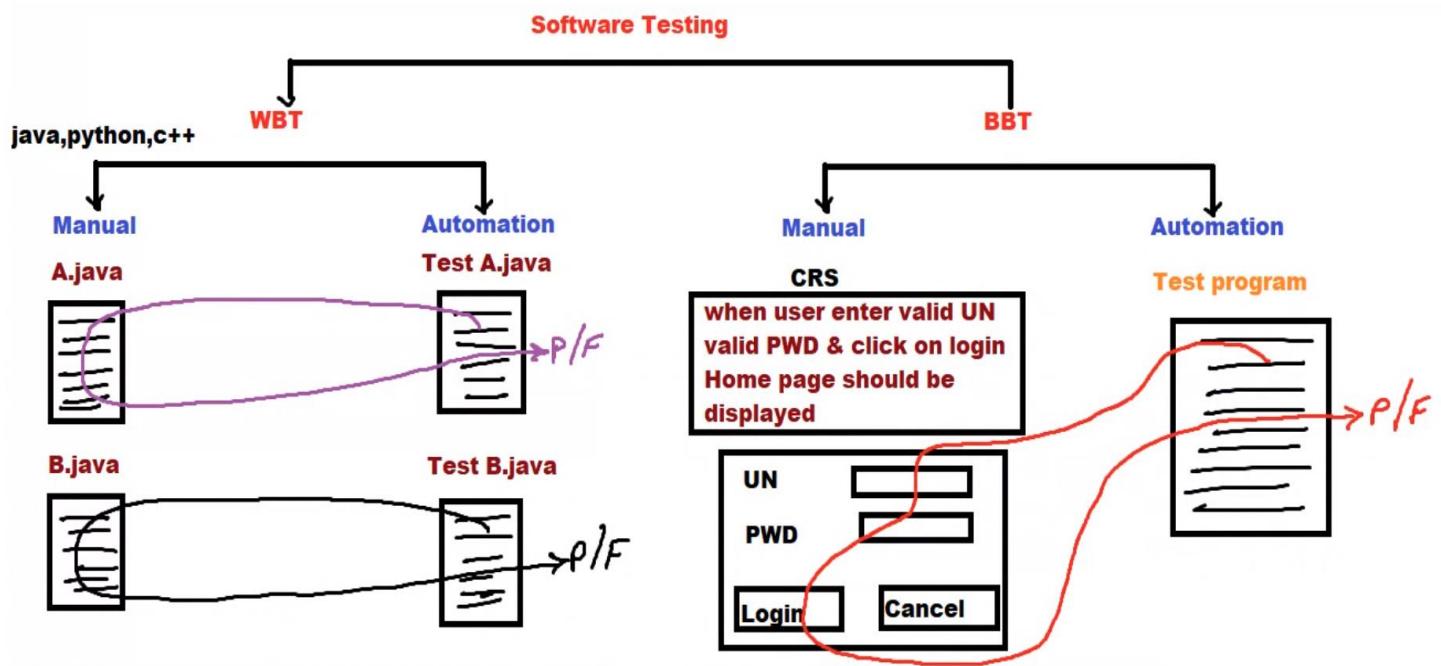
It is done by TE.

Since the code not visible it is done also called as Closed Box Testing.



What will happen if TE tries to fix the defect? or Why TE should not involve in fixing the defect?

- > If TE spent time in fixing 1 defect, he will not get time to test the remaining features.
- > If TE spent time in fixing 1 defect, he will not get time to catch the remaining bugs.
- > Chances are there fixing 1 defect will impact on the other features & because of that it will cause 1 more defect.



White Box Testing:

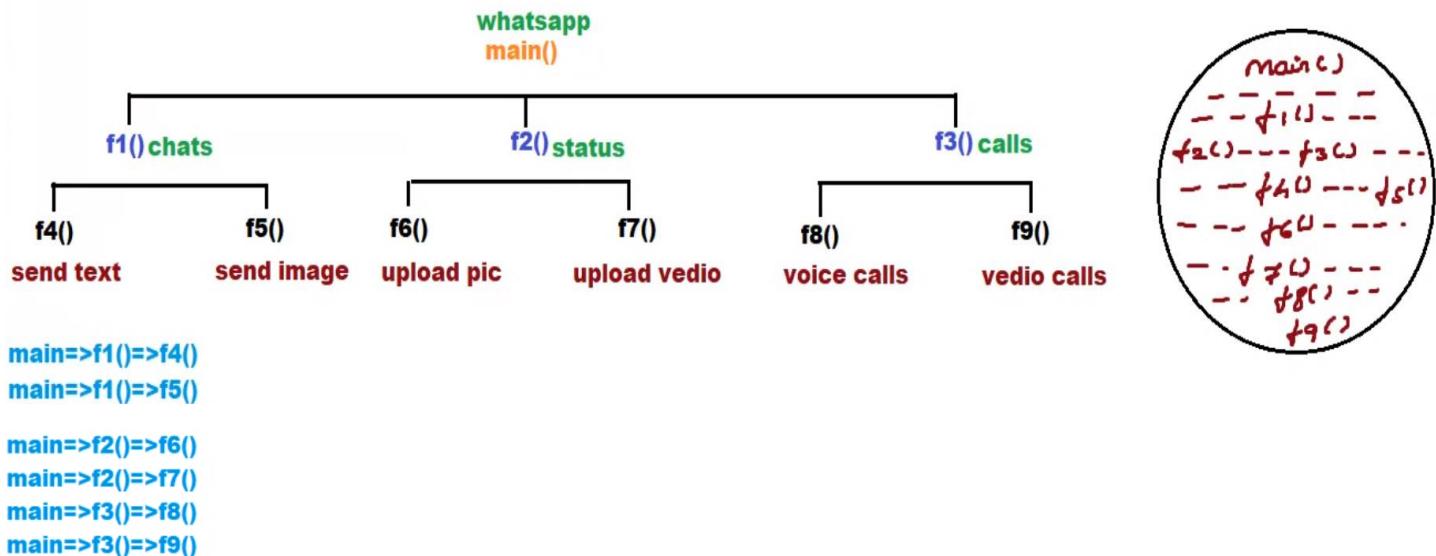
Testing each & every line of the code is called as White Box Testing.
It is done by developers.

Types of White Box Testing:

- > Path Testing
- > Conditional Testing
- > Loop Testing
- > WBT from memory point of view
- > WBT from performance point of view

Path Testing / Cyclomatic Complexity:

Here developer will write the flowchart & test all the individual paths, this is called as Path Testing.



Advantages:

- > Developer can find defects in the early stage itself.
- > Developer can ensure that there are no duplicate paths.
- > Developer can ensure that each & every individual path is thoroughly tested.

Conditional Testing:

Here developer will test for all the logical conditions, i.e., for both True & False conditions.

I
syntax: **if(condition)**
{

True condition
}
else
{

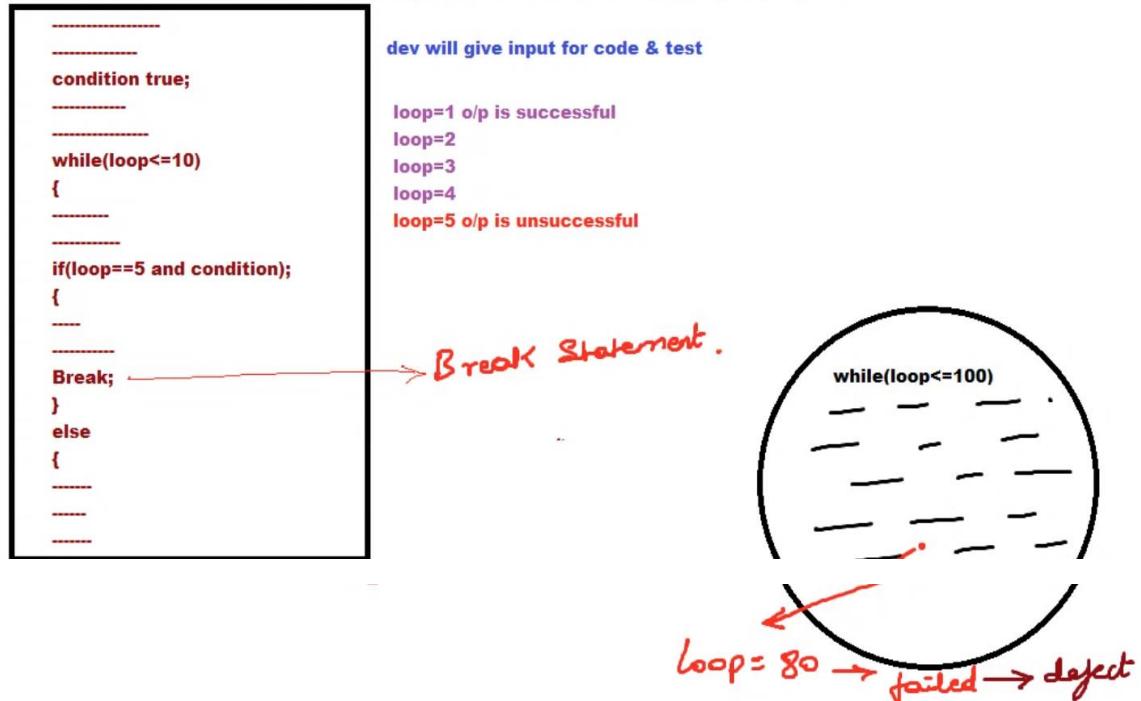
False condition
}

```
public static void main (String[] args)  
{  
int x=30;  
if(x==10)  
{  
System.out.println("The value of x":10);  
}  
else if(x==20)  
{  
System.out.println("The value of x":20);  
}  
else if(x<30) → instead of "!="  
{  
System.out.println("The value of x":30);  
}  
else  
{  
System.out.println("Invalid input");  
}  
}
```

- > Here developer will test the code for both true & false condition & ensure that code is working for both true & false condition, because chances are there sometimes code might work for true condition & may not work for false condition & vice versa, so to make sure that code is working for both true & false conditions developer will do Conditional Testing.

Loop Testing:

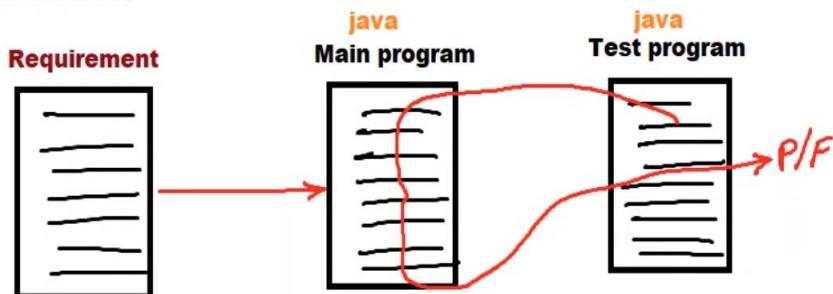
Here developer will test the loop & ensure that loop is repeating for defined no. of cycles, this is called Loop Testing & also developer will make sure that whether terminating condition is working properly or not.



> According to above example, while loop should repeat for 10 times, so developer will manually pass the input & start doing loop testing. Suppose if the loop is failed for the 5th time then it is a defect. Developer will fix the defect & again manually he will test the loop from the beginning & ensure that loop is working for 10 times. This is how developer will do loop testing.

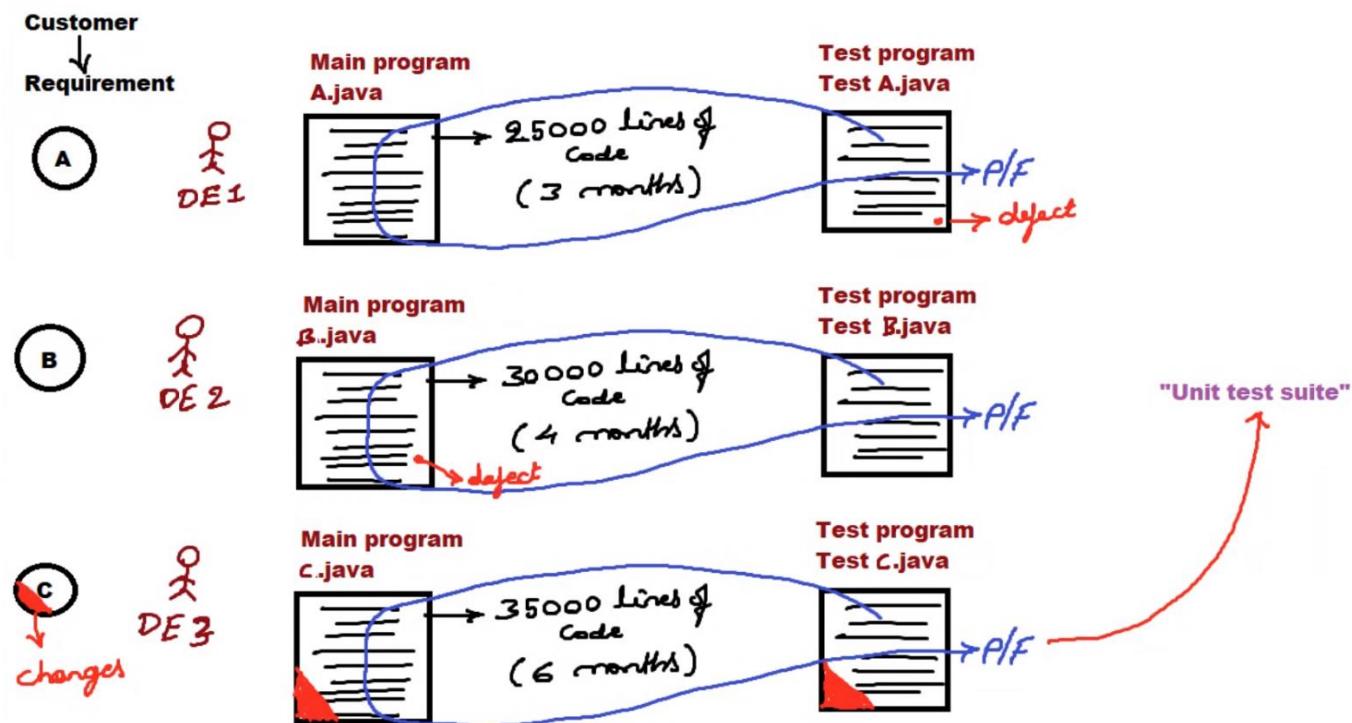
Note: If the loop is written for less no. of times developer will do loop testing manually. Suppose if the loop is written for more no. of times, doing loop testing manually is very tough job for developer, so developer will go for automation.

Automation WBT:



> Customer will give requirement & developer will write main program & corresponding test program in the same language & run the test program against the main program, wherein test program will test the main program automatically & gives the result as pass/fail this is called Automation WBT.

> While doing automation WBT we came across 3 different cases



Case 1: Assume that error is in Test Program.

In this case developer will fix the error in the test program & run the test program against the corresponding main program till the result is pass.

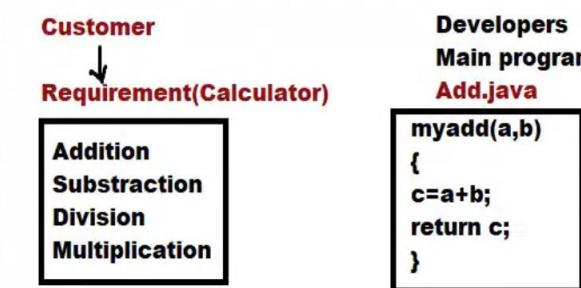
Case 2: Assume that error is in the Main Program.

In this case developer will fix the error in the main program & run the corresponding test program against the main program & chances are there fixing the defect in the main program might effect other main programs, so developer will run the entire 'Unit Test Suite'.

Case 3: Assume that customer will change the requirement for module 'C'.

Then developer will modify or change main program 'C' & also corresponding test program 'C' & run test program 'C' against main program 'C' & chances are there changing requirement for module 'C' might effect modules 'A' & 'B' so developer will run the entire 'Unit Test Suite'.

example:



Modules or
features or
Functionalities

Test program

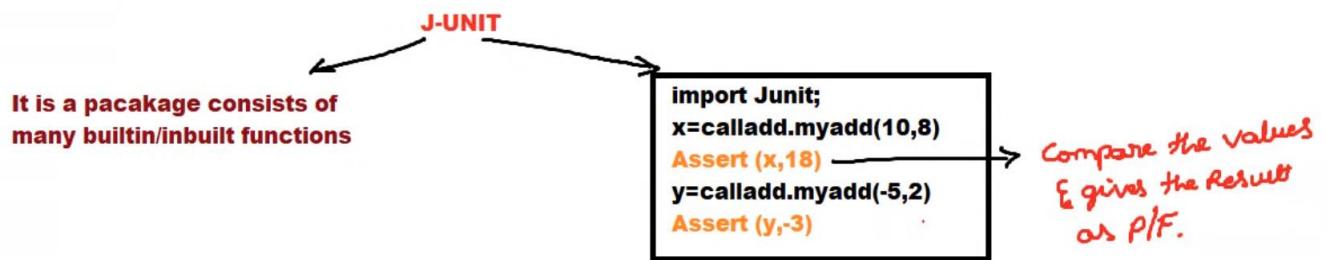
Test add.java

```
x=calladd.myadd(10,8)
if(x==18)
{
    System.out.println("pass");
}
else
{
    System.out.println("fail");
}
```

```
y=calladd.myadd(-5,2)
if(y===-3)
{
    System.out.println("pass");
}
else
{
    System.out.println("fail");
}
```

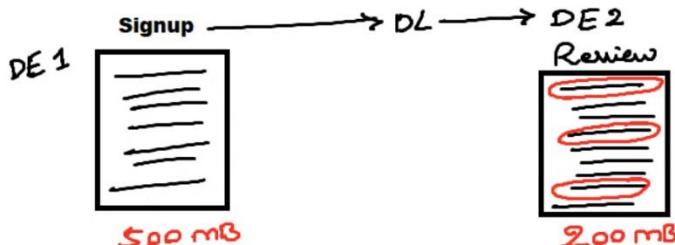
```
z=calladd.myadd(10,abcd)
if(z==error)
{
    System.out.println("pass");
}
else
{
    System.out.println("fail");
}
```

> To reduce the number of lines in test program, we use a tool called **J-Unit**.

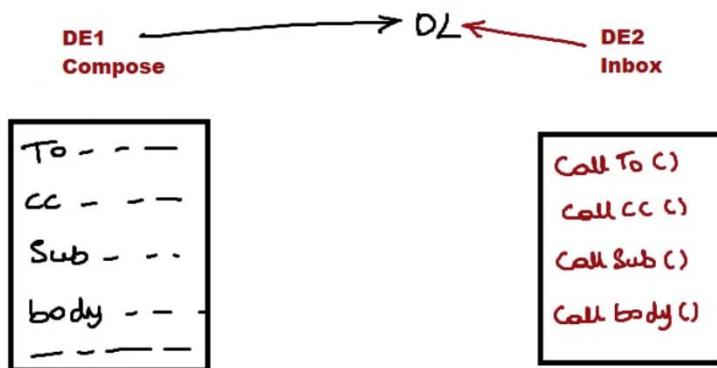


WBT from Memory Point of View:

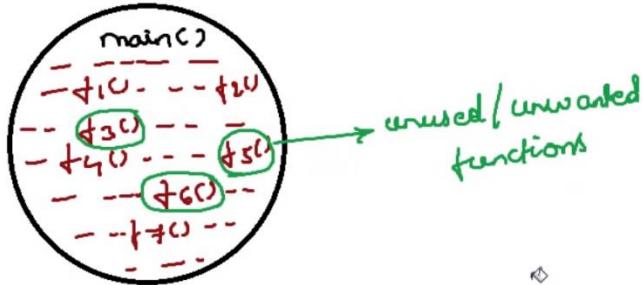
> By changing the logic of the code memory can be improved, because writing logic varies from person to person.



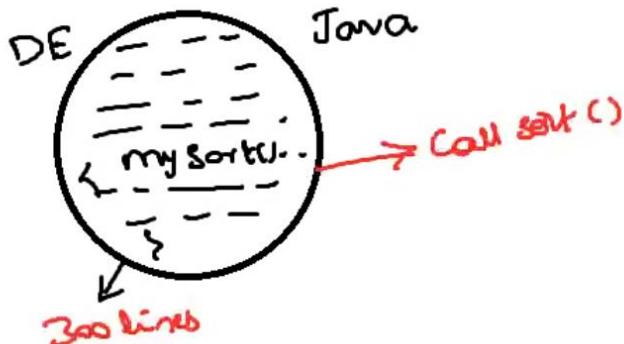
> If there is a repetition in the code or duplicate codes, we can convert that duplicate code into generic function or user defined function.



> By removing unused or unwanted functions, memory can be improved.



> By using in-built functions memory can be improved.



> By removing unused variable memory can be improved.

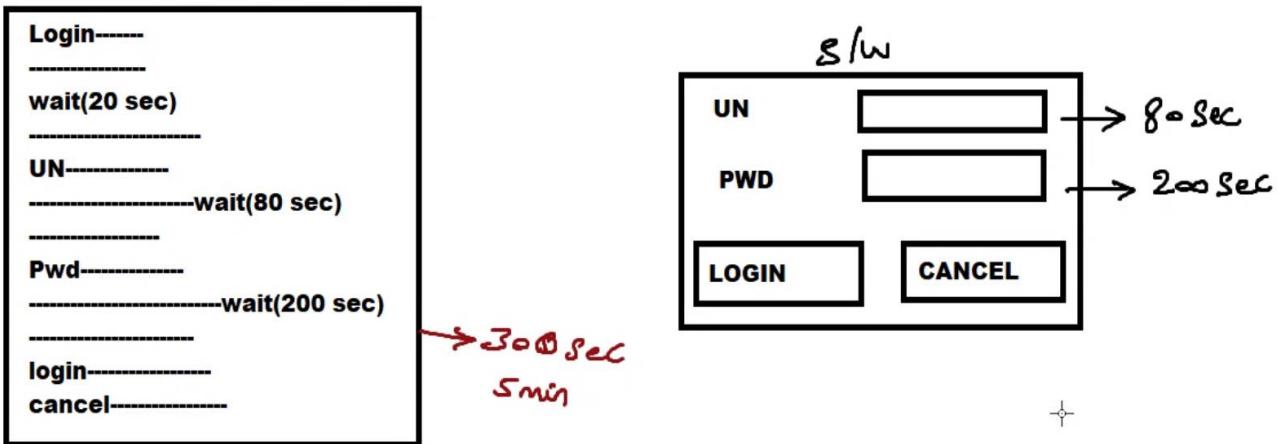
20 lakhs lines of code

```
Public static void main(String[]args)
{
int x=30; int y=40;
string "hello"
varchar()
float()
double()
{  
----  
----  
----  
----
```

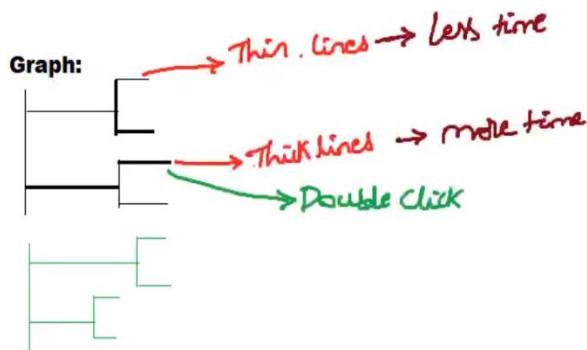
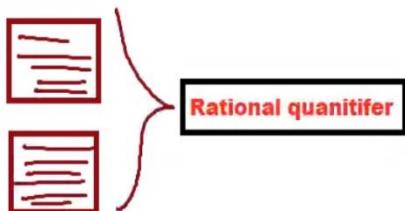
IBM
C, C++, Java
Rational purifier

WBT from Performance Point of View:

> By writing good or proper logic performance can be improved (by removing all unnecessary wait statement).



> AND & OR operators should be properly used while writing conditional cases.



What is difference between WBT & BBT ?

WBT

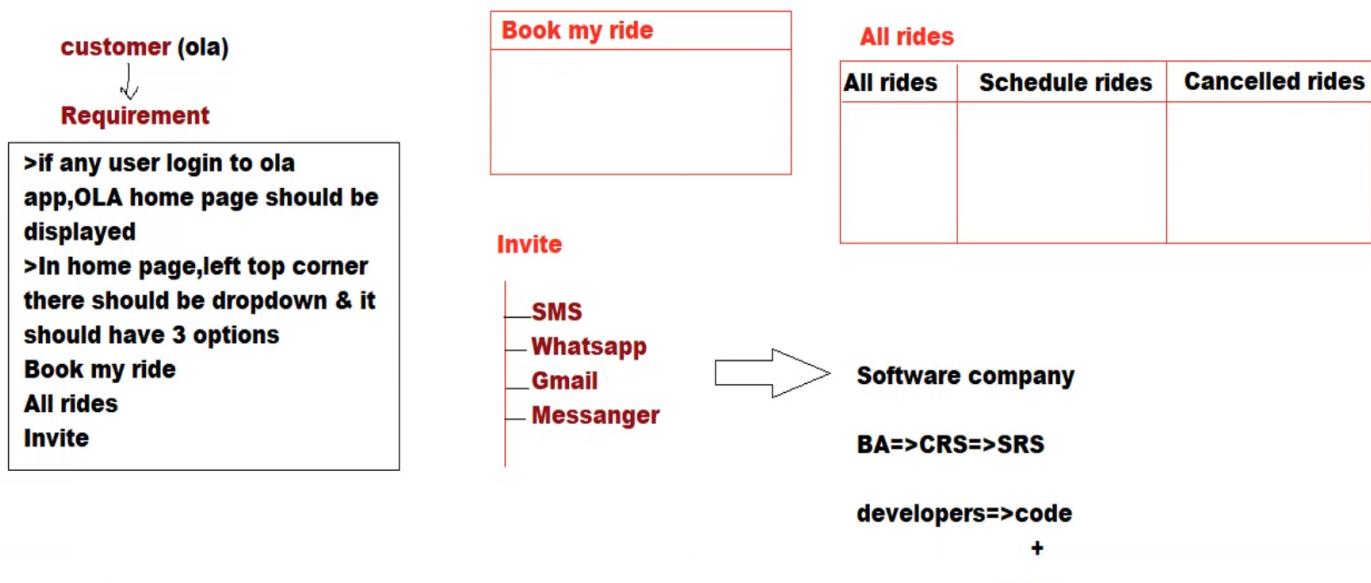
- >Developer will do WBT
- >Here code is visible
- >A person should have knowledge on program to do WBT
- >A person should have knowledge on internal design of code
- >Here they will test the logic of the code
- >WBT is done before software development
- >here developer will pass input into the source code & check the output according to requirement

BBT

- >TE will do BBT
- >Here code is not visible
- >Programming knowledge is not required
- >Knowledge on design is not required
- >Here we test the functionality of the software
- >BBT is done after software development
- >Here TE will pass input to the software & check the output according to the requirement

Black Box Testing

Verifying the functionality of an application against requirement specification is called Black Box Testing.
It is done by TE.

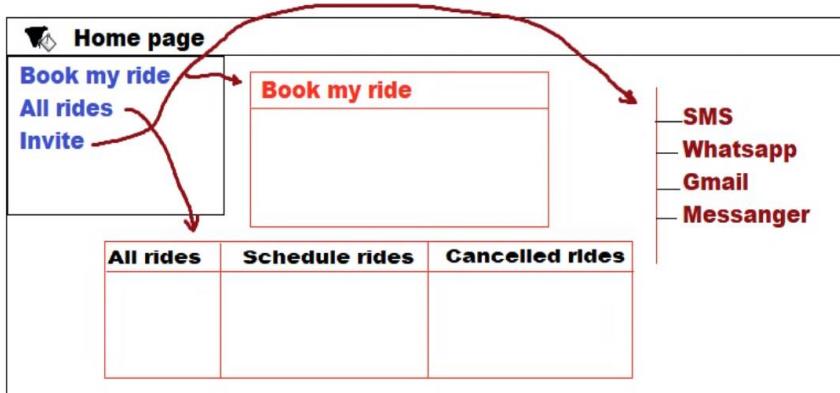


TE's =>BBT

Step 1: TE should install ola application

Step 2: TE register fo ola app by entering OTP

If any user opens OLA



Types of Black Box Testing:

- > Functionality Testing
- > Integration Testing
- > System Testing
- > Acceptance Testing
- > Usability Testing
- > Performance Testing
- > Smoke Testing
- > Adhoc Testing
- > Compatability Testing
- > Accessability Testing
- > Reliability Testing
- > Globalisation Testing
- >Regression testing

FUNCTIONALITY TESTING / FIELD TESTING/ COMPONENT TESTING

Testing each and every component thoroughly against requirement specifications is called Functionality Testing.

COMPONENT: It means text field, text area, drop-down, radio button, check box, link, normal button, scroll bar, major tab, minor tab, grid, widget, icon, etc.

NOTE: Here thoroughly means trying with all possible inputs or scenarios or cases.

EXAMPLE 1

Add user link:

- 1.when user clicked on add users link , add user page should be display.
- 2.user name text field:it should accept 5-32 char
- 3.pwd text field:it should accept min 6 char & include atleast 1 special char
- 4.designation dropdown:it is mandatory
- 5.emailid text field:it should accept valid mail id format
- 6.mobile no:it should accept 10 digit no

- 7.gender:it is mandatory
- 8.D.O.B text field:it should accept valid d.o.b
- 9.submit button:when clicked user should be created
- 10.cancel button:it should not create user

Software

Add user	Add user page
UN	<input type="text"/>
PWD	<input type="text"/>
Designation	<input type="text" value="Lead"/> Manager TE
email id	<input type="text"/>
mobile no	<input type="text"/>
Gender <input type="radio"/> Male <input type="radio"/> Female	
D.O.B <input type="text"/> / <input type="text"/> / <input type="text"/> <input type="text"/> / <input type="text"/> / <input type="text"/>	
<input type="button" value="Submit"/> <input type="button" value="Cancel"/>	

EXAMPLE 2

Requirement

- 1.welcome page
-
- 1.1login
-
- 1.1.1 user name text field:it should accept 6-32 char
- 1.1.2 pwd text field:it should accept min 5 & atleast 1 special char
- 1.2 forgot pwd:-----
- 1.3rem pwd:-----
- 2.loans
-
- 2.1personal loan:-----
- 2.1.1
- 2.1.2
- 2.2 home loan:-----
- 2.1.2 fixed intrest:-----
- 2.1.3floating intrest:-----
- 30.Amount transfer:
- 30.1FAN text field:
- 30.1.1 It should accept only 10 digit valid no's
- 30.1.2 It should accept those no's which are created by manager
- 30.2 TAN text field
- 30.2.1-----
- 30.2.2-----
- 30.3 Amount text field
- 30.3.1 It should accept only +ve integre between 100-5000
- 30.3.2 It should not accept more than balance

SOFTWARE

Amount transfer page

Amount transfer	FAN <input type="text"/> TAN <input type="text"/> AMOUNT <input type="text"/>
<input type="button" value="TRANSFER"/> <input type="button" value="CANCEL"/>	

100	✓
-100	✗
100-50	✗
1.00	✗
Blank	✗
0	✗
100/-	✗
Hundred	✗
90	✗
5001	✗
4999	✓

Why every requirement should be numbered?

- > There will be clarity in the requirement.
- > It will be easy to understand.
- > Communication between customer development team & testing team will be smooth.

How to test the feature/component thoroughly?

- > Always start testing the application with valid data. If the field is working for valid data then only test the component for invalid data.
- > If the component is working for one of the invalid data, we can still continue the testing for some more invalid data.

Functionality Testing can be done in 3 difference ways:

> Over Testing/Exhaustive Testing/Donkey Testing:

- > Testing the feature with the same scenario in different ways or testing with the scenario which doesn't make any sense is called as Over Testing.
- > Ex: 100 dollar, 0 dollar, 100 dollar only, 100.50, 50.14, etc
- > If we do this its waste of time.

> Under Testing:

- > Testing the feature with insufficient set of scenarios is called as Under Testing.
- > Ex: 100, -100, 100 dollar, 0, @100
- > If we do this we are going to miss lots of defects.

> Optimised Testing:

- > Testing the feature with scenarios which make sense is called as Optimised Testing.

> Here we are going to catch max bugs within optimum time.

> Ex: 100 , Hundred , 1.00 , -5000 , 40@
I

> There are 2 types of Optimised Testing:

- > Positive FT: Testing each & every component thoroughly by entering valid or expected data is called Positive FT.
- > Negative FT: Testing each & every component thoroughly by entering invalid or unexpected data is called Negative FT.

Note: 1. TE should not assume/propose the requirement.

2. If TE gets any queries or doubts he should interact with developer, manager & BA & get it clarified.

How to write Test Cases & Test scenarios for FT?

Naukri registration

component name: Registration now link

Test scenario: To check that when user clicks on registration now link tell us about yourself page should be displayed.

Test case:

Component name	Input	Expected result
Registration now link	NA	Tell us about yourself page should be displayed with following components Im a fresher button Im a professional button
Im fresher button	NA	personal page should be displayed with following components 1.Name text field should be displayed & it is mandatory 2.email id text field should be displayed & it is mandatory 3.create password text field should be displayed & it is mandatory 4.Mobile no text field should be displayed & it is mandatory 5.Current city dropdown should be displayed & it is mandatory 6.outside india check box should be displayed 7.upload resume button should be displayed 8.copy paste resume here link should be displayed 9.terms & condition check box & link should be displayed 10Registration now button should be displayed

Test scenarios

Component name:

1.Name text field:

1. To check that name text field should accept alphabets.
2. To check that name text field should accept alphabets with space
3. To check that name text field should accept alphabets with single quotes
4. To check that name text field should accept blank space
5. To check that name text field should not accept more than 35 characters
6. To check that name text field should not accept alphabet with special characters other than space, dot, single quote
7. To check that name text field should accept alphabet with special character
8. To check that name text field should not accept the combination of alphabet & numbers
9. To check that name text field should not accept any numbers
10. To check that name text field should not accept only special characters other than space, dot, single quote
11. To check that name text field should accept less than 34 characters
12. To check that name text field should accept only upper case character
13. To check that name text field should accept only lower case character
14. To check that name text field should accept both upper and lower character
15. To check that name text field when user enter invalid data it should break proper error message

Password text field

Current city drop down

1. To check that current city dropdown should accept all the cities which are present in dropdown.
2. To check that current city dropdown user should be able to scroll the cities by using scroll bar
3. To check that current city dropdown when user points the cursor on the city it should get highlighted
4. To check that current city dropdown should accept when user should be able to select only one city at a time
5. To check that current city dropdown should not accept blank space

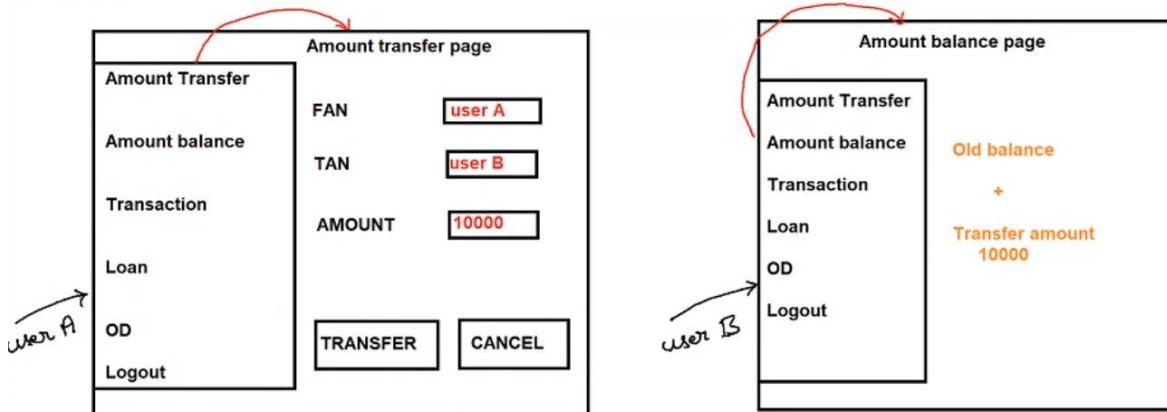
Integration Testing

Testing the dataflow between the modules is called as Integration Testing.



If module 'A' is able to send data & module 'B' is able to receive data, then Integration Testing between 'A' & 'B' is pass.

Real time example



Scenario:

1. Login as user A click on AT enter Account number as user A To Account number as user B Amount as 10000 Click on transfer button, Confirmation message will be displayed logout as user A.
2. Login as user B click on Amount balance proper balance should be displayed old balance + transferred amount & logout as user B.

How to do Integration Testing?

- > Understanding the application is very important, i.e.,
 - > Understand how each & every module works.
 - > Understand how each & every module is related.
- > Identify the scenarios.
- > Prioritize the identified scenarios.
- > Document the scenarios.
- > Execute the scenarios.
- > While executing if you get any defects then communicate defect to developer.

Positive IT:

> login as user A click on amount balance(10000)click on amount transfer, enter from a/c number as user A , To a/c no as user B amount as 10000 click on transfer confirmation message will be displayed,logout as user A.

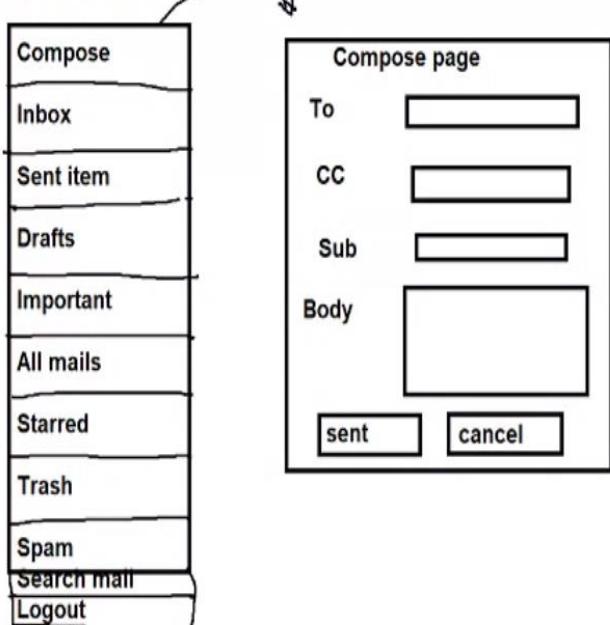
> login as user B click on amount balance proper balance + 10000 is displayed then +ve integration testing is pass suppose if some other amount is displayed then +ve integration testing is fail.

Negative IT:

> login as user A click on amount balance(10000)click on amount transfer, enter from a/c number as user A , To a/c no as user B amount as 15000 click on transfer confirmation message will be displayed,logout as user A.

> login as user B click on amount balance proper balance should be displayed that it is should display only old balance then -ve integration testing is pass suppose if some other amount is displayed then -ve integration testing is fail.

Example 2



Scenarios:

1.Login as user A click on compose enter values for all the fields and send mail to user B logout as user A.
Login as user B click on inbox check whether mail is displayed.

2.Login as user A click on inbox select a mail & delete a mail later click on trash check whether mail is displayed.



Test scenario:

login user A click on inbox select a mail & move to spam

Click on spam check wether mail is displayed.

Test case:

Step no	Action/Description	Input	Expected result
1	open the browser & enter URL	www.gmail.com	login page should be displayed
2	enter valid UN & PWD click on login button	UN:ABC PWD:XYZ	home page should be displayed
3	click on inbox	NA	Inbox page should be displayed
4	select a mail & move to spam	NA	mail should be moved to spam
5	Click on spam	NA	spam page & mail should be displayed
6	click on logout	NA	user should be logged out from the application sucessfully

What is difference between test case & test scenario?

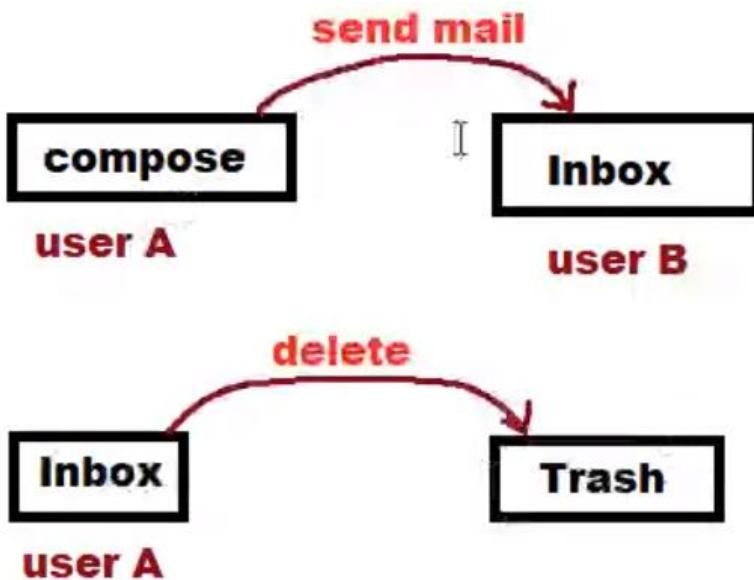
Test Scenario

- > It is a high level document of all the customer business workflow, according to requirement.
- > We can write test scenario by looking into requirement
- > By looking into scenario we can't test any application until we have very good product knowledge.
- > It is a 1 line or few lines description about the feature

Test Case

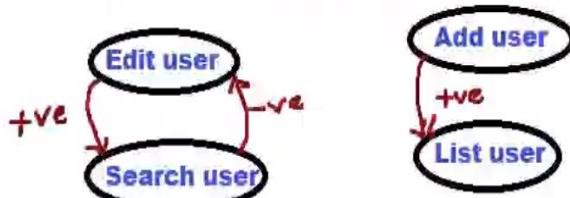
- > It is a detailed document of the scenario which helps TE to test the application.
- > We can write test cases by looking into both requirement & scenario.
- > We can test any application no matter whether we have very good product knowledge or not.
- > It is a detailed description about the feature.

INTEGRATION TEST SCENARIO BOX FORMAT

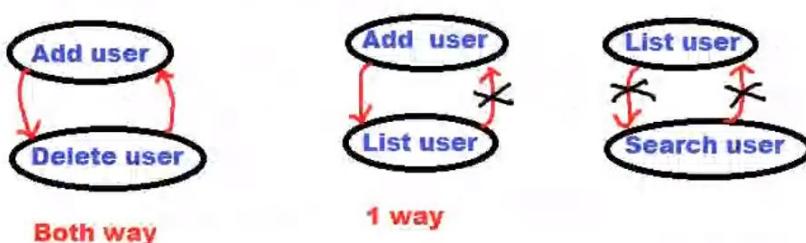


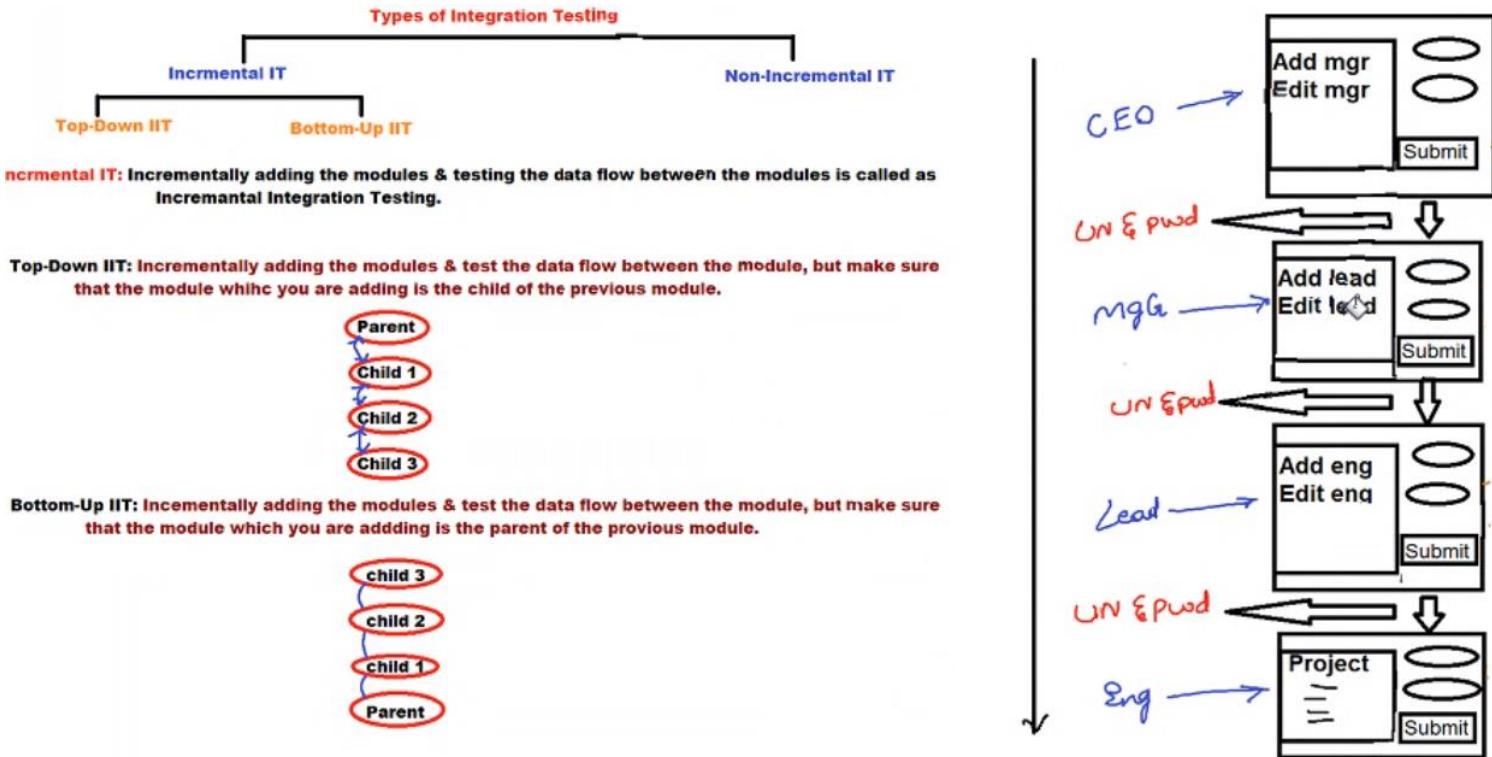
Note:

1. Prioritization is very important & we have to do it in all the stages, i.e.,
 - > Open the application & select which feature to be tested first & later.
 - > Go to first feature & select which component to be tested first & later.
 - > Go to first component & decide what value to be entered first & later.
2. Between 2 features we might do both positive IT & negative IT or we might do only positive IT, it just depends on how the feature is related.



3. Don't apply same logic everywhere, because testing logic varies from feature to feature.
4. Focusing on features is very important, i.e., take one feature completely test that feature & then only you move on to next feature.
It means take 1 feature, do thorough FT on that feature, for the same feature do thorough IT, then only move on to the next feature.
5. Between 2 features we might do IT, both ways or 1 way or we may not do IT at all, it just depends on how the features are related to each other.

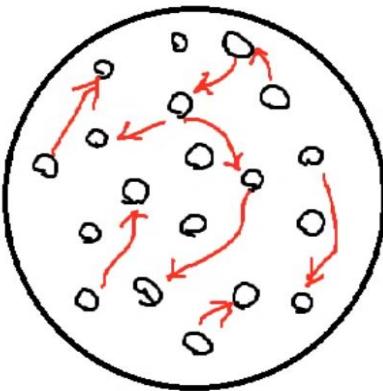




Note: When we don't know or when it is difficult to identify parent & child module we go for Non-Incremental IT.
When data flow is very complex, don't spend time in identifying parent & child module, go for Non-Incremental IT.

NON-INCREMENTAL INTEGRATION TESTING/ BIG BANG METHOD

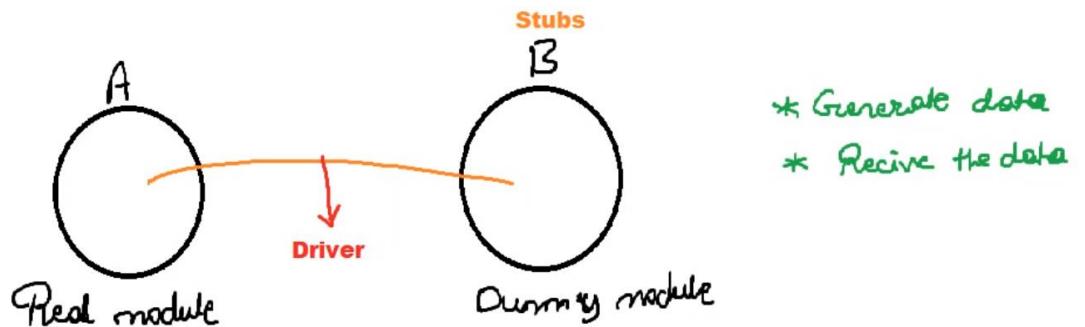
Here we combine all the modules in one shot and test the data flow between the modules.



> Drawbacks of Non-Incremental IT:

- > Chances are there we might miss some dataflow.
- > Chances are there we might test same thing again & again, because of this time taken will be more.
- > It is difficult to identify the root cause of the defect.

Assume that there are 2 modules, A&B. Module A is ready & module B is not ready, how will you do IT?



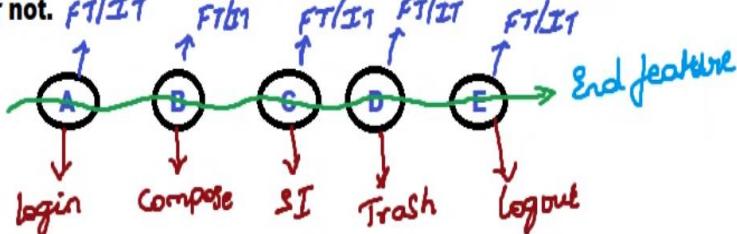
- * Generate data
- * Receives data
- * Analyse data
- * Store data

- > Yes we can do IT, by introducing 'Stubs' & 'Driver'.
- > Stubs: It is a dummy module which acts like a real module, whereas it will generate data & receive data.
- > Driver: It sets up the environment between real module & dummy module & does the transaction between them. It will analyse the result & send the report.

System Testing

It is end to end testing wherein test environment is similar to production environment.

- > **End to End Testing:** Navigating through all the features & check whether end feature or last feature is working as expected or not.



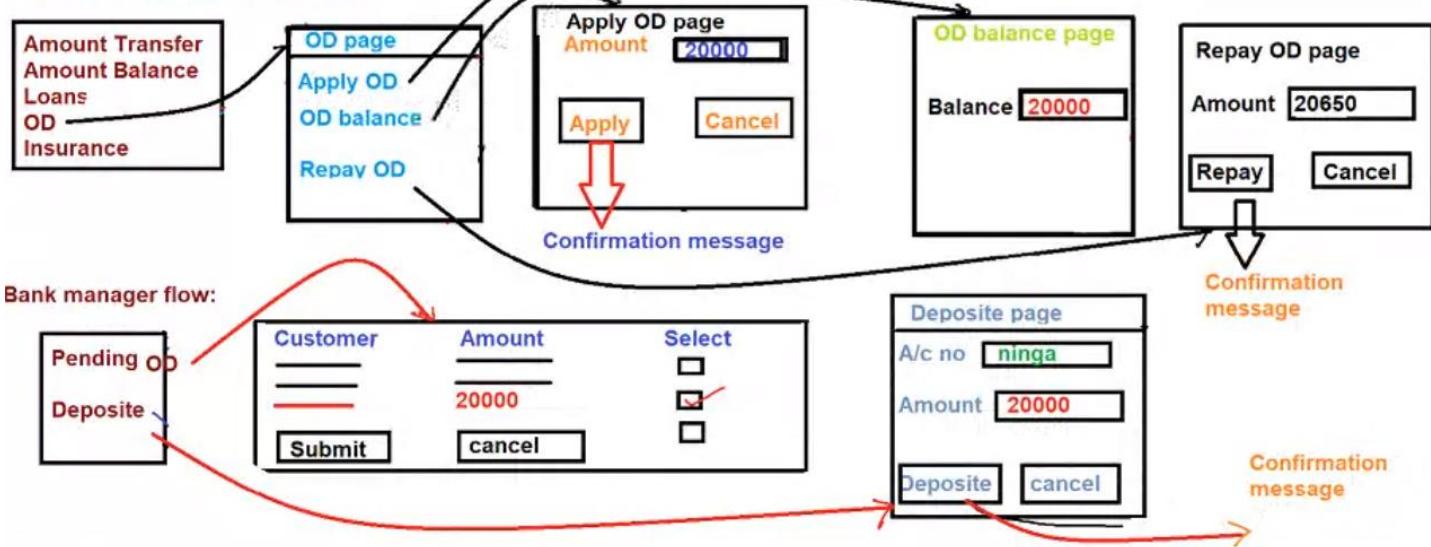
- > While doing end to end testing we will not worry about FT & IT because before we start with end to end testing already we will be done with FT & IT.

Overdraft: It is a type of loan given by the bank to the customer based on monthly salary & usually it will be approved within 2 days by Bank Manager.

Requirement:

- > If any new customer applies OD for Rs.20000 for the 1st time, then after 1 month bank will charge 2% ROI & also Rs.250 as activation fees.(20650)
- > If same customer applies OD for Rs.20000 for 2nd time, then bank will charge only 2% ROI.(20400)

Customer business work flow:



Login as user click on OD , OD page should be display,click on apply OD,enter amount as (20000) click on apply button,confirmation meesage ,Login as manager, click on pending OD,select the user and approve,confirmation meesgae, click on deposite enter user a/c number & enter the amount(20000),click on deposite ,confirmation meesage will display,Login as user,click OD balance(20000),change date in server for next 1 month,click on Repay OD , amount as 20650,click on repay button,confirmation message,check repay Balance will be 00000.

Test case:

step1: Login as user 'A' click on OD, Click on apply OD, enter amount Rs.20000 & click on apply button ,Confirmation message should be displayed.

step2: Login as manager click on pending OD select A's OD & approve confirmation message should be displayed.

step3: Login as manager click on deposite,deposite page should be displayed & enter A's a/c no,enter amount 20000 click on deposite button confirmation message should be displayed.

step4: Login as user 'A' click on OD balance page ,balance page should be display & balance amount should be dispalyed.

step5: Change the server date for next 1 month.

step6: Login as user 'A' click on OD,click on balance,OD balance page will display with balance amount.

step7: Login as user 'A' click OD ,click on repay OD (20650),repay OD page should be display click on repay , confirmation message should be displayed.

			Expected Result
—	—	— NA	—

Note:

- 1)Take any application start from login navigation through all the features & logout make it as one end - end scenarios
- 2)Take any application combine 6-7 features & make it as one end - end scenarios
- 3)Club any 3-4 integration scenarios & make it as one end - end scenarios

Example: Facebook

- 1)Signup for fb
- 2>Login to fb
- 3>Upload profile pic
- 4>Tag the photo
- 5>check some notification
- 6>Accept friend request
- 7>Replay for the messages
- 8>Send a friend request
- 9>Post a comment
- 10>Check your friend photo
- 11>Logout

Whatsapp

- 1)Install whatsapp
- 2>Register to it
- 3>Open whatsapp
- 4>Upload profile pic
- 5>Upload status
- 6>click on chats
- 7>Create a group
- 8>Replay for status
- 9>Add friend group
- 10>Click on call
- 11>Make video & voice call
- 12>logout

>Development Environment/Server:

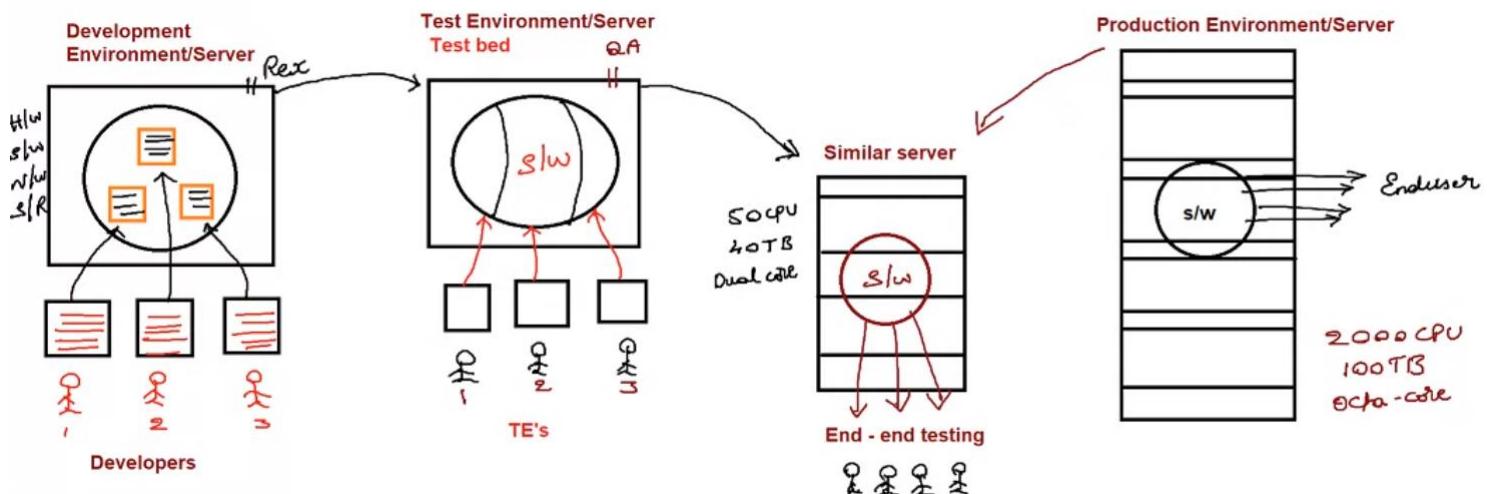
- > It is a setup which is used to develop the s/w.
- > It contains hardware, software, network, server.

> Test Environment/Server:

- > It is a setup which is used to test the s/w.
- > It contains hardware, software, network, server.

> Production Environment/Server:

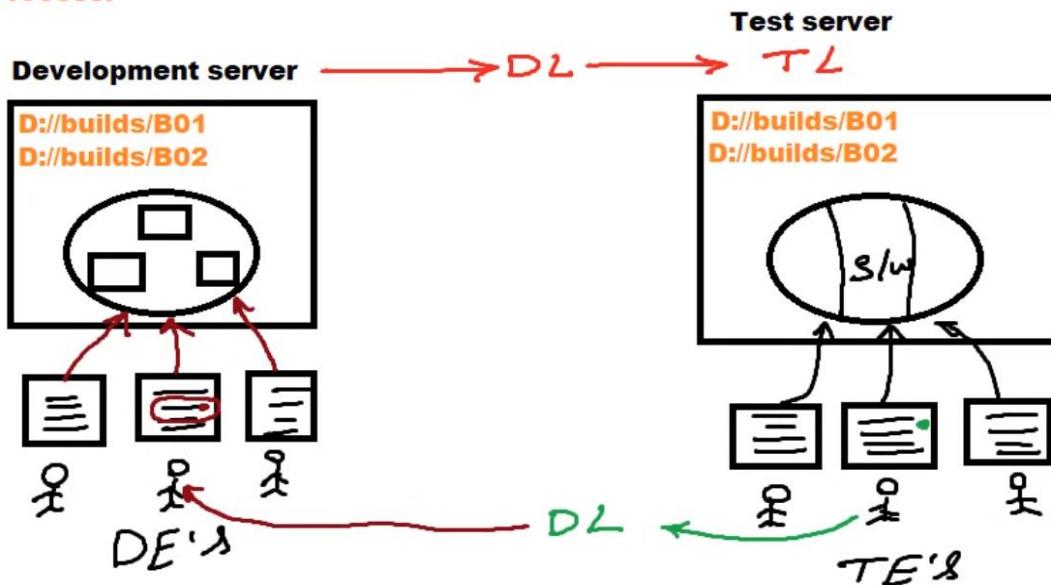
- > It is a setup which is used to run the s/w for business.
- > It contains hardware, software, network, server.



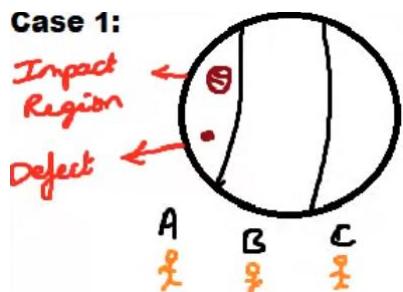
How they develope,test & release the product to the customer in real time project?

or

Build & Release Process.

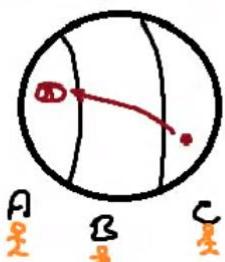


Case 1:



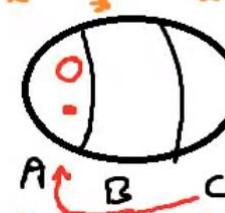
TE 'A' retest the defect & also test the impacted region

Case 2:



TE 'C' retest the defect & TE 'A' test the impacted region
(owner of the bug is to retest the defect
owner of the region should test the impacted region)

Case 3:



TE 'C' will retest the defect
TE 'A' will test the impacted region

Note: Whenever we get new build always

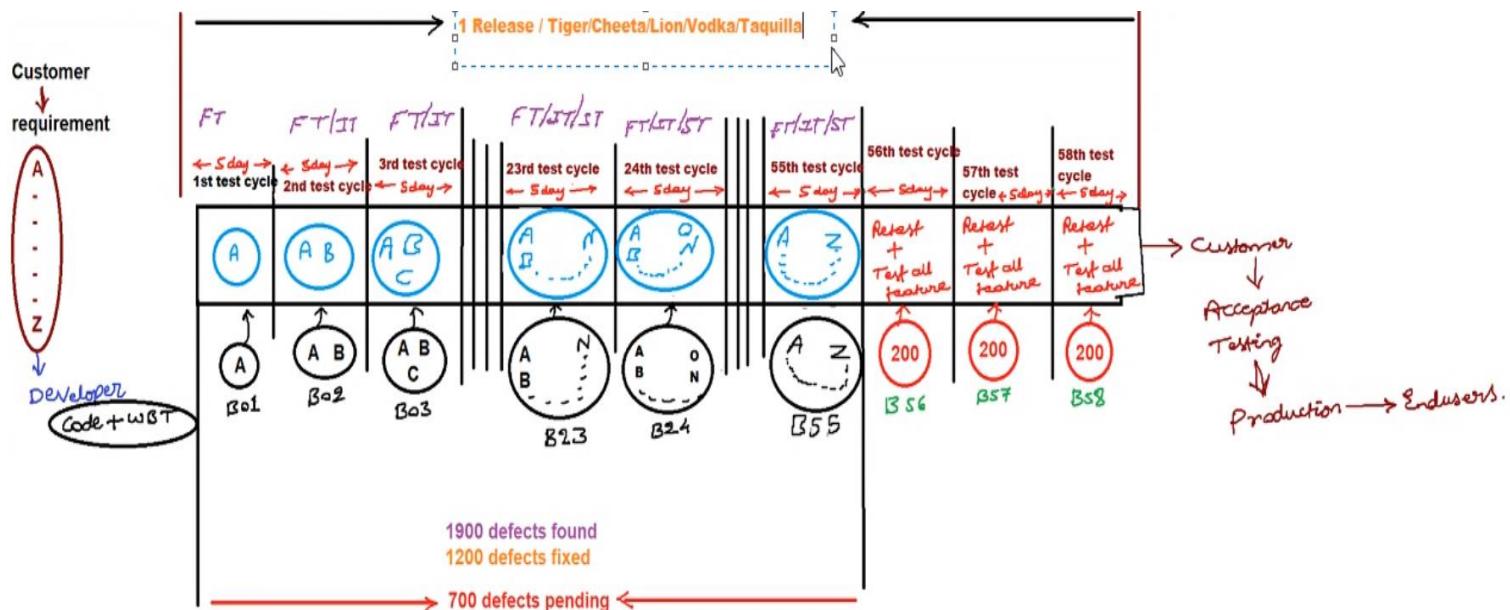
1. We should test the new module first because probability of finding defects will be more in new module.
2. Test the integration between new module & old module.
3. Retest the fixed defects in old module.
4. Test the old module.

Why do we get new defects in old module?

- > Chances are there TE would have missed in the previous cycle & catching in current cycle.
- > Chances are there fixing one defect might have impact on other defects.
- > Adding new feature may introduce defect in old feature.

When we do System Testing?

- > When all the basic features are functionally stable.
- > Minimum bunch of features should be ready.
- > Environment similar to production environment must be available.



What is Release?

- > Starting from gathering the requirement, developing the product & testing it for so many cycles, until we deploy the product to production is called as Release.
- > Suppose if company is following 'Agile Model' then the same release is called as Sprint.

What is Test Cycle?

- > It is the effort spent or time spent by TE to start & finish the testing.
- > 1 test cycle might take 3d, 5d, 10d, 15d. It varies from project to project & company to company.
- > It also depends on
 - > Size of application
 - > No. of engineers
 - > Complexity of application

How many test cycles you took in your previous project(for exp candidates)?

5 days

project duration => 10 months

1 month => 4 cycles

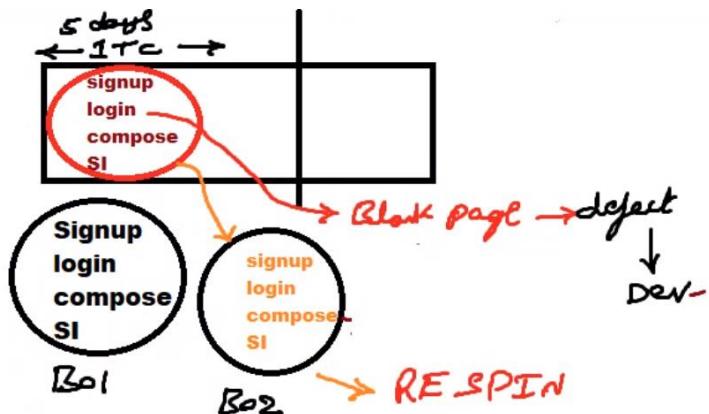
10 month => 40 cycles (+/- 3)

3 days => 1 cycle

21 days => 7 cycle

10 month => 100 cycle (+/- 3)

What is Re-Spin?



- > The process of getting new build within 1 test cycle is called Re-Spin.
- > The process of getting new build within re-spin is called as Re-Spin 1.

What is Build?

- > When you compile all the programs we get Binaries & when we compress all the binaries we will get Build & we will get build in the below mentioned formats:
- > .exe, .jar(java archive), .war(web application archive), .tar(tape archive), .apk(android), .ipa(iOS)

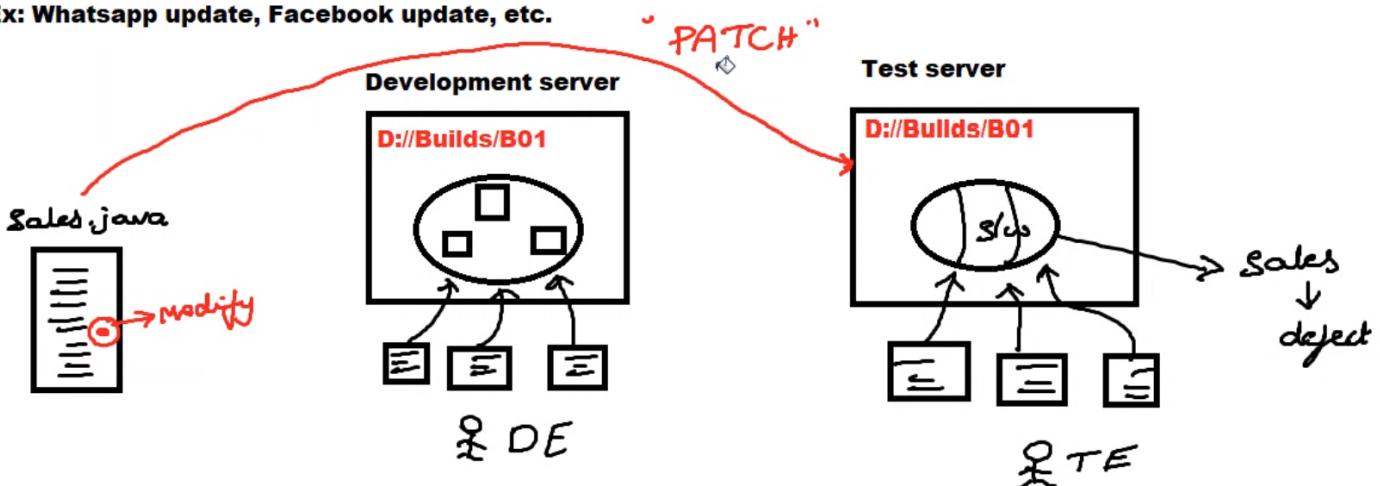
Note: 1 project can have 'n' number of releases.

In 1 release we can get 'n' number of builds.

Also 1 release can have 'n' number of test cycles.

Patch:

- > It is a small s/w which contains new program, modified program & deleted records.
- > When we install patch it goes & replaces with old s/w.
- > Ex: Whatsapp update, Facebook update, etc.

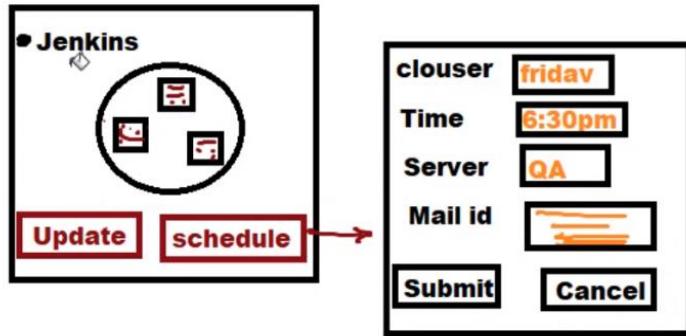


Who will be involved in installation of the product?

- > Anybody from the Development Team(Dev Manager, Dev Lead, Sr.Dev, Jr.Dev, Fresher).
- > Anybody from the Testing Team(Test Manager, Test Lead, Sr.TE, Jr.TE, Fresher).
- > Build Engineer or Release Engineer.

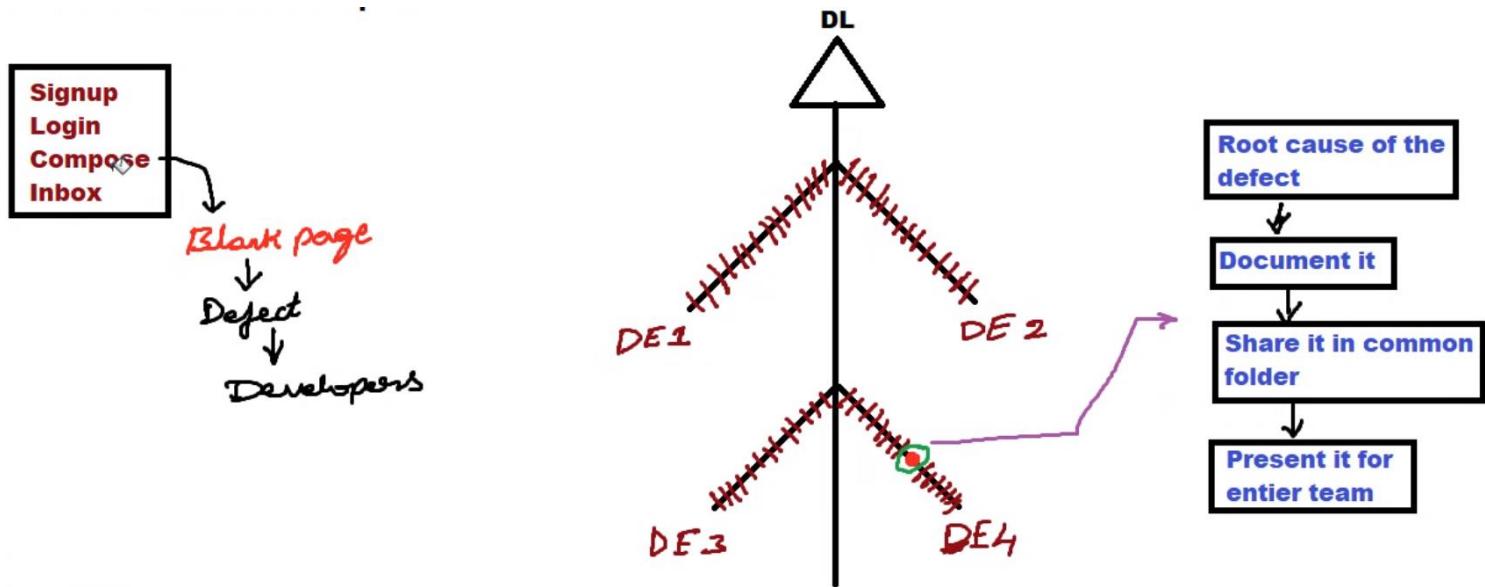
Continuous Integration:

Development Environment getting continuously integrated with Test Environment is called Continuous Integration.



Fish Bone Technique / Ishikawa Method / Root Cause Analysis Meeting (RCA):

When TE finds a defect they will communicate defect to developers, all developers will sit together find root cause of the defect, document it, share it common folder & present it to the entire team. This technique is called as Fish Bone Technique.



Accessibility Testing / ADA(American Disability Act) Testing / 508 Testing:

Testing the user friendliness of the software from physically challenged person point of view is called as Accessibility Testig.

Ex: 1. Here we check whether Red & Green coloured objects are present or not.

2. Here we check whether just by using keyboard whether we are able to access the software or not.

Reliability Testing:

Testing the functinality of an application continuously for a particular period of time is called as Reliability Testing.

Recovery testing:

Testing the application to check wether how soon application recovered from the crash.

Exploratory testing:

Explore the application,Understand the application based on understanding identify all possible scenarios & documanet the scenarios(FT,IT,ST)& test the application based on documented scenarios.This is called as exploratory testing

When/ Why we do exploratory testing?

1)When there is no requirement

2)When req is present & not able to understand

3)When req is present & able to understand but there is no time to read

Comparission testing(Parallel testing):

Comparing the newly built application with similar kind of application which is already released in the market is called as comparission testing.

Here we check for strenght & weakness(Advantages & disadvantages)& we make sure that all the features are present in newly built application.

Ex:Amazon & flipkart

Ola & Uber

Zomato & swiggy

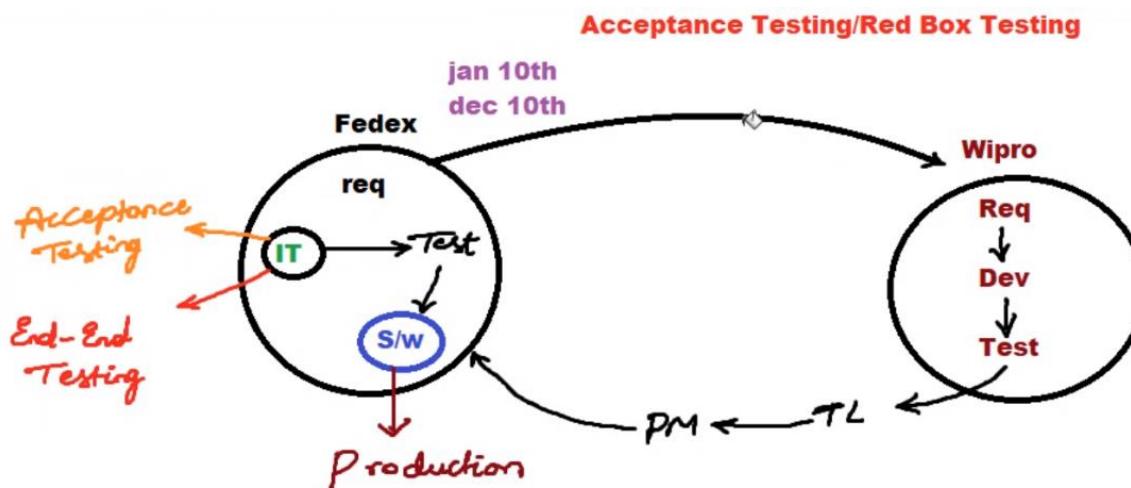
Phone pay & google pay

facebook & instagram

Yellow box testing:

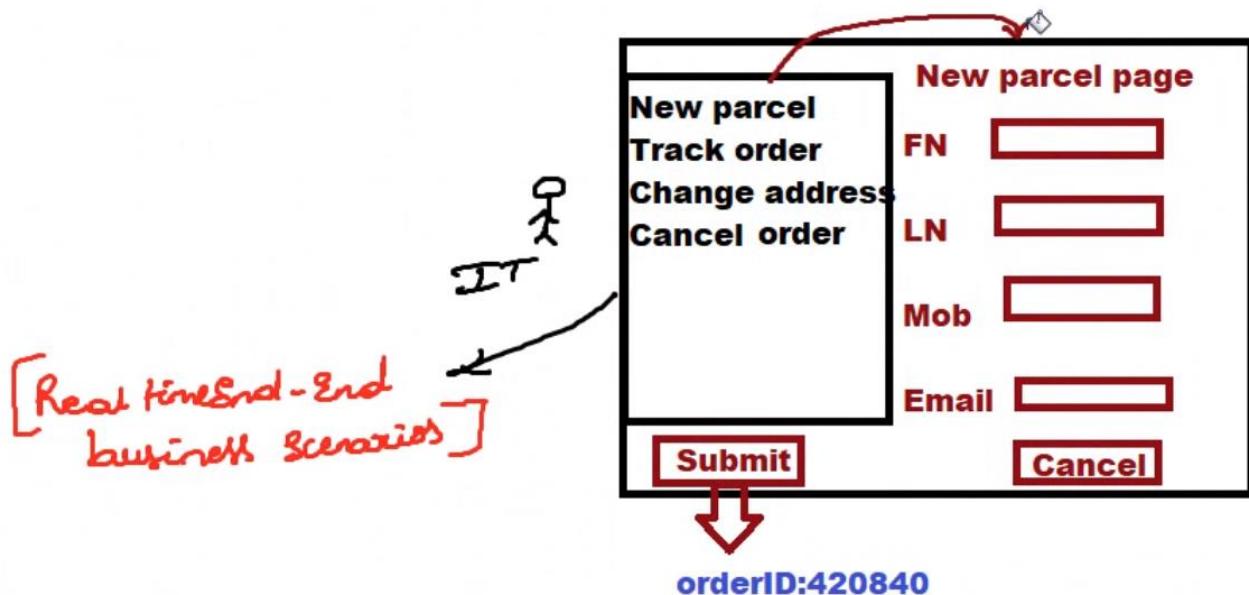
Testing the warning messages of the application is called as yellow box testing.

It is a subset of usability testing.



What is Acceptance Testing?(Approach 1)

It is the end to end testing done by IT Engineers sitting in customer place wherein they consider end to end real time business scenarios & check whether the s/w is capable of handing it or not.

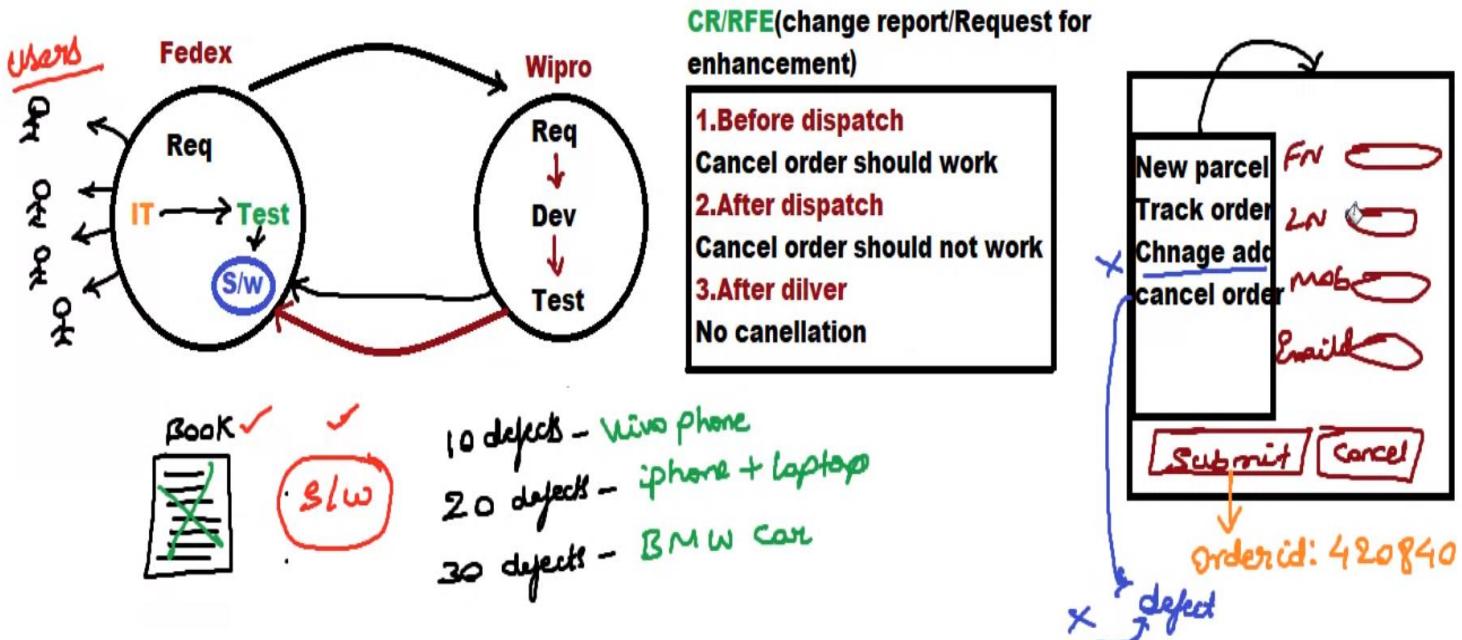


Why we do Acceptance Testing?

- > Under business pressure software company might push the s/w with a lot of defects to avoid that customer will do Acceptance Testing.
- > Chances are there developer might misunderstand the requirement given by the customer & develop a wrong feature, to find that customer will do Acceptance Testing.
- > If the customer move product for production with lot of defects they will undergo severe loss, so to avoid that customer does Acceptance Testing.

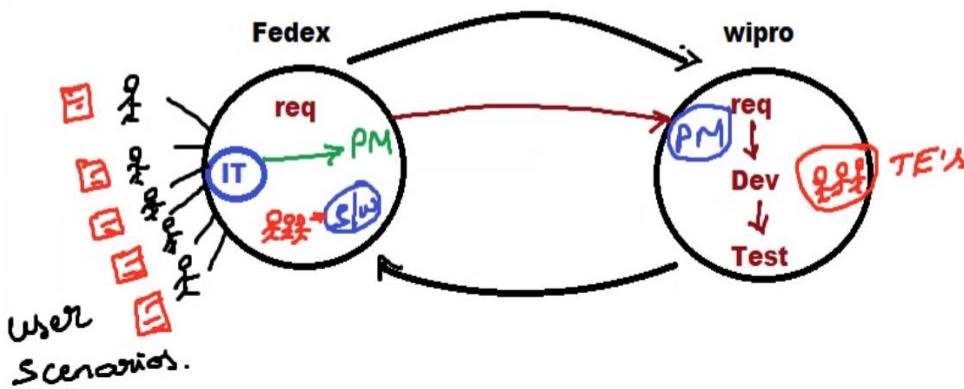
Approach 2:(User Acceptance Testing)

It is the end to end testing wherein they use the s/w for business for a particular period of time & they check whether s/w is capable of handling all real time business scenarios.



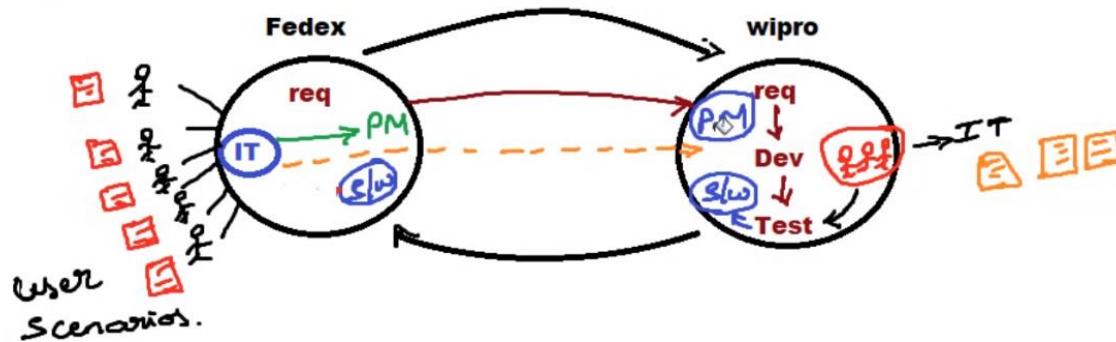
Approach 3:

It is the end to end testing done by our own engineers sitting in customer place wherein they refer user scenarios given by customer & check whether s/w is capable of handling it or not.



Approach 4:

It is end to end testing done by our own engineers sitting in our own place wherein they refer user scenarios given by customer & check whether s/w is capable of handling it or not.



What is the difference between Service based company & Product based company?

Service based company

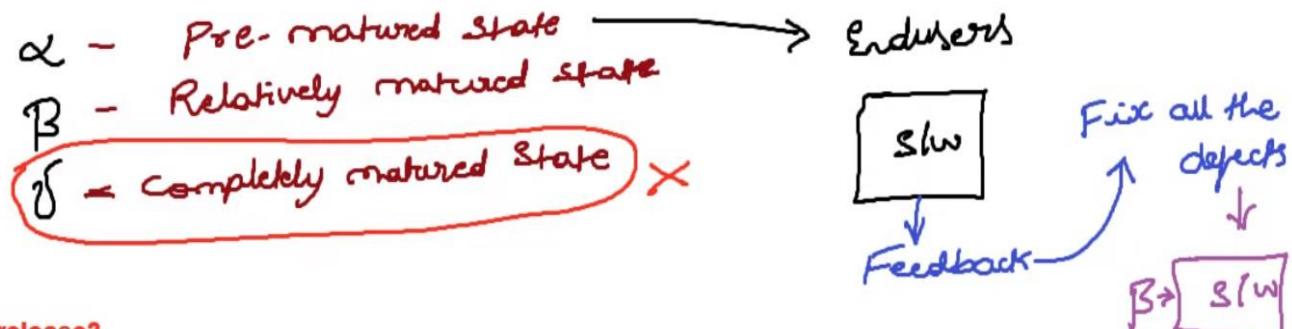
- > Ex: Wipro, TCS, Accenture, Infosys, etc.
- > Customer will give the requirement.
- > BA will be present in the service based company & he will do requirement collection.
- > Company will develop & test the s/w & give it back to same customer who gave requirement.
- > Company doesn't have any rights to sell same s/w to other customer.
- > Company doesn't have any rights to keep source code with them.

Product Based Company:

- > Ex: Google, Samsung, Sony, Nokia, IBM, Philips, Amazon, etc.
- > Initially customer will not be present.
- > Product analyst will do market research & he will collect requirement.
- > Company will develop & test the s/w & sell same s/w to multiple customer.
- > Company has all rights to keep source code with them.

What is Alpha & Beta Testing?

- > **Alpha Testing:** TE will do alpha testing before they would move product for AT.
- > **Beta Testing:** End user will do beta testing, based on their feedback, product will be released.



Why we do Beta release?

- > They will get thousands of free users to test the application.
- > Users will use it in variety of platforms & variety of different ways & catch defects which TE cannot easily simulate within the testing team.

Usability Testing/Yellow Box Testing/GUI(Graphical User Interface)/Cosmetic Testing

Testing the user-friendliness of the application or s/w is called as Usability Testing.

Whatsapp

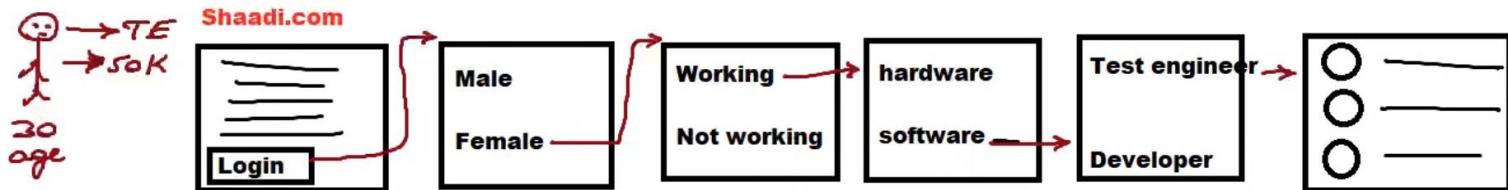
Chats
Good

Instagram

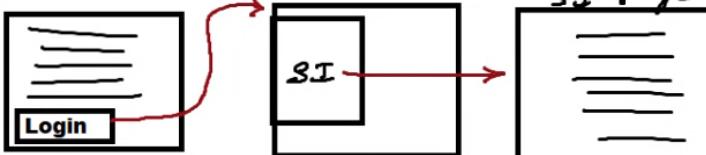
Chats
Bad

How to do Usability Testing?

- > I will check whether look & feel of the application is good or not.
- > I will check whether application is simple to understand or not, i.e., it will check if it takes less time to understand or not.
- > Important feature or frequently used feature must be given to user within 3 clicks.



Gmail



> Important feature or frequently used feature must be easily accessible.

Ex: Frequently used feature should be present either at left navigation bar or top navigation bar.

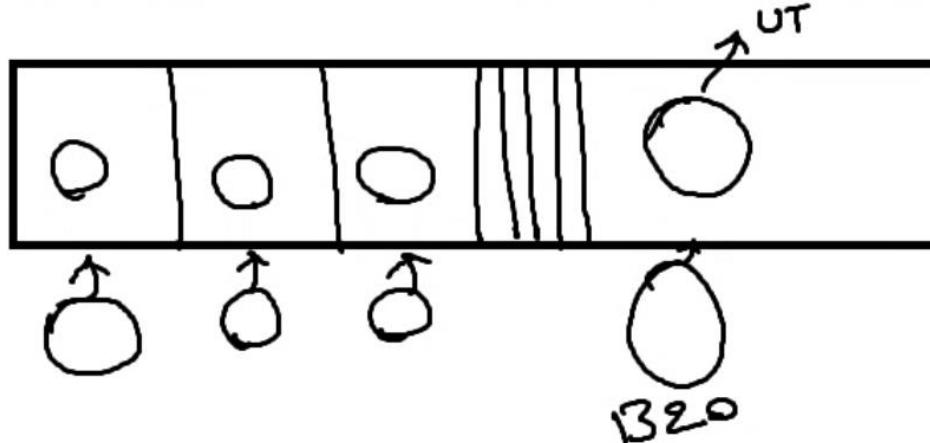
> To do any simple activity in the application it should take less number of actions.

For what kind of application we do Usability Testing?

- > Those applications which are used by multiple users or 'n' number of users, we need to do Usability Testing.
- > Those applications which generate lot of revenue we have to do Usability Testing.
- > Those applications which are used by multiple users without providing training to endusers how to use the application, i.e., we can expect themselves to learn about the application.

When we do Usability Testing?

- > Different applications we do Usability Testing in different ways,
 - > Whenever a product is functionally stable, then we go for UT.



- > In certain application we do Usability Testing in the beginning of SDLC itself.

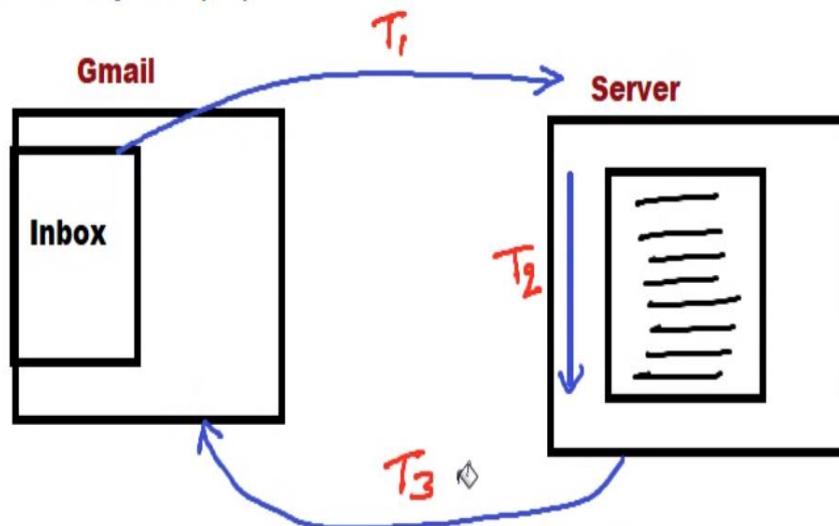
Performance Testing/Base Line Testing/Spike Line Testing/Bottle Neck Testing

Testing the stability & response time of an application by applying load is called as Performance Testing.

Stability: It is the ability to withstand load.

Load: It is nothing but the number of users.

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

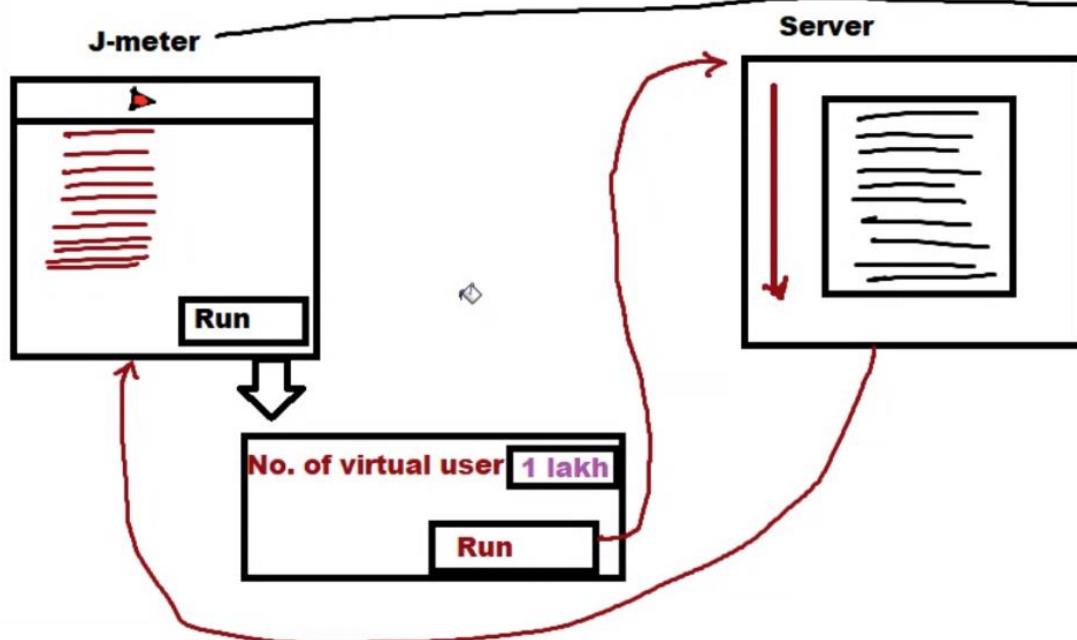


Performance Testing Tools:

1. J-meter
2. Neo-load
3. Load-runner

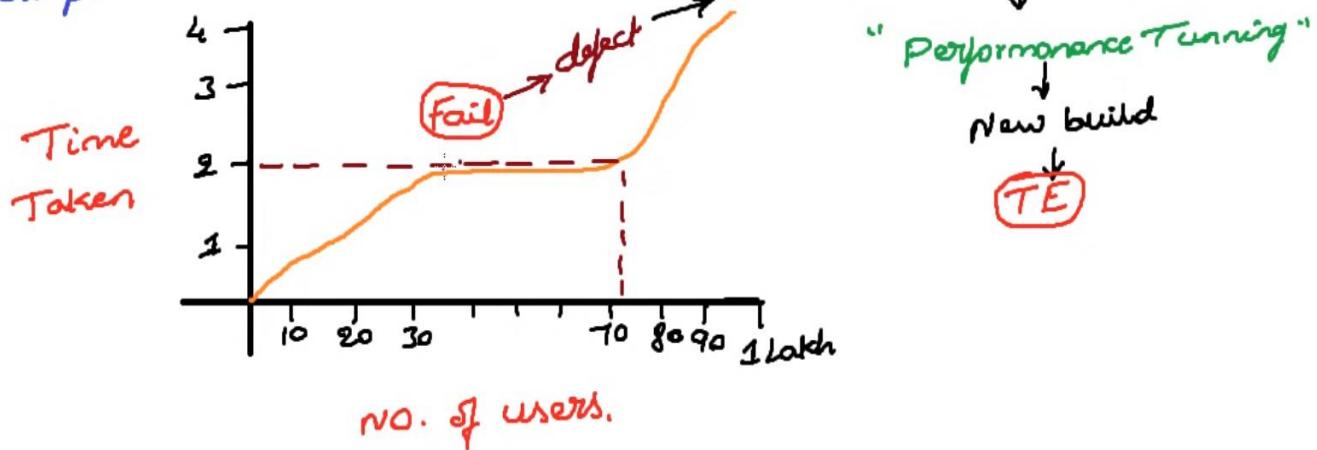
How to do Performance Testing using performance testing tool?

Result:
Graph

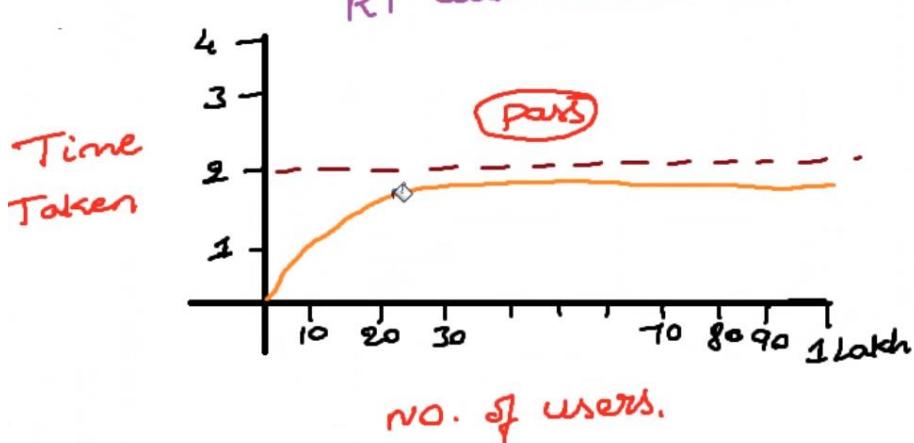


Result:
Graph

RT within 2 Sec 70k users



RT within 2 Sec upto 1L users



- > Take any performance testing tool (J-meter) & write test scripts.
- > Click on run, tool will ask to enter the number of users.
- > Those many requests will be fired at the server, now there is heavy load on the server.
- > Server sends the response to the performance testing tool (J-meter).
- > Tool will analyse the response & give the result in the form of graph.
- > TE should analyse the result & decide whether test is pass or fail.
- > If it is fail, then it is a defect, communicate to developer, developer will alter the code to improve the performance, i.e., nothing but they do 'performance tuning' & give new build to TE.
- > Once again TE should run the script & repeat all the steps.

Types of Performance Testing:

> Load Testing:

- > Testing the stability & response time of an application by applying load less than or equal to designed number of users is called as Load Testing.

> Stress Testing:

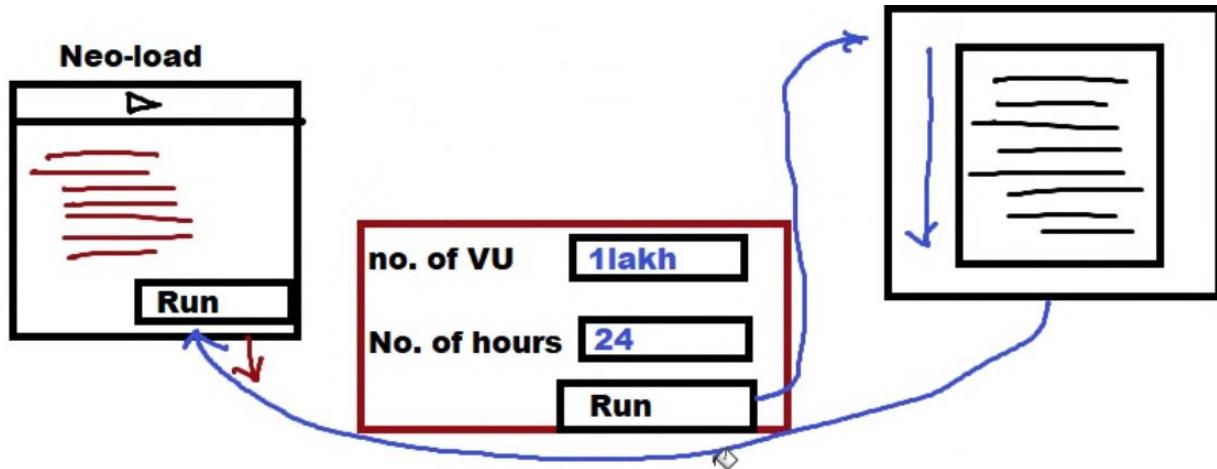
- > Testing the stability & response time of an application by applying load more than the designed number of users is called as Stress Testing.

> Volume Testing:

- > Testing the stability & response time of an application by transferring huge volume of data is called as Volume testing.

> Soak Testing:

- > Testing the stability & response time of an application by applying load continuously for a particular period of time is called as Soak Testing.



For what kind of applications we do Performance Testing?

- > Any application which is used by multiple users.
- > Any application which generates lot of revenue.

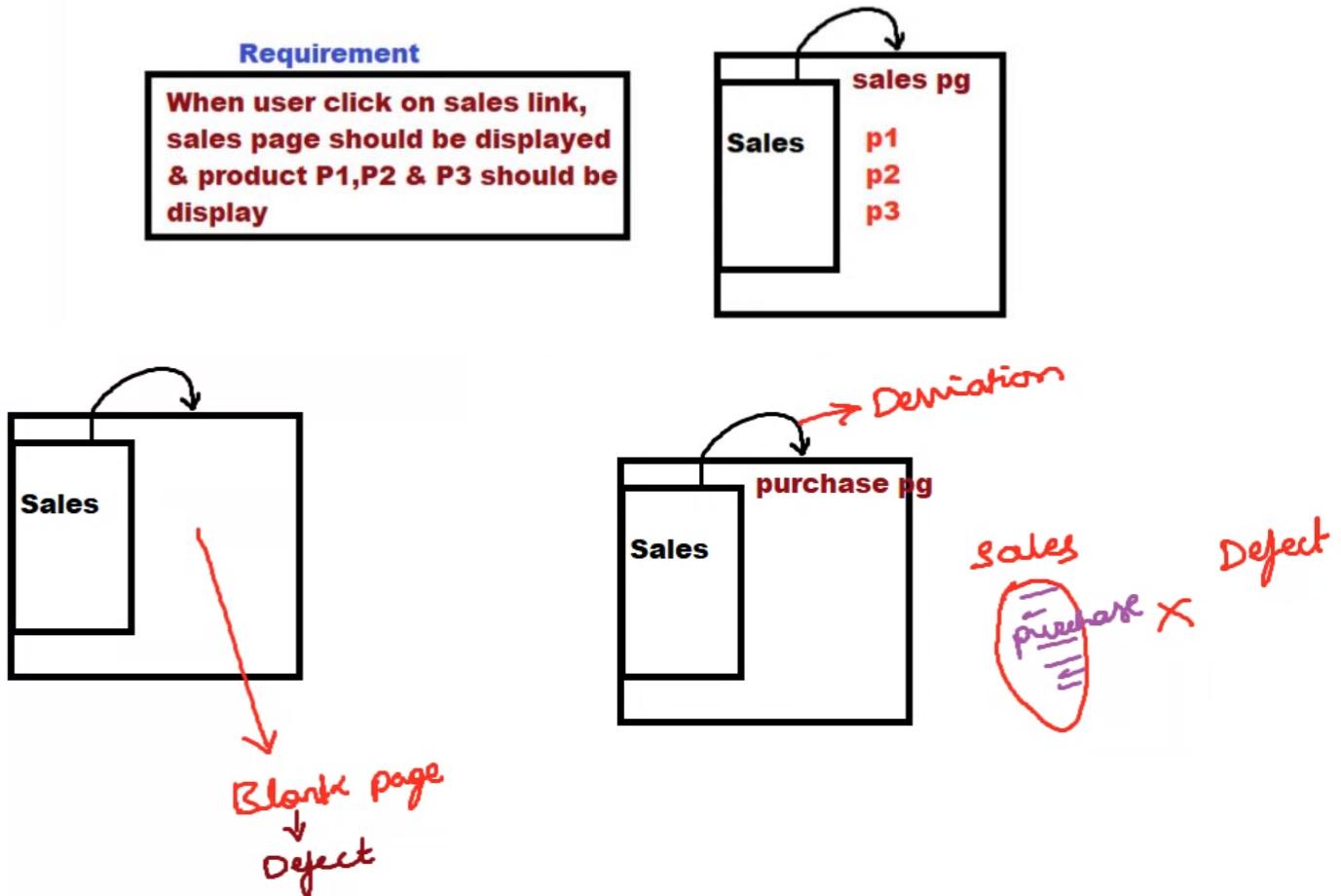
When we do Performance Testing?

- > Different projects we will do performance testing in different ways,
- > Certain project they will do performance testing when the product becomes functionally stable.
- > Certain project they implement performance testing from the beginning of SDLC.

Defect Tracking

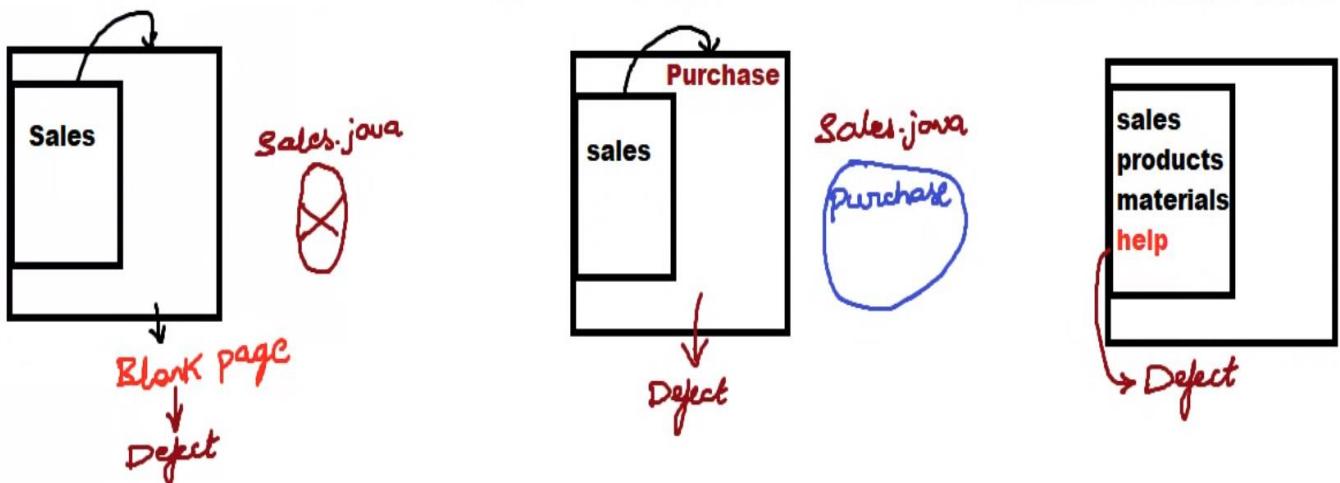
What is a Defect?

- > Any feature which is not working according to the requirement specification is called as Defect.
or
- > Deviation from the requirement specification is called as Defect.



Why we get defects? or types of defects?

- > Due to Missing Implementation
- > Due to Wrong Implementation
- > Due to Extra Implementation



What is the difference between Error, Defect, Bug & Failure?

- > **Error:** It is a mistake done in the code because of which developer will not be able to compile the code & run the code. It is a term used by developers
- > **Defect:** Error found in the s/w or application is called as Defect.
- > **Bug:** It is an informal name given to defect.
- > **Failure:** Defect in the s/w leads to failure. It is a term used by Enduser/Customer.

ISTQB

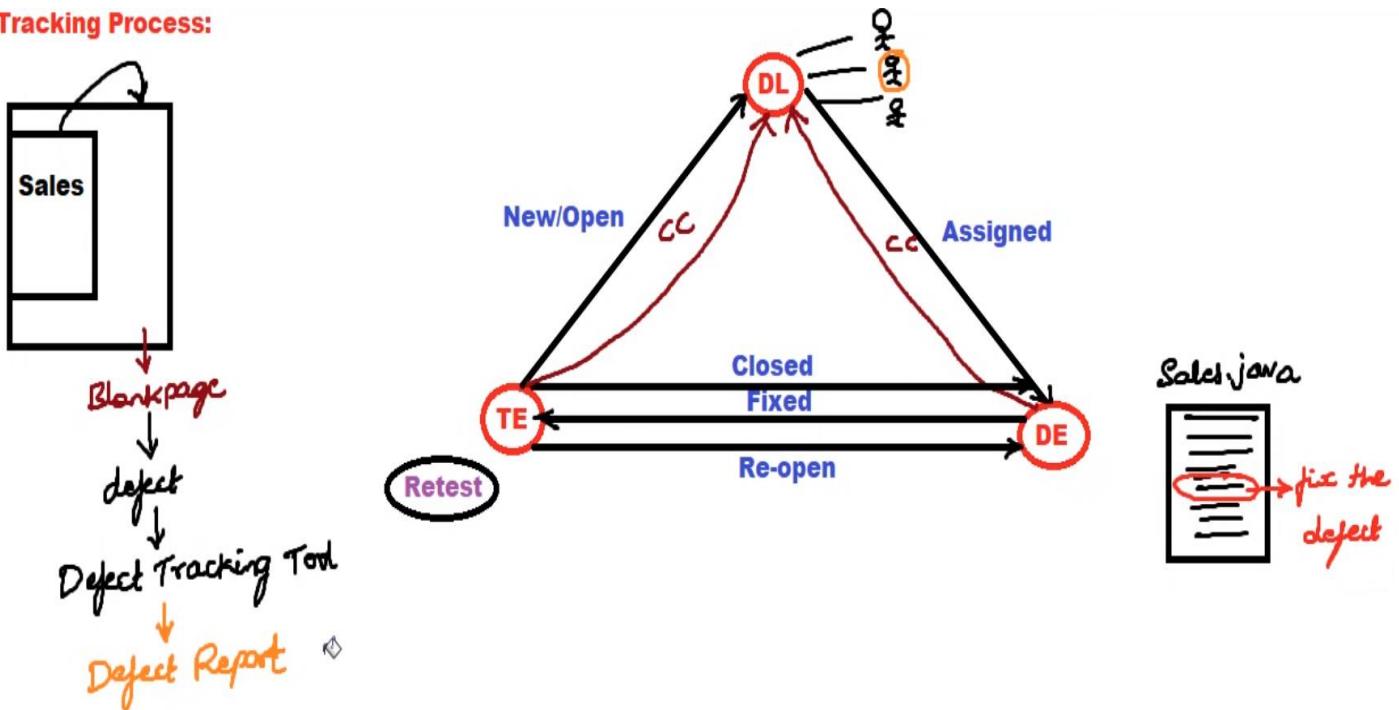
1) Defect accepted by the developer is called as _____ ?

- a) Bug b) Failure c) Error d) none

2) Deviation from the requirement specification which is visible to end users is called as _____ ?

- a) Defect b) Error c) Bug d) Failure

Defect Tracking Process:



Why we should not wait for the permission from TL to send defect report to DL?

- > There will be delay in communicating defect report.
- > As a TE we will be knowing our module in depth. So better take a decision & send report to DL.

Why we should keep CC for the TL while sending defect report to DL?

- > TL is the one who will be attending all meetings such as customer, management & developer meetings. So he should be aware of what are there in the product.
- > To get the visibility the TE is working.

Note: As soon as you find defect we have to communicate defect to developer immediately because,

- > chances are there you might forget the defect.
- > chances are there someone else might send your defect.
- > developer will get sufficient time to fix the defect if you send it early.

Note: For every defect TE will give severity & priority.

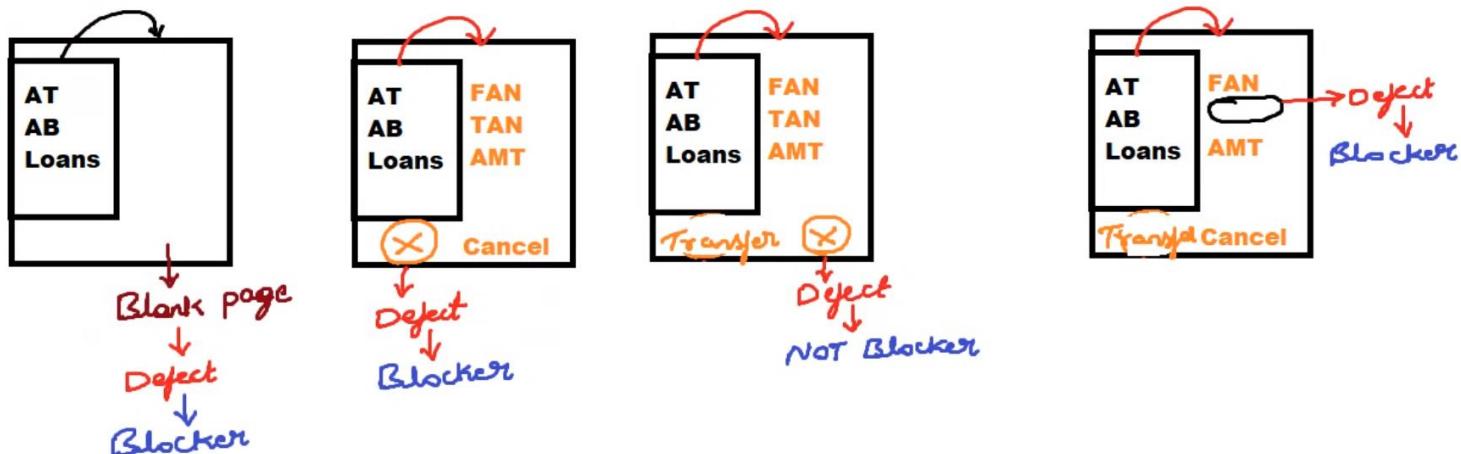
One defect will have one severity & one priority.

Severity

- > It is decided based on the impact of defect on the customer business.
- > There are 4 levels of severity, i.e.,
 - > Blocker/Show Stopper/Fatal Defect
 - > Critical Defect
 - > Major Defect
 - > Minor Defect/Trivial defect

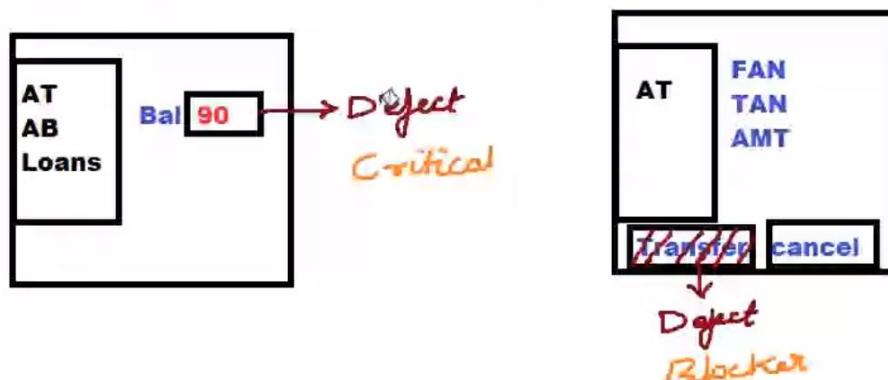
Blocker Defect:

Assume that there is a defect in the application & we are 100% sure that this defect will affect customer business workflow & also block the TE to test the feature. This kind of defects are called as Blocker Defect.

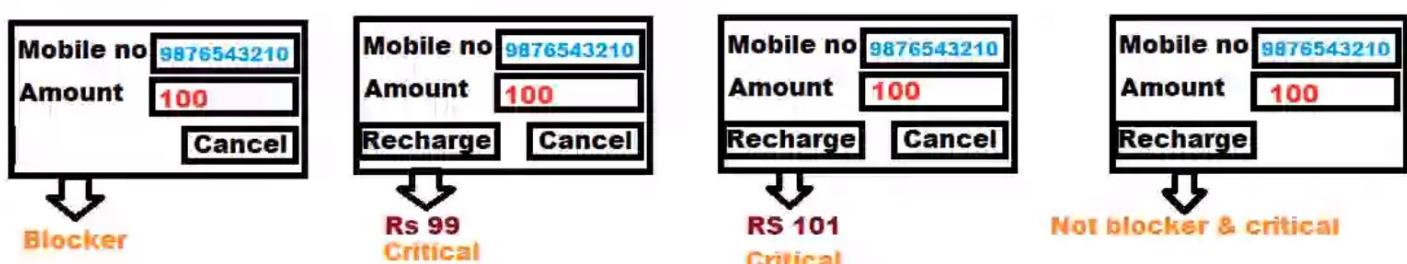


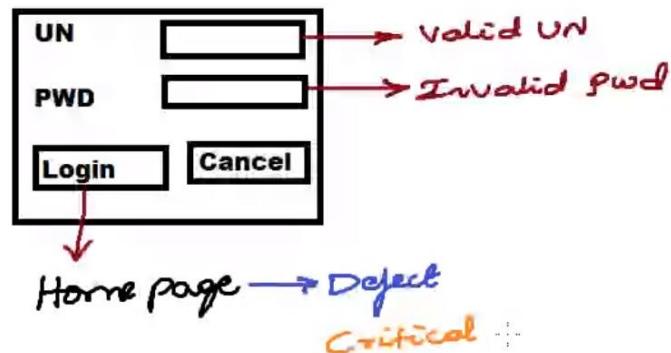
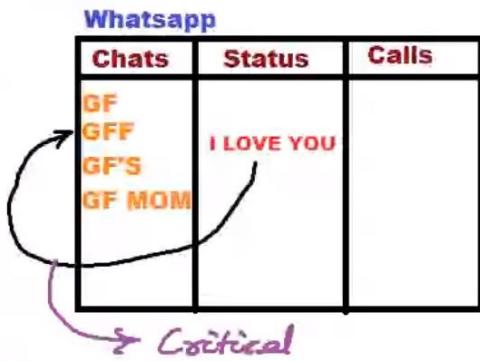
Critical Defect:

Assume that there is a defect in the application & we are 100% sure that this defect will affect customer business workflow. But it will not be blocking the TE to test the feature, i.e., we can still continue testing. This kind of defects are called as Critical Defect.



Req: When user recharge for Rs100, he should get full talktime i.e., Rs.100





Major Defect:

Assume that there is a defect in the application & we are not sure that this defect will affect customer business workflow. This kind of defects are called as Major Defect.

AT
AB
LOANS

FAN
TAN
AMT

Transfer
cancel

✗ Confirmation message
Major

Mobile no 9876543210

Amount 100

Recharge
Cancel

✗ Confirmation message
Major

20.1.

Minor Defect:

Assume that there is a defect in the application & we are 100% sure that this defect will never affect customer business workflow. This kind of defects are called as Minor Defect.

- > Alignment Issues
- > Overlapping Issues
- > Spelling Mistake

Amount **tarfset**

From account no

To account no

Amount

Transfer Cancel

Severity for ATM machine:

1. If any user insert ATM card into ATM machine & enter correct pin, consider it is not accepting. **Blocker**
2. If any user insert ATM card into ATM machine & enter wrong pin, consider it is accepting. **Critical**
3. If any user withdrawal the amount in the ATM confirmation message is not displaying. **Major**
4. If there is spelling mistake in the balance enquiry feature. **Minor**

Assignment:

- 1) Identify severity for ATM machine
2 blocker, 2 critical, 2 major, 2 minor

2) Own application

10 blocker, 10 critical, 10 major, 10 minor
(or)

10 applications

1 blocker, 1 critical, 1 major, 1 minor

Priority:

- > Importance given to fix the defect is called as Priority.
or
- > How soon the defect must be fixed by the developer.
- > There are 3 different levels of priority. They are:
 - > High/Priority P1
 - > Medium/Priority P2
 - > Low/Priority P3
- > **High/P1:** If the defect has priority P1 then developer should fix the defect immediately.
- > **Medium/P2:** If the defect is having priority P2 then developer can fix the defect within same test cycle or within same build or within a release.
- > **Low/P3:** If the defect is having P3 then developer can fix the defect in the upcoming release or within 2-3 releases.

(Examples)

1. In whatsapp user is not able to do video call.

severity:Blocker

priority:P1

2. In amazon in the payment page at the bottom there is a spelling mistake.

severity:Minor

priority:P3

3. In ola user is able to book a ride & finish it by entering wrong OTP.

Severity:Critical

Priority:P1

4. In whatsapp mute functionality is not working.

Severity:Major

Priority:P2

5. In youtube user is not able to play the video.

severity:Blocker

priority:P1

6. In youtube if 1000 users subscribes it is showing as 1500 as subscribers.

Severity:Critical

Priority:P1

7. In facebook login page remember password functionality is not working.

severity:Major

Priority:P1

Note: Any defect which is directly visible to customer in the main page will always take highest priority.

Combination: Interview

1. High Severity (Blocker/Critical)	High Priority (P1)
2. Low Severity (minor)	Low Priority (P3)
3. High Severity	Low Priority
4. Low Severity	High Priority

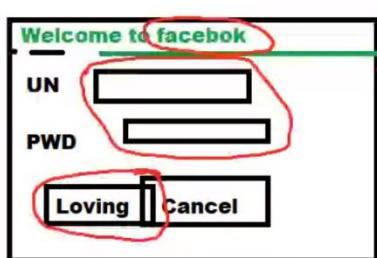
High Severity & Low Priority

Gmail → help → Blank page → Blocker P3

whatsapp → Install 50th time → Blocker P3

whatsapp → Invite → SMS
 — messenger
 — Gmail
 — Inshare → Blank page — Blocker P3

Low Severity & High Priority



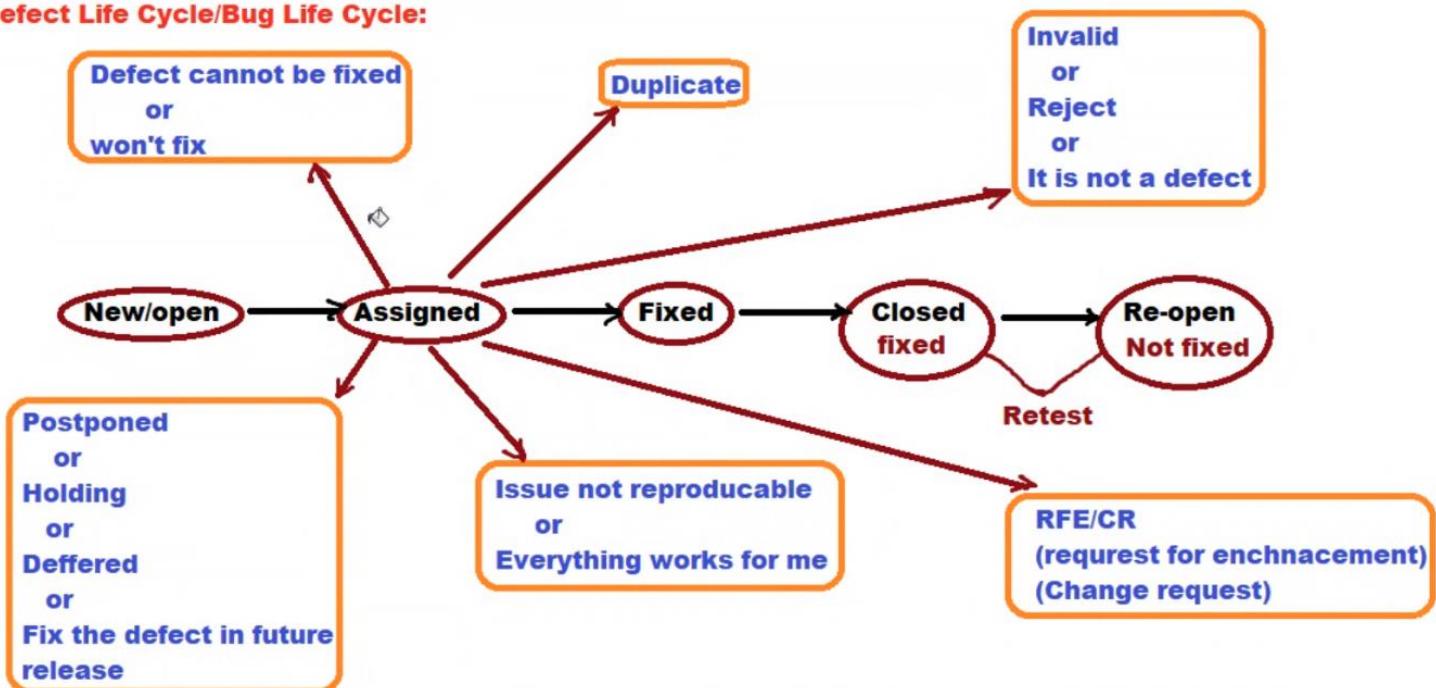
minor P1

Cats	status	Cells

→ minor P1

1) identify 15 defects for any application & give severity & priority

Defect Life Cycle/Bug Life Cycle:



Explain Defect Life Cycle?

> Test Engineer:

- > TE will find a defect.
- > He will login to defect tracking tool & prepare defect report.
- > He will put status as New/Open.
- > He will defect report to DL.

> Development Lead:

- > He will read the report & understand the problem.
- > He will identify the DE who did the mistake.
- > He will put the status as Assigned.
- > He will send the defect report to DE.

> Development Engineer:

- > He will read the report & understand the problem.
- > He goes to the source code & fix the defect.
- > He will change the status as Fixed.
- > He will send the defect report to TE & also keep CC for DL.

> Test Engineer:

- > He will read the report & understand the problem.
- > He will retest the defect.
- > He will change the status as Closed if the defect is fixed, suppose the defect is not fixed then he will change the status as Reopen & send defect report to DE & also keep CC for DL.

>Defect Life Cycle consists of below mentioned status

- > New/Open
- > Assigned
- > Fixed
- > Closed
- > Reopen
- > It is not a defect/Invalid/Reject
- > Duplicate
- > Defect cannot be fixed/Won't fix
- > Fix defect in future release/Postponed/Deferred/Holding
- > Issue is not reproducible/Everything works for me
- > Request for Enhancement(REF)/Change Request(CR)

ISTQB:

What is the difference between new & open status?

For the 1st time while TE will find a defect then status will be new.

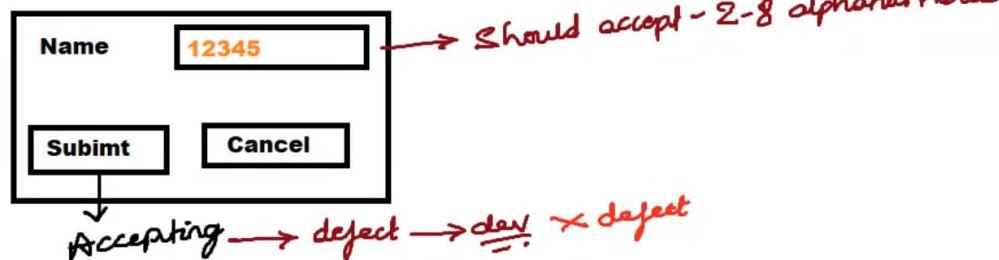
When developers start to work on the defect then the status will be open.

What is Reject Status?

> TE will assume that is is a defect & send it to developer, now developer say that it is a feature & change status as 'It is not a defect' or 'Reject'.

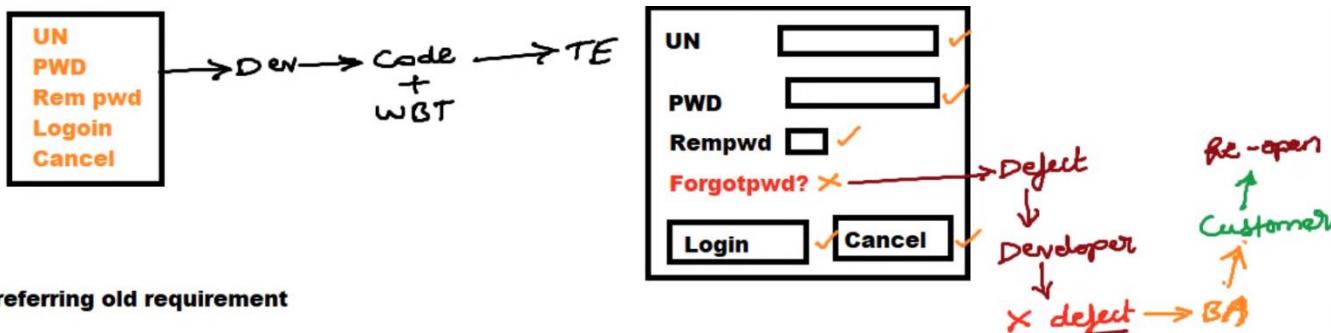
Why we get Reject Status?

> Because of TE misunderstanding the requirement.



> Because of adding extra features

Note: Whenever TE calls extra features as a defect chances are there developer might reject it, in this case TE should reopen the defect & ask developer to update in the requirement.



> Because of referring old requirement

> When build or s/w is wrongly installed/configured

If TE install build wrongly & find defect in software & communicate to developer now developer will tell it is not a defect because code is perfect but TE has not installed software properly.

What is Duplicate Status?

> TE will find a defect & communicate to developer, if already same defect is tracked by some other TE, then developer say that this new defect is duplicate of old defect.



Why we get Duplicate Status?

> Because of testing common features.

> Old TE has already found lot of defect & communicated to developer, if new TE joins same old project & send defect then developer say that this defect is duplicate of old defects.

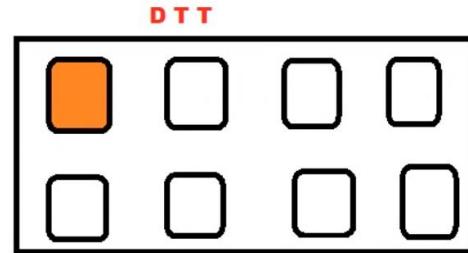
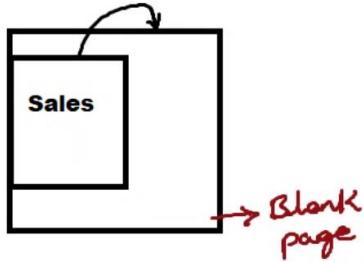
Interview

Why developer say that it is duplicate?

- > To reduce defect count.
- > To reduce duplicate effort.

How to avoid duplicate defects?

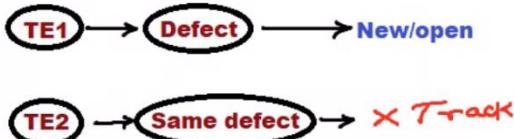
- > TE when he find defect & communicate to DL, he will keep CC for TL & also TE should keep CC for all the TE who is working on same project.
- > Before TE log the bug he should search for duplicate defects in D.T.T by entering certain keywords using advanced search.



Advance search
sales_blankpage

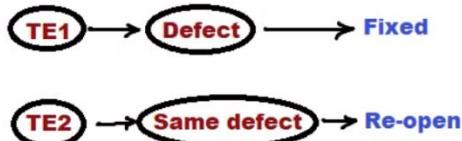
> TE when he find defect, before he prepare report he should cross check with Sr.TE or TL or sometimes developer so that he can avoid duplicate defects.

Case 1:



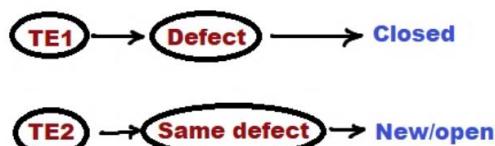
TE will find a defect in the s/w if already same defect is present with status as New in this case we should not track the defect.

Case 2:



TE will find a defect in the s/w if already same defect is present with status as Fixed in this case we should Reopen the defect.

Case 3:



TE will find a defect in the s/w if already same defect is present with status as Closed in this case we should track it as New defect.

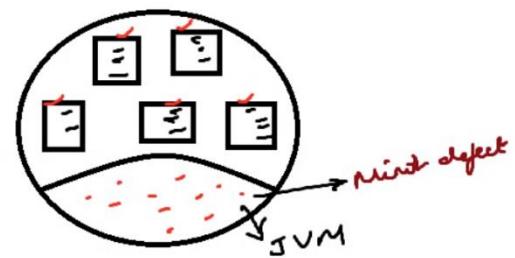
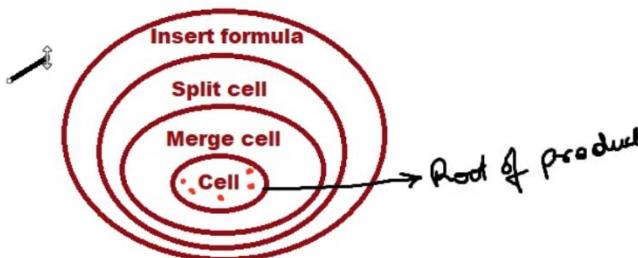
What is Defect cannot be fixed Status?

> Here developer are accepting that it is a defect but they are not in a position to fix the defect in this case developer will give status as defect cannot be fixed.

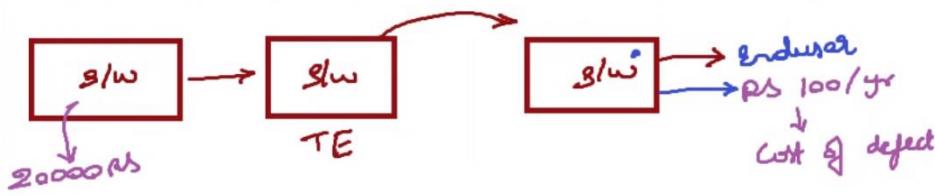
Why we get Defect cannot be Fixed Status?

> If TE finds minor defect in the root of the product & if it is not affecting customer business then developer say that 'Defect cannot be Fixed'.

(suppose if it is Blocker/Critical developer should fix the defect)



> If the cost of fixing the defect is more than the cost of the defect then developer say 'Defect cannot be Fixed'. Here cost of the defect means loss in business because of having the defect in s/w.



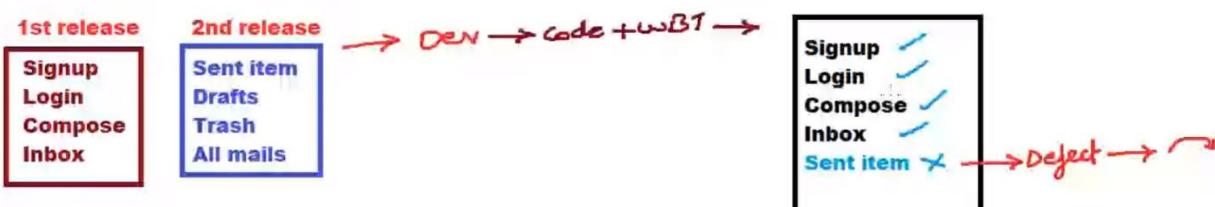
What is Postponed Status?

> Here developer are accepting it as a defect but they are not ready to fix the defect i.e., they will be fixing the defect little later.

Why we get Postponed Status?

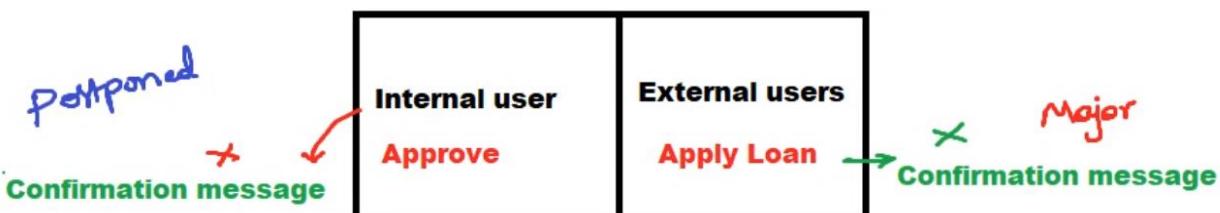
> If TE finds a minor defect in the s/w during the release of a s/w to the customer developer will not fix the defect due to shortage of time & put the status as Postponed.

> If TE finds a defect in the feature which is not required by the customer current release then developer will tell that he will fix the defect in the future release.



> There is a defect when it is sent to the developer they say that customer is expecting lot of changes in the same feature so better postpone the defect until we get clarity from the customer.

> If there is a defect which is exposed to internal users & if that defect is major or minor defect then developer will postpone the defect.

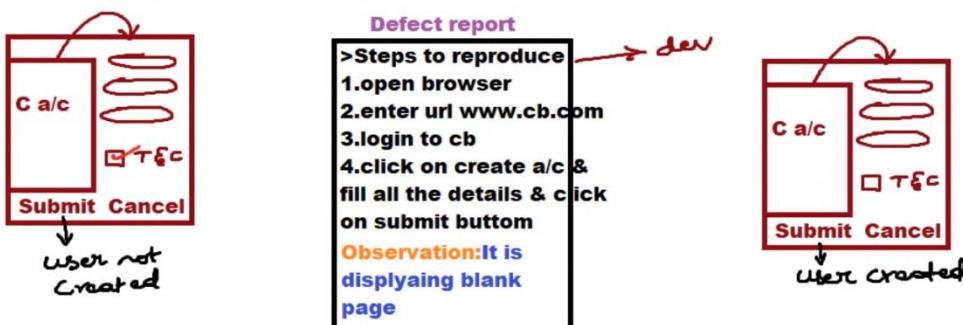


What is Issue not Reproducible Status?

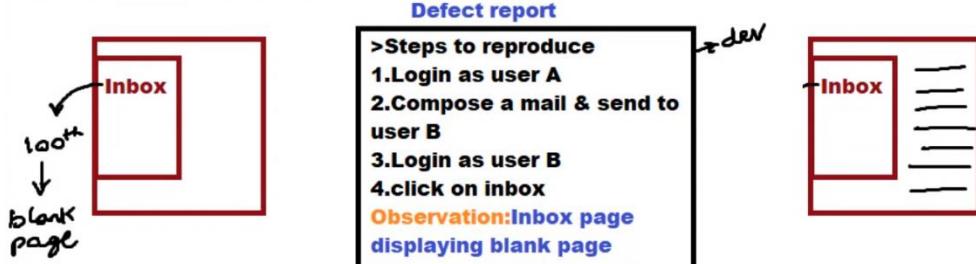
> TE is able to see the defect but developer are not able to see the same defect in such case developer say that 'Issue not Reproducible'.

Why we get Issue not Reproducible Status?

- > Because of platform mismatch
- > Because of OS mismatch
- > Because of browser mismatch
- > Because of browser version mismatch
- > Because of browser setting mismatch
- > Because of improper defect report



> Because of incorrect data



> Because of inconsistent defect

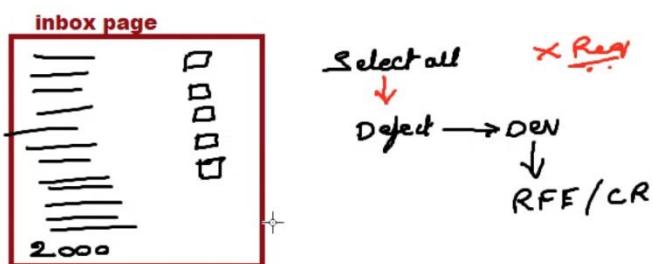
What is Inconsistent Defect?

> Sometimes feature works & sometimes same feature may not work (or) Sometimes defect appears & sometimes it disappears. This kind of defect is called as Inconsistent Defect.

Whenever TE finds this kind of defects he should track a defect & put a note in the defect report stating that 'Defect is Inconsistent'.

What is RFE or CR?

> While testing the s/w if TE finds a defect & if that defect is not a part of the requirement then developer will change status as RFE or CR.

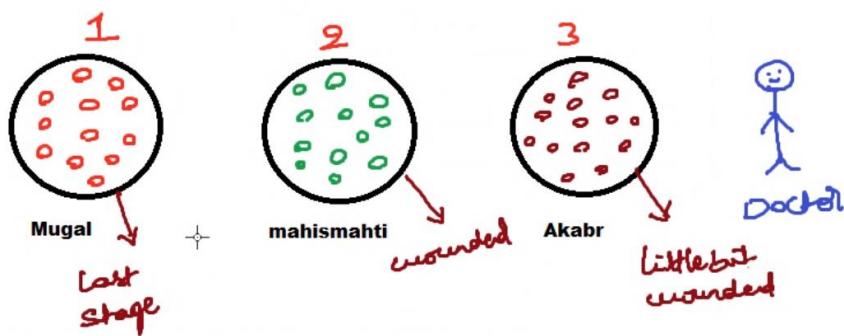


Who can give RFE?

> It can be given by TE or developer & also sometimes by customer.

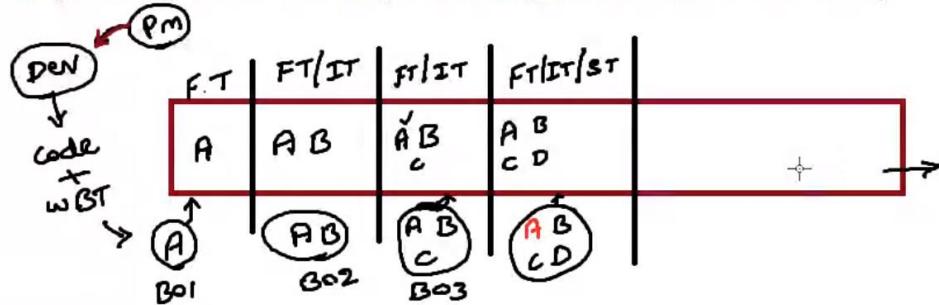
What is Defect Triage?

> When time is very less & if there are too many pending works (defect that are not fixed) they will categorize the defects into different status.
 > All the business critical defects they are going to move it to assigned status.
 > There are some defects which developer says they are going to fix it little later such defects they are going to move it to postponed status.
 > There are some defects which developer says that he cannot fix the defect i.e., they move such defect to defect cannot be fixed/won't fix status.



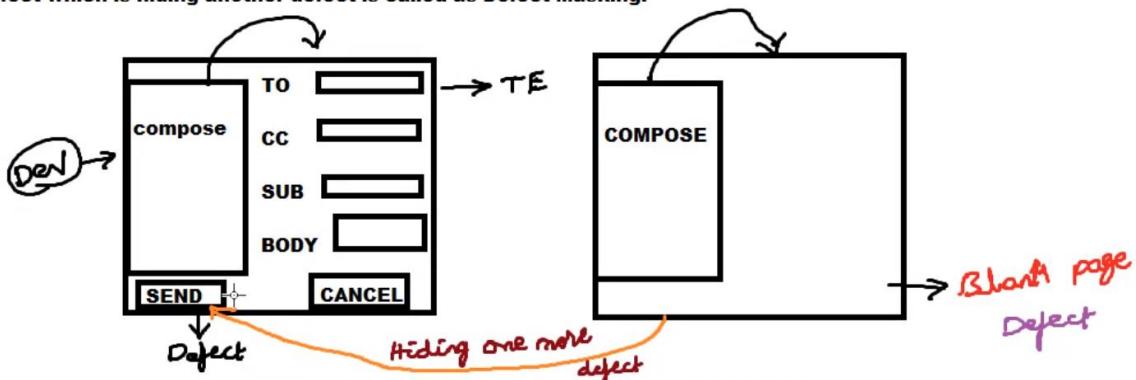
What is Defect Seeding?

> Intentionally introducing the defect & testing the efficiency of TE is known as Defect Seeding.



What is Defect Masking?

> One defect or any defect which is hiding another defect is called as Defect Masking.



What is Defect Pesticide?

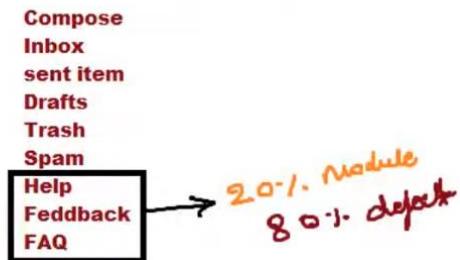
> If the same test cases are repeated again & again we will no longer find new defect this concept is called as Defect Pesticide.

To overcome this we need to update the test case or we need to write new test case.



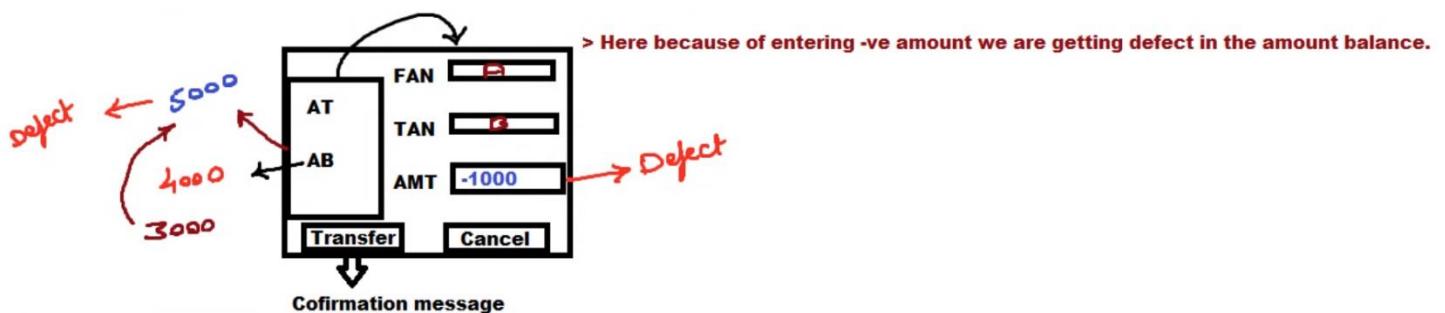
What is Defect Clustering?

> According to 'Parrato Principle' 80% of the defect will be found in 20% of the module/feature.



What is Defect Cascading?

> One defect triggering another defect is called as Defect Cascading.



What is Defect Leakage?

> After developing the s/w i.e., after the realease if customer or enduser find the defect by using the application in the initial stage itself this kind of defect is called as Defect Leakage.

What is Latent Defect?

> If the defect is present in the s/w for longer duration of time then that type of defect is called as Latent Defect.



Defect Report/Defect Report Template/Explain the attributes of Defects

1. Defect Id:
2. Build Name:
3. Release Name: **Tiger**
4. Requirement No.: **90.1**
5. Module Name:**Compose**
6. Build Id: **B01**
7. Status: **New/open**
8. Severity: **Blocker**
9. Priority: **P1**
10. Reporter: 
11. Assign To:
12. Test Case Name: **gmail_compose_blank page**
13. Test Data: **UN PWD**
14. Test URL: **www.gmail.com**
15. Defect Summary/Brief Description: **When user clicks on compose it is displaying blank page**
16. Detailed Description: OS:
Browser:
Browser Version:
Steps to Reproduce: **open browser,enter URL, click on compose,it is displaying blank page**
17. Observation/Actual Result: **Blank page**
18. Expected Result: **Compose page**
19. Attach Screenshot: 
20. CC
21. Submit
22. Cancel
23. Reset
24. Bug History

What is defect tracking tool?Name some defect tracking tool?

>Defect Tracking Tool is a software which is mainly used to store the defects in a centralised place & communicate defect to developer in an organised way.

Tools:

- 1.Bugzilla
- 2.QC/ALM(Quality center/Application lifecycle management) **Project management tool**
 - Store requirement
 - Track defect
 - Write test case
- 3.Jira
- 4.Bugzini
- 5.Mantis
- 6.Bug pro
- 7.Rational clear quest
- 8.Core plus
- 9.Bug net

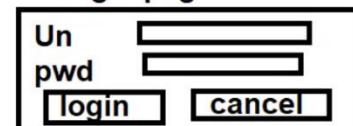
How does defect tracking tool works?

Bugzilla

Step 1:Open the browser & enter Url https:// Bugzilla//login.jsp

HOME

Login page



The form consists of a rectangular box with a border. Inside, there are two input fields: one for 'Un' and one for 'pwd'. Below the fields are two buttons: 'Login' on the left and 'cancel' on the right.

→ Here TE will Search for duplicate defect by Entering defect ID

click on Advance Search

Step 2: TE will search for duplicate defects by using search text field If you know defect ID. Suppose if you don't know defect ID then you should go for advance search.

Advance search page

Project name	<input type="text"/>
Release name	<input type="text"/>
Build ID	<input type="text"/>
Status	<input type="text"/>
Severity	<input type="text"/>
Reporter	<input type="text"/>
Defect summary	<input type="text"/>
<input type="button" value="Submit"/>	<input type="button" value="Cancel"/>

List of matching defects			
Defect id	Status	Defect summary	Severity
6789	New closed	Compose is going to be blank page	Blocker
55256	New	Compose button is disabled	Blocker

Step 3: If there are no duplicate defects then TE will click on create issue/enter new defect.

1.Defect ID:
2.Defect name:
3.Release name: Tiger
4.Requirement no: 90.01
5.Module name: Compose
6.Build id: B01
7.Status: new / open
8.Severity: Blocker
9.Priority: P1
10.Reporter:
11.Assigned To:
12.Test case name: gmail_compose_blank page
13.Test data: UN PWD
14.Test url:
15.Defect summary/Brief description: when user click
16.Detailed description: compose,balnk page

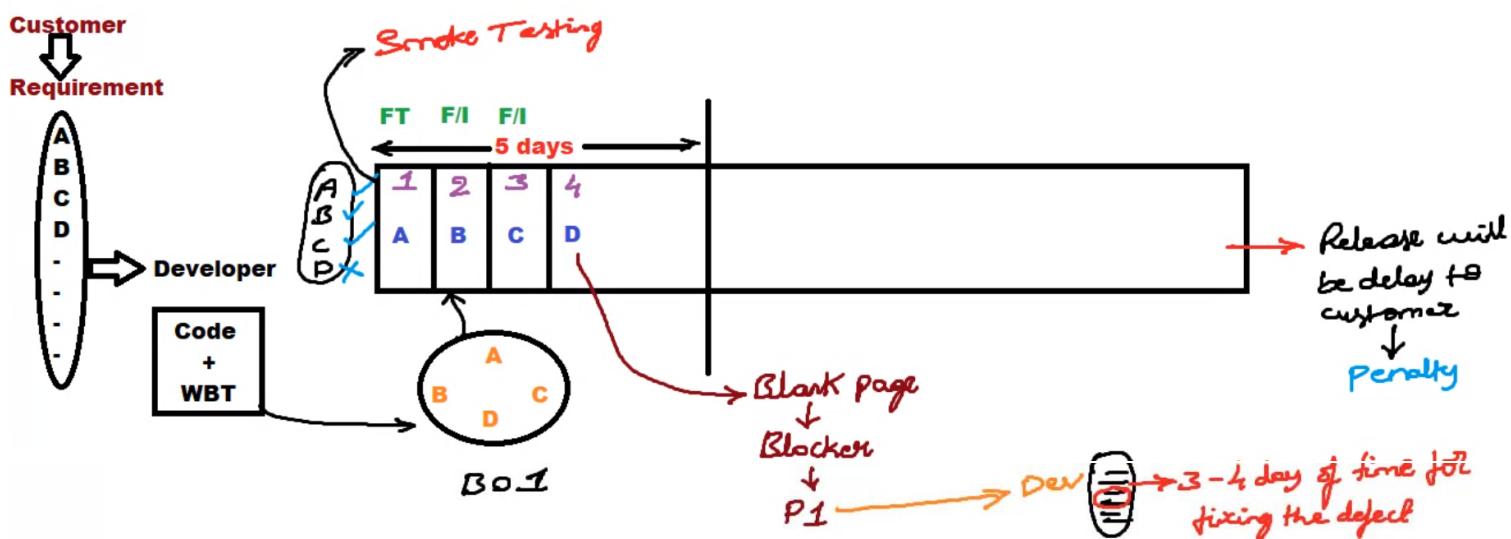
Developer name	Timmings	Change
dog	1:54:55sec am	

How you will convince dev that it is really a defect?

- 1.I'll send proper test data and i'll tell os & browser used.
- 2.I'll write proper procedure ,still if he is not getting the defect then i'll send the screen shot then i'll ask developers to get connect to my computer and also i'll give demo how I am getting the defect.
- 3.still if the defect is not visible means i'll try it in 4-5 computers near by me,if it is not visible in all the computers means i'll close the issue, if the defect is visible in all the computer means i'll ask developer to fix the defect.

Smoke testing/Sanity Testing/Skim Testing/Dry Run Testing/Build Verification Testing/Confident Testing/Health Check of Product Testing/Positive testing

Testing the basic & critical features of an application before we do thorough testing is called as **Smoke Testing**.



> According to above example customer will give requirement for A, B, C, D modules. Developer will write code & do WBT & give build for TE. TE will start doing thorough FT for A, B, C modules for first 3 days, 4th day when he click on D module it is displaying blank page. TE will immediately communicate defect to developer & developer will tell he wants 3-4 days to fix the defect, because of this test cycle gets postponed, release will be delayed & customer will charge penalty for the company.
> In order to overcome this problem in every company as soon as they get build they should do **Smoke Testing**.

Advantages of Smoke Testing:

- > TE can find all blocker defects in the beginning itself.
- > Developer will get sufficient time to fix the defect.
- > Test Cycle will not get postponed & release will not be delayed.

Note: 1. In Smoke Testing we test only basic & critical features.

2. Here we will take basic & critical features & test for only 1 or 2 important scenarios.

3. Here we should do only positive testing.

Q) Which testing is called as positive testing? -> Smoke Testing

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

How to do Smoke Testing?

Gmail

Signup,login,compose,inbox,sentitem,trash,drafts,spam,outbox,allmails,FAQ,starred,important,help,feedback,schedule,snooze d,Setting,primary,promotions,social,search,calender,contacts,meet,replay all,levels,privacy & policy,mark all,logout.

Features to be tested as a part of smoke testing	Features to be tested as not a part of smoke testing
signup login compose inbox sent item all mails settings search repalyall logout	Trash drafts spam outbox FAQ starred important help feedback schedule snoozed primary promotions social

Smoke scenarios:

login

1. To check that when user click on login, login page should be display
2. To check that when user enter valid UN & PWD click on login button, home page should be display

Inbox

1. To check that when user click on inbox, inbox page should be dispaly
2. To check that when user click on any one mail, mail should be display & content should be display

Write smoke scenario for below mentioned application(list min 30 features)

- 1) Facebook
- 2) whatsapp
- 3) Swiggy
- 4) Ola



How to do Smoke Testing for Mobile?

power button, Home screen, Speaker, Camera, Volume button, About phone, Network, Ram, Sim slot, Rom, Calling button, Brightness, Settings, Hotspot, Alarm, data, Security, Bluetooth, Torch, Airplane mode, Location, Wifi, Gallery, Finger print, Scanner, Screen lock, Charger plugin slot, Headphone plugin, DND, Auto rotate, screen cast, Battery saver, screen recorder, Mute, Memory card slot, Time.

Features to be tested as a part of smoke testing	Features to be tested as not a part of smoke testing
Power button	Finger print
Home screen	Screen lock
Speaker	Charging plug
Camera	Headphone plugin
Volume button	Mute
Network	Memory card slot
Ram	Time
Sim slot	
Rom	
Calling button	
Settings	
Data	
Security	
	About phone
	Brightness
	Hotspot
	Alarm
	Airplane mode
	Location
	Wifi
	Gallery
	Scanner
	DND
	Auto rotate
	Screen cast
	Battery saver
	Screen recorder

When we do Smoke Testing?

- > Before developer give build to testing team they should do Smoke Testing so that if too many defects are there means they need not give build to testing team.
- > Whenever new build comes from development team we always start with Smoke Testing, because adding/modifying/removing the features/fixing the bugs might have impact on basic & critical features of an application. To avoid that we do Smoke Testing.
- > One who install product to confirm that product is installed properly, he will do smoke Testing.
- > Before customer does Acceptance Testing he should do Smoke Testing
 - > To confirm that complete product is received properly or not.
 - > To confirm that product is carefully installed & configured or not.

Why we do Smoke Testing?

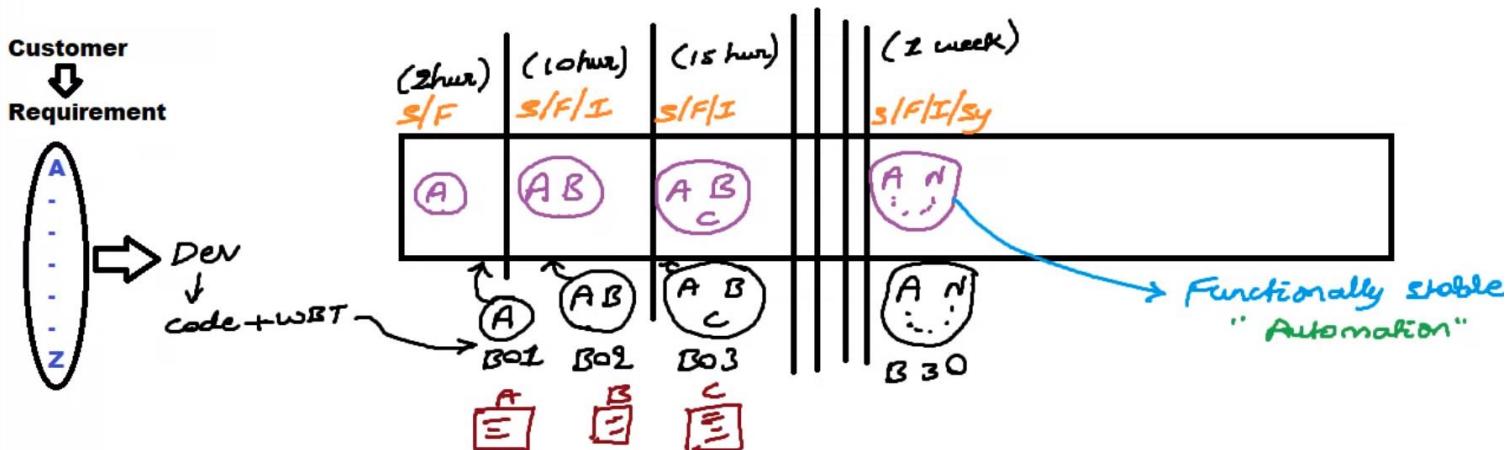
- > To check whether the product is testable or not:

It means in the beginning only if you find too many defects product is not testable so better stop testing & spend all the time identifying some more scenarios.

- > Do smoke testing in the beginning itself, if you find blocker defect communicate it to developer so that developer will have sufficient time to fix the defect.
- > To check whether the product is installed properly or not.
- > To check whether we have received broken build from the development team.
- > It is like health check of the product so we should do smoke testing.

Note: Smoke Testing can be done in 2 ways

- > Manual
- > Automation



(IQ)

Do you write smoke scenarios & test cases?

Yes

Will you go for automation for smoke testing?

Yes

What is the difference between test scenario, test case & test script?

Test Script: Conversion of manual test cases into automation is called as test script or automation script.
By looking into test case we write test scripts.

What is the difference between smoke & sanity testing?

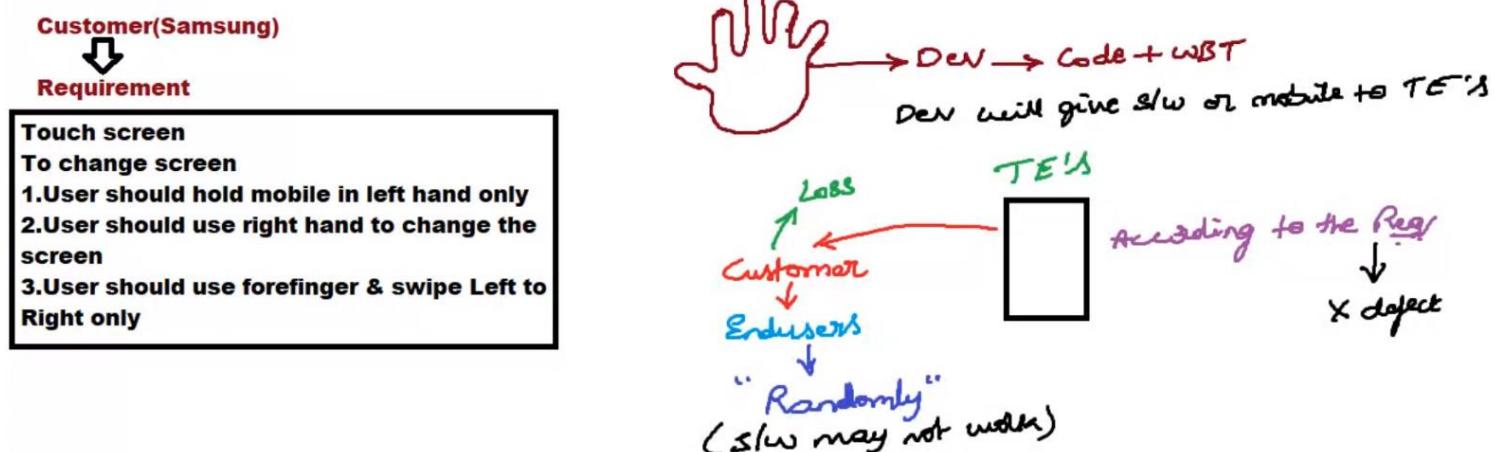
As per my knowledge there is no difference between smoke & sanity testing but i have gone through some of the websites & got to know that there is some difference between them.

- > Guru99.com
- > Google.com
- > Quora.com
- > Allinterviews.com
- > Softwaretestinghelp.com

Smoke	I	Sanity
>It is shallow & wide testing.		>It is deep & narrow testing.
>Only positive testing.		>Both positive & negative testing.
>Here we write test scenarios & test cases.		>Here we don't document scenarios & test cases.
>Here we go for automation.		>Here we don't go for automation.
>It is done by both developer & TE.		>It is done by only TE.

Adhoc testing/-ve testing/Out of box testing/Monkey testing/Gorilla testing

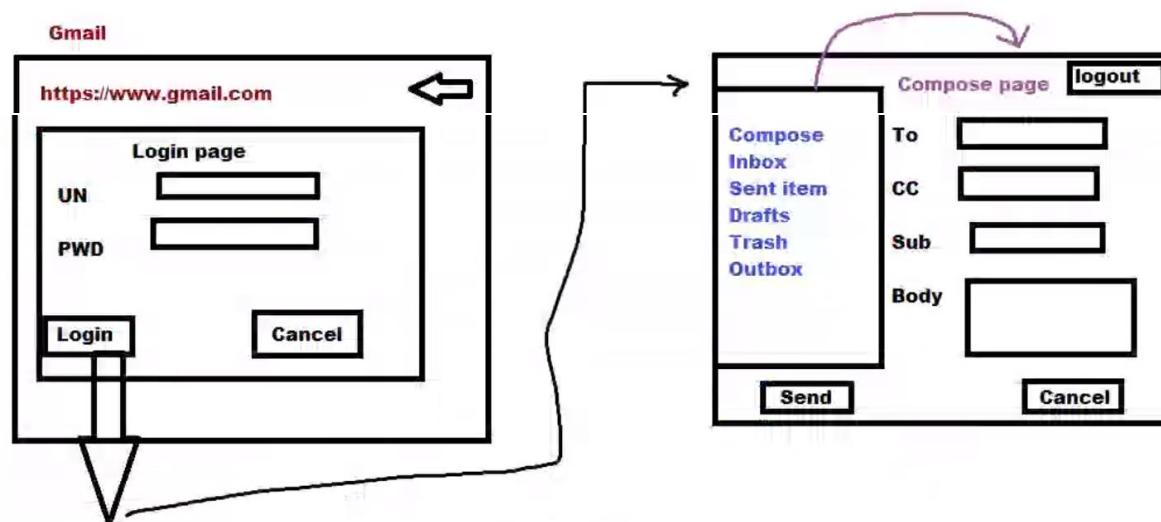
Testing the software or application randomly is called as Adhoc Testing, wherein we don't refer any kind of formal documents like test case or test scenarios or requirement.



Why we do Adhoc Testing?

- > Chances are there enduser or customers might use the software reandomly & they might find defects, in order to avoid that we have should do Adhoc Testing.
- > If you look into requirement & test the software the number of defects which we are going to catch will be less, so we have to think out of the requirement & test the software to find more number of defects.
- > We do this to increase the defect count.
- > We do this to improve test coverage.
- > Since requirement are not followed by customer or end user we should do Adhoc Testing.
- > The intention of doing Adhoc Testing is to somehow break the product.

How to do Adhoc Testing?



> Login as user click on compose, enter values for all the fields, click on send, logout. Click on browser back button & check whether login page is displayed or it should ask you to enter username & password.

(IQ) Write 5 Adhoc Scenarios for login page.

1. Open the browser enter URL, in the login page enter invalid UN & Valid PWD click on Login.

4 scenario

Adhoc Scenarios for different application:

1. Take 2 mobile instal whatsapp in both the mobile & try to register in the same mobile no for both the mobile & check wether user can register or not.
2. open Phonepay application transfer the amount through unregistered mobile no & check wether user can transfer or not.
3. Login to facebook account in 2 different browsers in one of browser change password in another browser do any action check wether application is asking to enter the change password.
4. In ola application enter destination address and same as current address check wether it is accepting or not.
5. In filpkart search a product & book a product without entering shipping address & check wether user can book or not.

When we do Adhoc Testing?

- > As soon as you get build we should do smoke testing, since smoke is positive testing here we should not do adhoc testing.
- > While doing FT/IT/ST either in between or in the end if you have time we can do adhoc testing, if time is not there then document the adhoc scenarios & communicate to TL.
- > Once after complete software is tested as per requirement then we can do adhoc testing by thinking outside the requirement.
- > Once after software is tested for 10-15 cycles as per requirement & if the product is stable then we can do adhoc testing.
- > While doing FT/IT/ST if you come up with good adhoc scenarios stop doing FT/IT/ST test for the adhoc scenario but don't spend too much of time in doing adhoc testing immediately switch back to FT/IT/ST.

(IQ)

1. Do you write adhoc scenarios & test cases?

> Yes sir

2. Why we should not do adhoc testing in early stage itself?

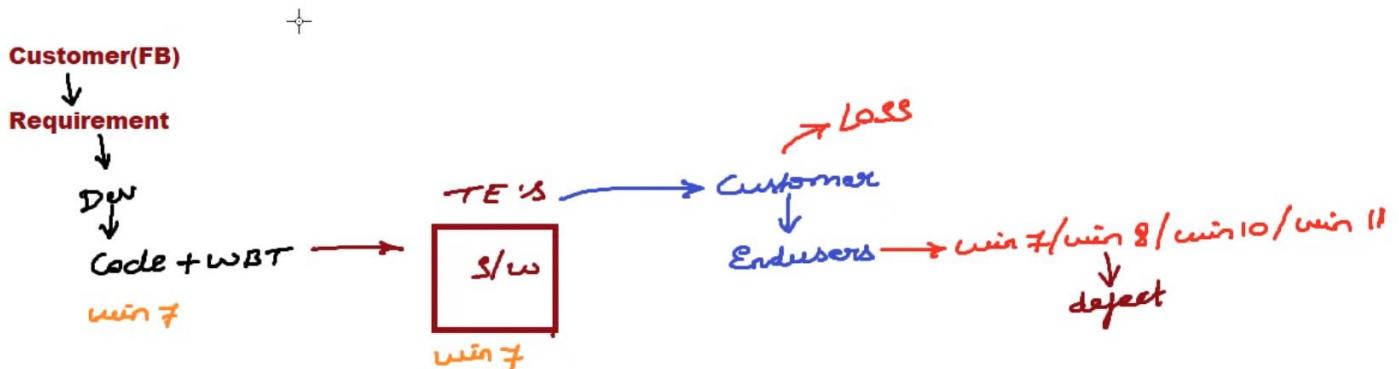
- > As a test engineer first we should check whether software is testable or not, so we should not do adhoc testing.
- > Since customer always use software in positive way so first we should do positive testing & not adhoc testing(negative testing).
- > In the beginning only if you spend more time in doing adhoc testing at the end we might not get sufficient time to do positive testing.

3. Types of Adhoc Testing?

- > **Buddy Testing:** TE will sit with developer & come up with creative scenarios & test the software.
- > **Pair Testing:** TE will sit with another TE & come up with creative scenarios & test the software.
- > **Monkey Testing:** TE should test the software like a monkey without applying any logic.

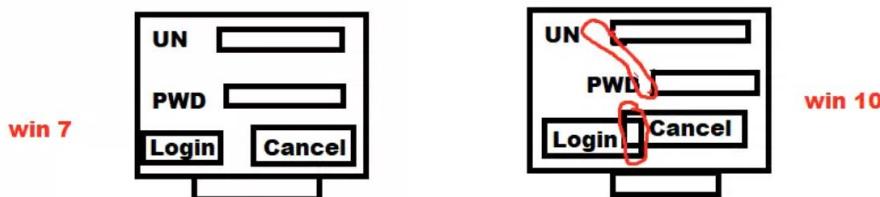
Compatibility testing/Configuration/Portability

Testing the application or software in different hardware & software environment is called as Compatability Testing.

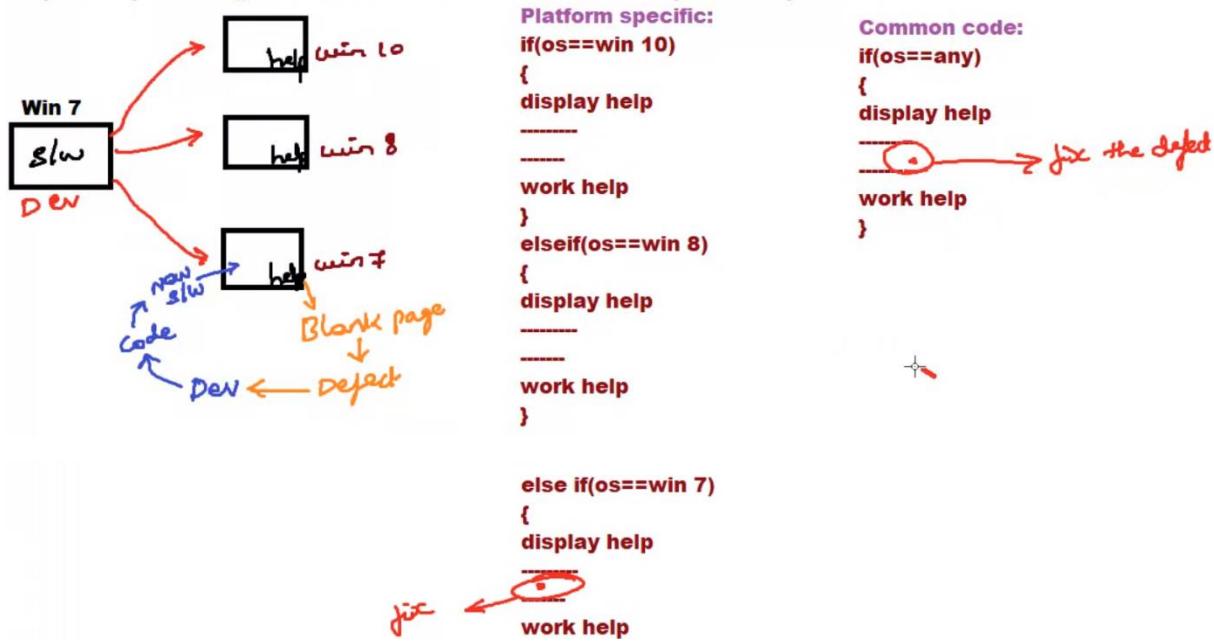


Why we do Compatability Testing?

- > Chances are there developer would have developed the application in one platform, TE would have tested the application in same platform. When the product is launched, endusers might use the application in different platforms. Software which works in one platform may not work in other platform because of some defects, so bad name spreads in the market & also customer usage goes down. To avoid this we should do Compatability Testing.
- > We do compatability testing to check whether features are working consistently in all the platforms.

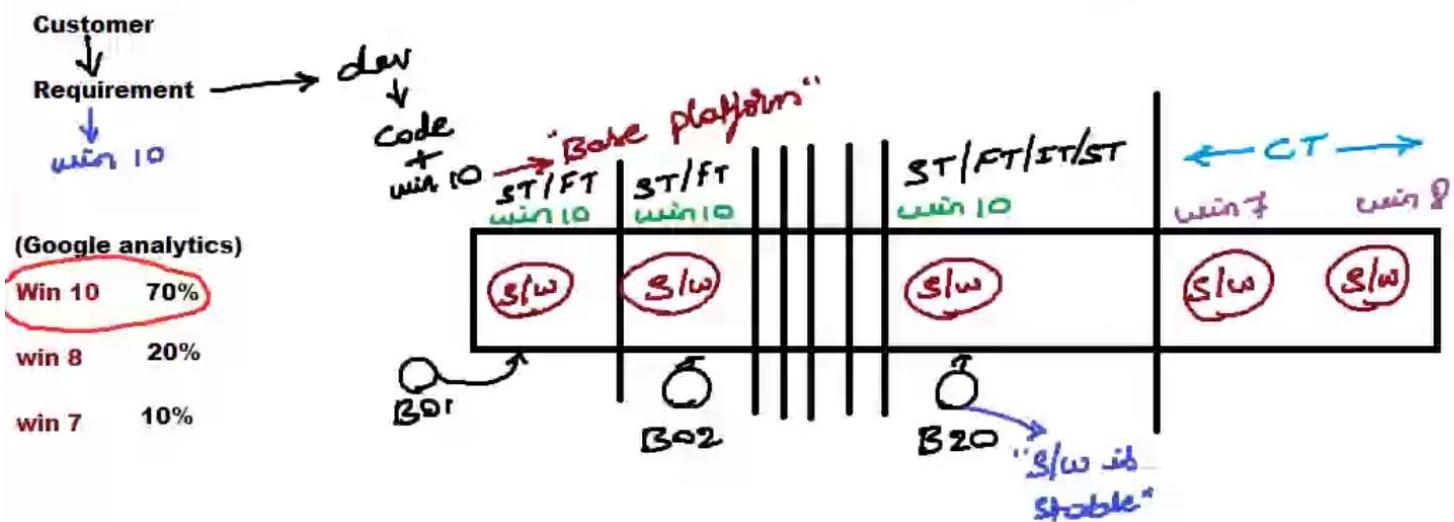


> Developer would have written common code & they claim that it works in all the platforms, so we may have to test in every platform & confirm that it really works. Chances are there developer may write platform specific code & say that it works in respective platforms,, in that case we have to test it in that particular platform.



When we do Compatibility Testing?

> When the product is functionally stable in base platform then we think about testing the software in other platforms.



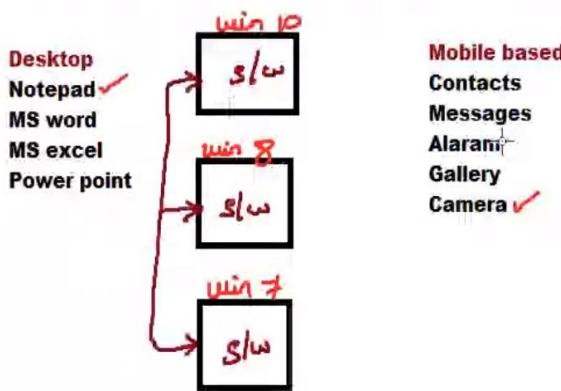
How to do Compatibility Testing?

> It depends on the type of application, there are 3 types of application.

> Stand Alone Application:

Here we take .exe file or setup file & install in computer or mobile & only one user can access the software at a time. To use this software both internet & server is not required. This kind of application are called as Stand Alone Application.

Ex: Calculator, Alarm, Calender, MS Paint, etc



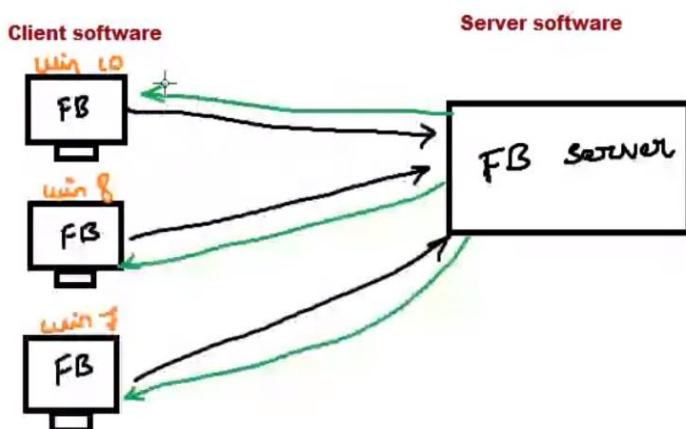
Device name	Andriod version			
	Jalleybean	kitkat	oreo	Nought
Redmi	✓	✓	✗	✗
Moto	✓	✓	✗	✓
Samsung	✓	✗	✓	✓

> Client-Server Application:

Here there are 2 software i.e., client software & server software, we use client software to interact with server software. To use this software we need both internet & server. This kind of applications are called as Client-Server Application.

Ex: Prime, Netflix, Facebook, Whatsapp, etc

Note: Here we do compatibility testing for client software & not for server software.



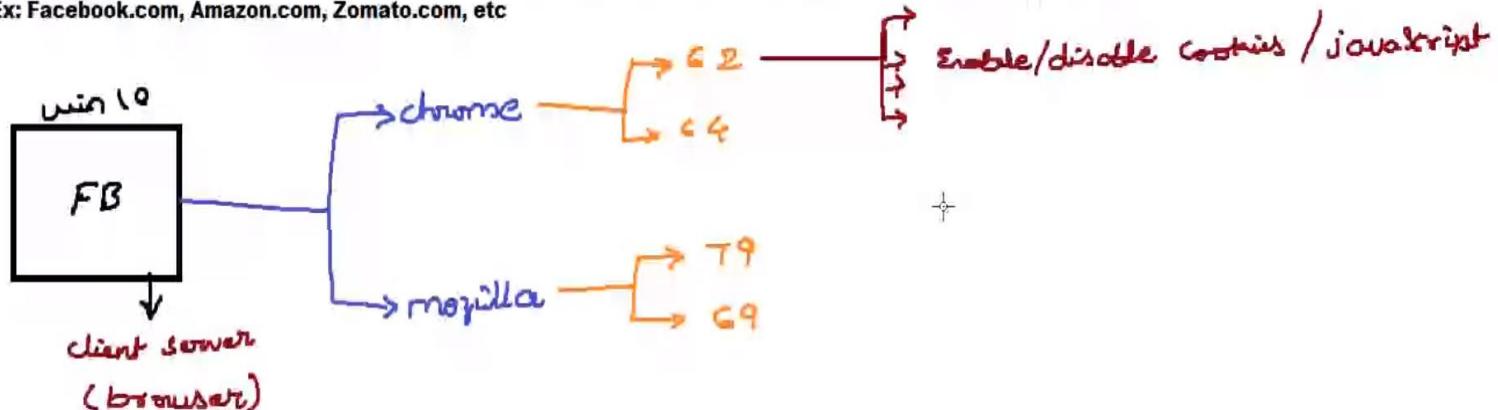
> Web Application:

It is kind of client-server application wherein browser behaves like a client.

(or)

Any application that can be accessed by opening the browser & entering url can be called as Web Application.

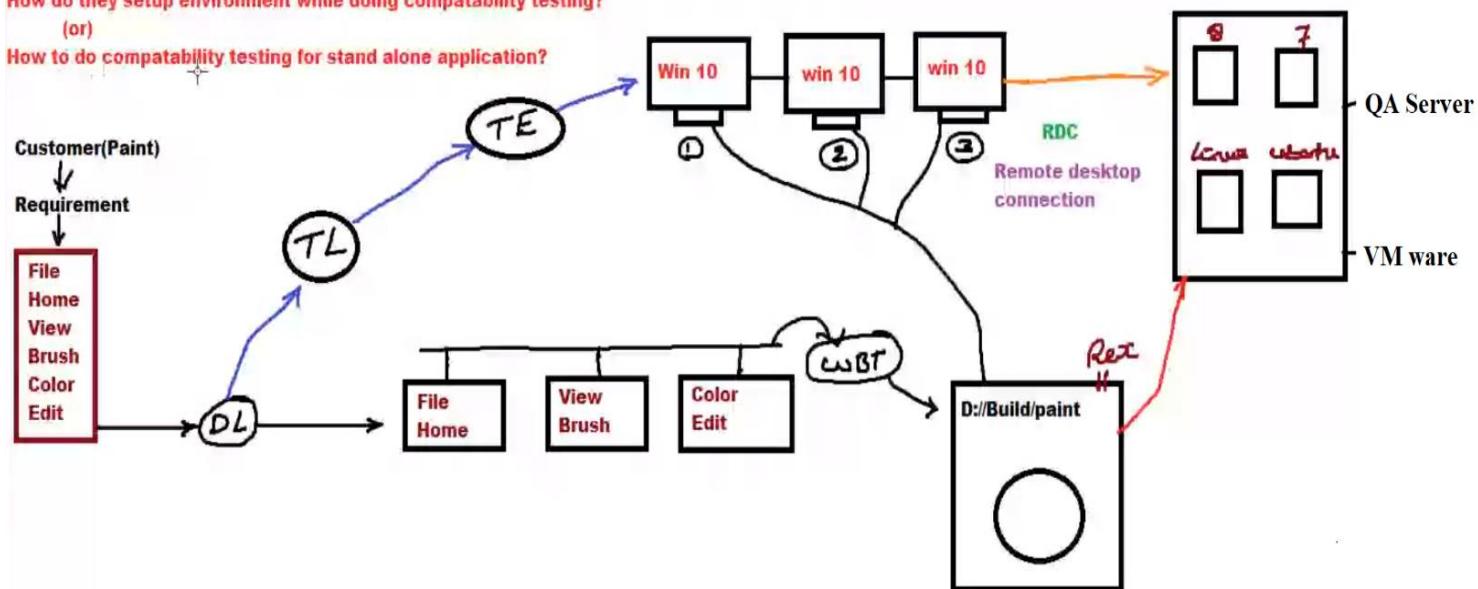
Ex: Facebook.com, Amazon.com, Zomato.com, etc



How do they setup environment while doing compatibility testing?

(or)

How to do compatibility testing for stand alone application?



> Customer will give requirement for paint for both developer & TE. DL will assign features to developer & every developer will write code & do WBT for their respective features. Once WBT is completed they will compile the code & compress & store the build in development server in below mentioned path D://Builds//Paint

> DL will send a mail to TL stating that 'Build is for testing'. TL will assign features to TE & ask them to do Compatibility Testing. Every TE from their computer connect to Rex server & copy paste & install build in Windows 10 platform (Base Platform) & TE will test paint software for base platform (Smoke, FT, IT, ST, Adhoc).

> Once after testing is completed for base platform, then TE by using RDC (Remote Desktop Connection) from their computer will remotely connect to QA server, there from every individual OS TE will connect to development server & copy paste the build & install paint software in every individual OS & start doing compatibility testing for paint software in different OS.

> This is how they do Compatibility Testing for Stand Alone Application.

What is Hardware Compatability Testing?

- > Testing the application or software in different hardware environment is called as **Hardware Compatability Testing**.
- > We should test it in
 - > Different processor:
 - > i.e., we should test it in different makes & manufacturers.
 - > Ex: AMD, Intel, Gigabyte, Snapdragon.
 - > Different RAM:
 - > i.e., we should test it in different makes & manufacturers (basically test for different size & speed)
 - > Ex: HP, Sandisk, Kingston, HyperX.
 - > Different Motherboard:
 - > Ex: Acer, Intel, Gigebyte, etc.
 - > Different VGA Cards:
 - > Ex: Nvidia, Faxon, ATI, AMD.

What type of defects are found while doing Compatability Testing?

- > Alignment Issue
 - > Overlapping Issue
 - > Scattered Contents
 - > Change in look & feel of the application
 - > Image with certain formats may not be displayed in certain browsers.
 - > Scroll Bar Issue
 - > Sometimes scroll bar may not be present or sometime scroll bar will be there but it may act like image.
- Note:** If the feature is not working in all the OS & browser then it is functionality defect.
If the feature is not working in any 1 of the browser or OS then it is Compatability Defect.

(IQ)

When time is very less what is your approach for testing?

- > We have to follow very focused approach
 - > We have to test all the critical features first.
 - > Test those features which gives more coverage in less time.
- > I will come up with all the tricks to catch more defect in less time
 - > I will test new or modified features.
 - > I will test those features which are having impact because of new or modified features.
 - > I will test those features which are developed by new developers.
 - > I will test those features which are developed by using new technology.
 - > I will test those features which is developed by the developer who manually does more mistakes.
- > I will not do certain activities which is really not important.
 - > I will not do more negative testing.
 - > I will not do Adhoc Testing.
 - > I will avoid doing usability testing
 - > I will not test in all the platforms
 - > I will test it in only important platforms
 - > I will stretch myself for longer hours or long duration & complete the testing.

What is Forward & Backward Compatability Testing?

- > **Forward CT:** Testing the software for latest or new version of OS & browser.
- > **Backward CT:** Testing the software for old or previous version of OS & browser.

Test case

What are the drawbacks of not writing Test Case?

or

What will happen if we test the software by looking into requirements?

> There will be no consistency in testing, if you look into requirements & test the software.

> TE will miss lot of scenarios & defects.

> Testing does not depends on memory power of TE

> Quality of testing varies from person to person if you look into the requirements & test the software.

> Testing is not depends on mood of the TE.

> Testcase coverage will not be good.

30. Amount transfer

30.1. FAN text field

> It should accept valid a/c no

> It should accept those no which is created by manager

30.2. TAN text field

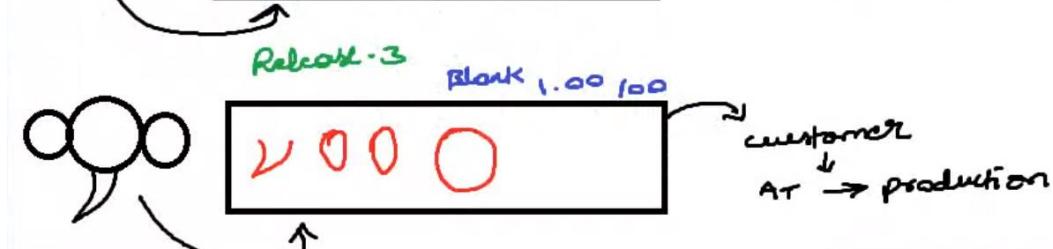
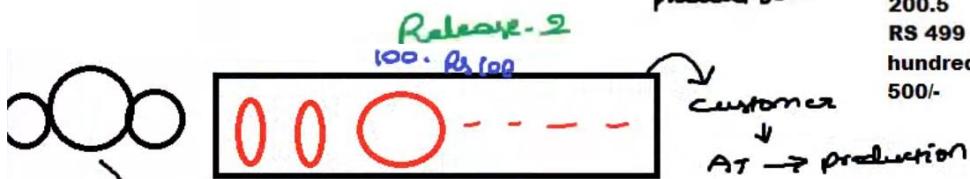
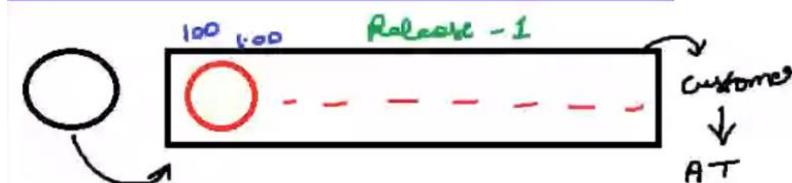
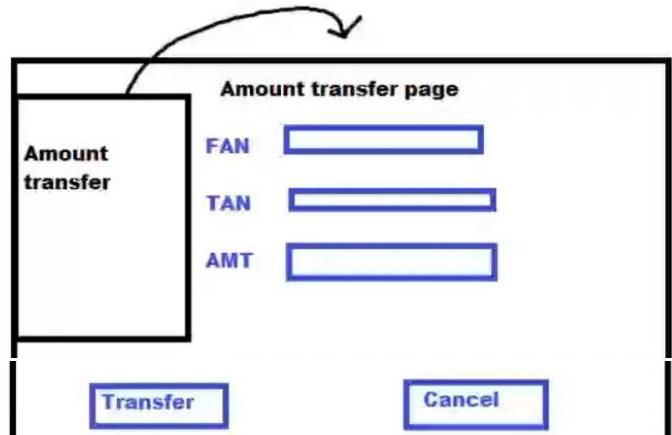
> _____

> _____

30.3. Amount text field

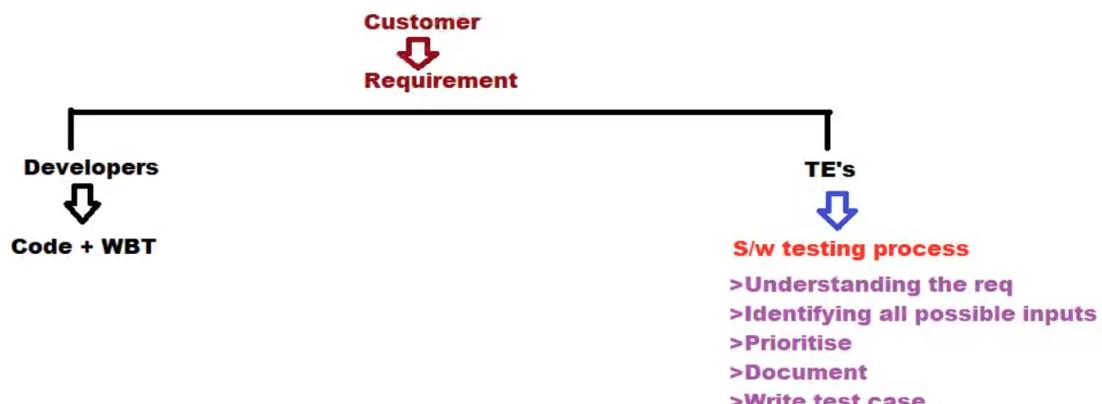
> It should accept +ve integer value between 100-5000

> It should not accept more than the balance.



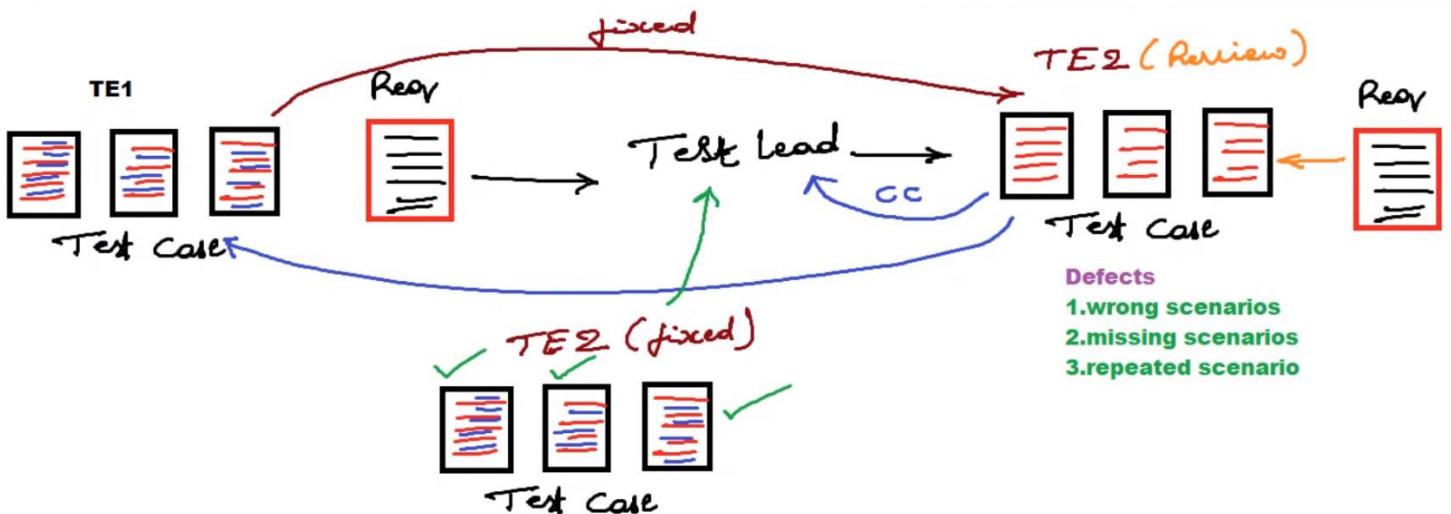
200	099	100.
99	\$100	
5001	10*10	
1	5000/2	
0	400+400	
-500	50.5*4	
200.5	6000-1000	
RS 499	blank	
hundred	1 00	
500/-	.0	

How to overcome the above drawbacks?

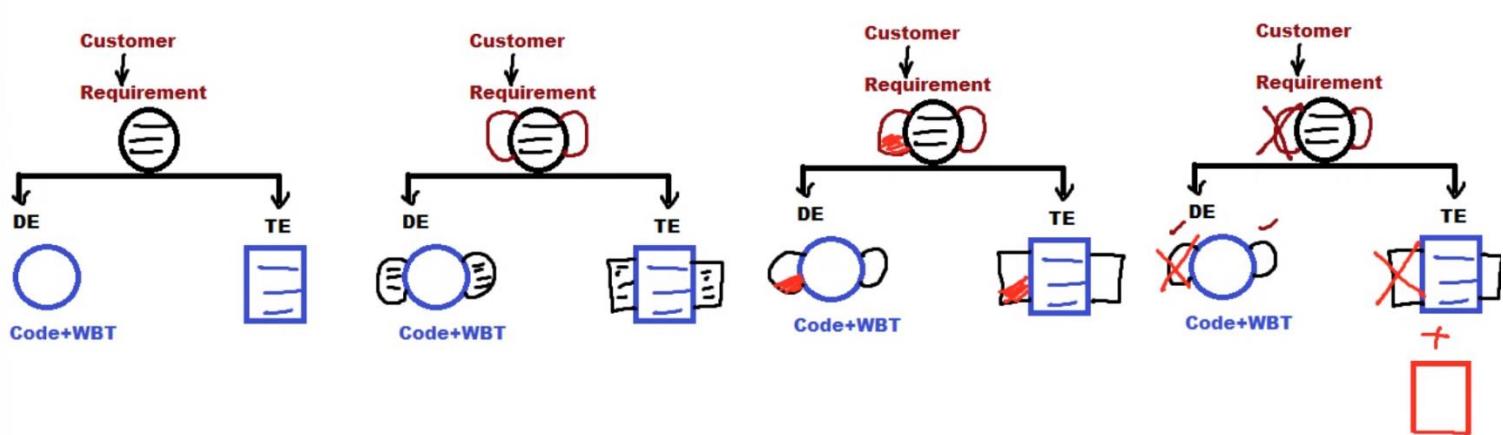


90% => Test management tool

10% => MS excel



When we write Test Case?



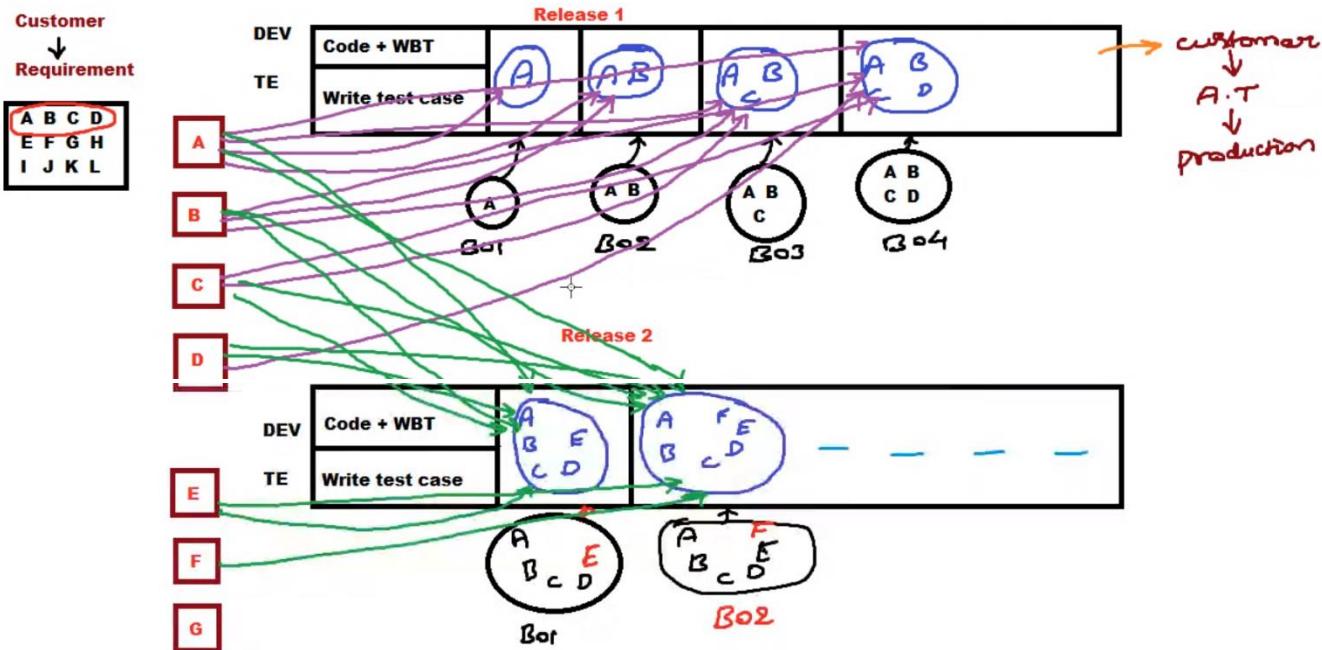
> When customer gives requirement developer are building the product, TE will be writing Test Case.

When developers give the product for testing, testing team will test the product based on test cases.

> When customer add requirement, developer will add the feature & TE will add the test case.

> When customer modifies the requirement developer will modify the feature & TE will modify the test case.

> When customer removes the requirement developer will remove the feature parallelly TE removes the test cases for respective feature & also write test cases to make sure that feature is really removed or not.



Why we write Test Case?

- > We write test case to have better test case coverage i.e., when requirement comes developer will be busy in building the product at the same time TE as free so they identify all possible scenarios & document it, when the build comes they can spend time in executing the scenarios, because of this no. of scenarios that you are going to cover will be more.
- > To have consistency in test execution i.e., if you have documented the scenarios you can make sure that you are executing all the scenarios in all the test cycle or sprint or release.
- > It depends on process rather than the person.
- > It avoids training every new TE about requirement of the product.
- > Test case acts like a proof for customer, development team & management team saying that you have covered all possible scenarios.
- > Test case acts like base document for writing automation script. If you refer test case & write automation script same kind of coverage we can expect even in automation.
- > If we have documented the test case no need of remembering the scenarios.
- > If we have documented the test case test execution happens in very organized way.
- > If we have documented the test case time taken to execute will be very less.

Header:

- > Test Case Name:
- > Requirement No.:
- > Test Data:
- > Pre-Condition:
- > Test Case Type:
- > Severity:
- > Brief Description:

Body:

Step No.	Action/Description	Input	Expected Result	Actual Result	Status	Comments
1	Enter alphabets into Amt text field & click on transfer button	ABCD	Approved error message should be displayed			
2	Enter negative integer into amount text field & click on transfer button	-100	Approved error message should be displayed			

Footer:

- > Author:
- > Reviewer:
- > Approved By:
- > Approved Date:

Test case design technique/Testing techniques/Design techniques/Black box testing technique

> It is a technique used while writing test case to imporve the test case coverage.

> Drawbacks of not following the techniques:

- > TE will miss lot of scenarios.
- > TE will miss lot of defects.
- > Test case coverage will not be good.

> Types of Design Techniques:

- > Error Guessing(EG)
- > Equivalence Class Partition(ECP)
- > Boundary Value Analysis(BVA)

Note: According to ISTQB there are 2 more techniques:

- > Decision Table Technique
- > State Transition Technique

> Error guessing:

- > Here we guess all possible errors & derive the scenarios, based on that.
- > We guess error based on expirience, intuition, requirement.
- > Ex: -100, 100Rs, 0, 1.0.0, 100/-, hundred, etc.

> Equivalence Class Partition:

- > Here there are 2 methods

> Pressman Rule:

- > Rule 1: If the input is in range of values then design the test case for 1 valid & 2 invalid values.

Amount → 100 – 5000

100

EG=> -200,2,20

ECP=> pressman rule=>rule 1 => 1 valid — 3000
=> 2 invalid— 1.01 , 5001

- > Rule 2: If the input is in set of values then design the test case for 1 valid & 2 invalid values.

https://amazon.com	
Search	<input type="text"/>
Printer	10
Mouse	20

10

EG=> 60, 50

ECP=> pressman rule => rule 2 => 1 valid —20
=> 2 invalid—1 ,00

- > Rule 3: If the input is in boolean then design the test case for both true & false values.

C a/c	<input checked="" type="radio"/> Male	<input type="radio"/> Female
	<input type="button" value="Submit"/> <input type="button" value="cancel"/>	

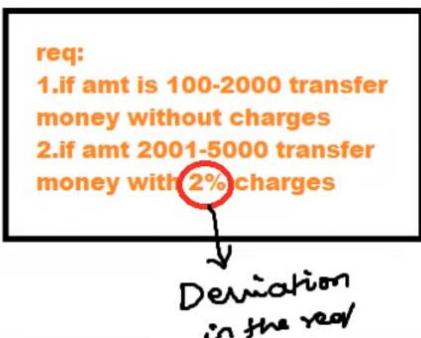
Drawback of pressman rule:

Amount → 100 - 5000

valid

EG

ECP->PR->rule 1 1 valid 2000 ✓
2 invalid 90,6000



20

```
if(amt>=100)&(amt<=2000)
{
    _____
    _____
}

Transfer money;
}
else if(amt>=2001)&(amt<=5000)
{
    _____
    _____
}

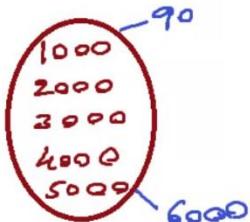
Transfer money;
}
else
{
    _____
    _____
}

error message;
```

> Practice Method:

> If the input is in range of values then divide the range into equivalent parts & test for all those values & atleast check for 2 invalid values, this is called as Practice Method.

Amount → 100 - 5000



```
if(amt>=100)&(amt<=5000)
{
    _____
    _____
}

Transfer money;
}
else
{
    _____
    _____
}

error message;
```

Note: 1) If there is deviation in between range of values then we go for Practice Method.

2) If there is no deviation in between range of values then we go for Pressman Rule.

3) By looking into requirement we will get to know whether there is deviation in between range of values or not.

> Boundary Value Analysis:

> If the input is in range of values between A to B then design the test case for A, A+1, A-1 & B, B+1, B-1.

Amount → 100 - 5000
A B

100	5000
100+1 = 101	5000+1 = 5001
100-1 = 99	5000-1 = 4999



```
if (amt > 100) & (amt < 5000)
{
    _____
    _____
}

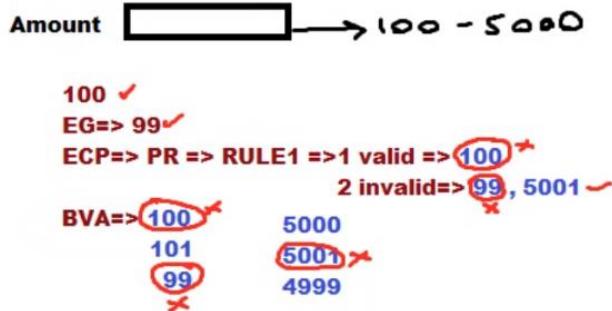
Transfer money;
}
else
{
    _____
    _____
}

if (amt <= 100) & (amt >= 5000)
```

> According to above example TE will enter 100 & click on transfer it will throw error message & it is a defect. The reason for this is developer has done mistake while writing conditional operators because of this usually we find more defects near the boundaries that is the reson TE should write test case by applying BVA.

> **Test Case Optimization:**

> The process of removing duplicate test case is called as **Test Case Optimization**.



Lesson 1 : Before we actually start writing test cases we should come up with options & select best option.

Lesson 2 : Always start writing test cases with navigational steps.

Lesson 3 : Always use should be / must be in expected result,dont use can be ,maybe,might be could be,would be etc.,,

Lesson 4 : Elaborate only those steps in which you have the focus don't elaborate all the steps unneccasirily.

Lesson 5 : Always write generic test cases dont write hardcoded test cases.

Lesson 6 : Whenever we are writing the test cases we should imagine the application.

Lesson 7 : If you organise the steps properly total number of steps will be reduced.

Lesson 8 : If something is covered in Functional Test case dont cover same in Integration Test Case.

If something is covered in Integration Test case dont cover same in System Test Case.

SL.NO	Description	Input	Expected Result	Actual Result	Status	Comments
1.	Open the browser & enter test url	https://<server name>cb.com	Welcome Page should be displayed.			
2.	Login as User A with valid UN & Password	UN: Pwd:	Home Page should be displayed			
3.	Click on Amount Transfer Link	NA	Amount Transfer Page should be displayed & following objects should be displayed. 1)FAN text field 2)TAN text field 3)Amount Text field 4)Transfer Button 5)Cancel button			
4.Following are the steps to test From Account Number Text Field :-						
4.1	Enter valid A/C No into FAN	2028589310	Error message should not be displayed	-	-	-
4.2	Enter alphabets into FAN T.F & click on transfer button	ABCDEFGHIJ	Error message should be displayed	-	-	-
4.3	Enter Special Characters with numbers into FAN & click on transfer button	@20285893178	Error message should be displayed	-	-	-
4.4	Enter Numbers with conditional operators into FAN Text field & click on transfer button	>20285893173	Error message should be displayed	-	-	-
4.5	-----					
4.6	-----					
4.10	Enter valid 10 digit A/C no into FAN text field & click on transfer button	2028589345	Error message should not be displayed	-	-	-
4.11	Enter invalid 9 digit invalid A/c no into FAN text field & click on transfer button	202898566	Error message should be displayed	-	-	-
4.12	Enter invalid 11 digit invalid A/c no into FAN text field & click on transfer button	20289856456	Error message should be displayed	-	-	-

5.0 Following are the steps to test TAN text field make sure that FAN text field is having valid data

5.1
5.2
5.3

5.13 Enter same A/C no which is entered for FAN text field into TAN text field & click on transfer button 2028589310 Error message should be displayed

6.0 Following are the steps to test Amount text field make sure that both FAN & TAN text fields is having valid data

6.1 Enter valid amount into amount text field & click on transfer button. 100 Confirmation page should be displayed
6.2 Click on browser back button NA Amount transfer page should be displayed
6.3 Enter decimal value into amount text field & click on transfer button 100.50 Error message should be displayed
6.4 Enter alphabets with numbers into amount text field & click on transfer button 100Rs Error message should be displayed
6.5 Enter -ve integer into amount text field & click on transfer button. -100 Error message should be displayed

6.11 Enter invalid amount into amount text field & click on transfer button 99 Error message should be displayed
6.12 Enter invalid amount into amount text field & click on transfer button 6000 Error message should be displayed
6.13 Enter valid amount into amount text field & click on transfer button 101 Amount transfer page should be displayed
6.14 Click on browser back button NA Amount transfer page should be displayed
6.15 Enter valid amount into amount text field & click on transfer button 4999 Confirmation page should be displayed
6.16 Click on browser back button NA Amount transfer page should be displayed
6.17 Enter valid amount into amount text field & click on transfer button 5000 Confirmation page should be displayed

Format - 2

SL.NO	Description	Expected Result	Status	Comments
1.	Open the browser & enter test url <a href="https://<server name>cb.com">https://<server name>cb.com	Welcome Page should be displayed.	-	-
2.	Login as User A with valid UN & Password	Home Page should be displayed	-	-
3.	Click on Amount Transfer Link	Amount Transfer Page should be displayed & following objects should be displayed. 1)FAN text field 2)TAN text field 3)Amount Text field 4)Transfer Button 5)Cancel button	-	-

4.Following are the steps to test From Account Number Text Field :-

4.1 Enter valid A/C No 2028589310 into FAN T.F & click on transfer button Error message should not be displayed.
4.2 Enter following invalid A/C no's into FAN text field & click on transfer button
1)ABCDEFGHIJ
2)@20285893178
3)>20285893173 Error message should be displayed

APPROACH TO WRITE FUNCTIONAL TEST CASE :-

- 1)Go to the body of test case
 - 2)Start with navigation steps
 - 3)Take first field(FAN)
 - a)Start with valid data
 - b)Write error guessing scenarios
 - c)Write Equivalence class partition scenarios
 - d)Write BVA scenarios
 - 4)Take second field(TAN)
 - a)Start with valid data
 - b)Write error guessing scenarios
 - c)Write Equivalence class partition scenarios
 - d)Write BVA scenarios
-

INTEGRATION TEST CASE

SL.NO	Description	Expected Result	Status	Comments
1.	Open the browser & enter test url <a href="https://<server name>cb.com">https://<server name>cb.com	Welcome Page should be displayed.		
2.	Login as User A with valid UN & Password	Home Page should be displayed		
3.	Click on Amount Transfer Link	Amount Transfer Page should be displayed & following objects should be displayed. 1)FAN text field 2)TAN text field 3)Amount Text field 4)Transfer Button 5)Cancel button		
4.	Enter valid A's A/C no into FAN T.F & B's A/C no into TAN & amount 1000 into amount text field & click on transfer button	Confirmation message should be displayed		
5.	Click on amount balance link	a)Amount balance Page should be displayed b)Balance amount should be equal to old balance - transferred amount		
6.	Click on transaction link	a)Transaction Page should be displayed b)Transferred amount,date,time,B's A/c no should be displayed.		
7.	Click on logout link	Welcome page should be displayed		
8.	Login as user B with valid UN & Pwd	Home page should be displayed		
9.	Click on amount balance link	a)Amount balance Page should be displayed b)Balance amount should be equal to old balance + transferred amount		
10.	Click on transaction link	a)Transaction Page should be displayed b)Transferred amount,date,time,A's A/c no should be displayed.		

APPROACH TO WRITE INTEGRATION TEST CASE :-

- 1)Take one feature,identify all possible scenarios.
 - 2)Prioritise the identified scenarios & document it.
 - 3)Go to body of the test case.
 - 4)Start with navigation steps.
 - 5)Cover scenario 1 ,scenario 2-----
-

How to fill Test case Header & Footer?

HEADER :-

1)Test Case Name :-

Format :- Project Name_Module Name_Scenarios

Example:- CB_Amount Transfer_Integration Scenarios

2)Requirement Number :-

BA when he convert CRS to SRS,in the SRS for every individual modules & components BA will mention requirement number.
Ex:-30.1 Amount Transfer

3)Test Data / Test Harness :-

It is the data which is written by the T.E & has to be done before executing step no 1.

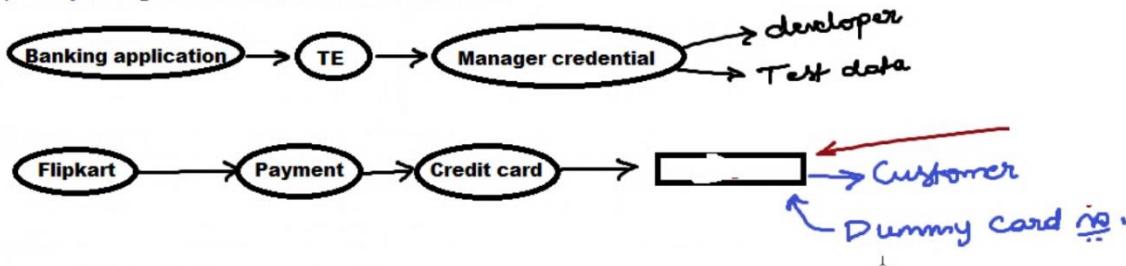
Ex:- TE should have UN,Pwd & A/C no of Users.

Test data can be created in two ways i.e

a)TE can create Test Data themselves



b)TE depending on someone else for Test Data.



IQ

How do you manage Test Data?

1)Few projects will create test data manually (If it is for smaller & simple applications)

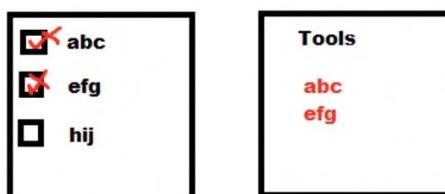
2)In many of the projects they write automation scripts,as soon as the build comes they run the scripts,
The script will create test data & sends mail to TE's.

4)Pre-Condition :-

It is the set of actions or settings which has to be done before executing step no 1.

Ex:-TE should have checked the balance of users.

While installing the software you should have selected some checkbox.



5)Test Case Type :-

Here TE will mention which test case he is writing.

Ex:-Functionality Test Case / Integration Test Case

6)Severity :-

TE will give severity for every individual test cases based on how important & complex the feature is

TE will always execute test cases based on severity.

There are 3 types of severity for test cases

1)Critical 2)Major 3)Minor

Example:-Gmail

Features	Test case	Severity	Execution Order
1>Login	Test case 1	Critical	T.C 1 ✓
2>Drafts	Test case 2	Major	T.C 5 ✓
3>Inbox	Test case 3	Critical	T.C 3 ✓
4>Trash	Test case 4	Minor	T.C 2 ✓
5>Compose	Test case 5	Critical	T.C 4 ✓

7) Brief Description :-

It describes about complete test case & the behaviour of the test case.

Ex:- In Amount Transfer module amount should accept positive integer & should not accept more than balance.

FOOTER :-

1) Author :- A person who writes test case.

Ex:-

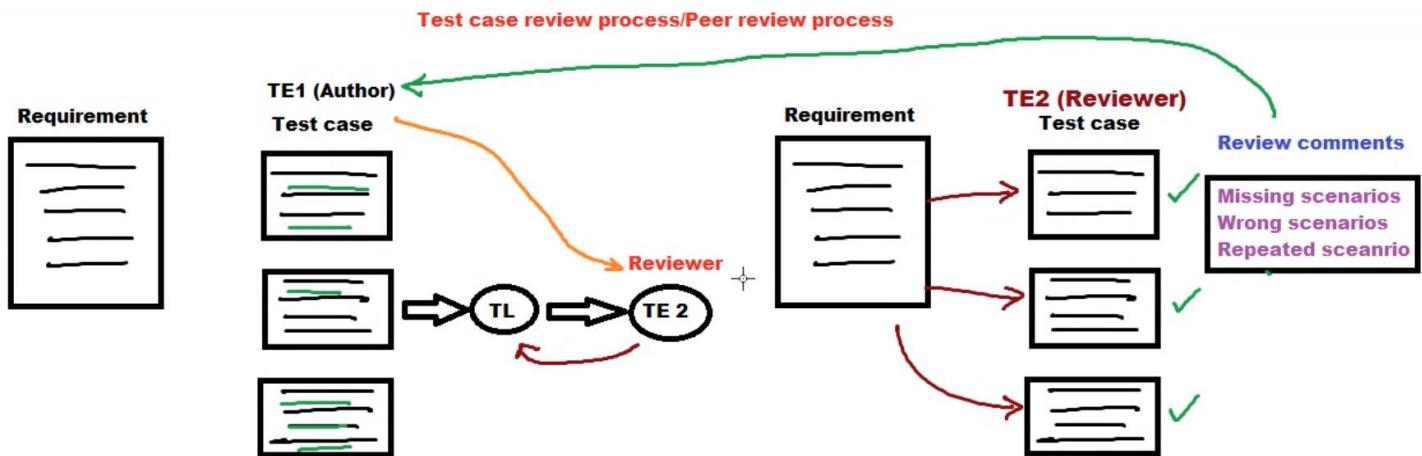
2) Reviewer:- A person one who reviews test case.

Ex:-

3) Approved By:-A person one who approves test case.

Ex:-

4) Approval date:- Here TE should write the date of approval.



Q. On what basis they assign test case for review?

They will assign to the person:-

- 1) Working on similar or related module in the project.
- 2) Worked on the same module in the previous project.
- 3) Working in the project since beginning & knows every corner of the product.
- 4) Who is responsible, who will understand the requirement very fast & identifies more mistakes.

Q. How do you ensure that reviewer does his job?

- 1) Assign primary & secondary reviewer
- 2) TL should also randomly review & find mistakes.
- 3) TL will intentionally introduce some mistakes & check whether it is found by the TE.

Q. Review Ethics :-

- 1) Always review the contents not the author.
- 2) Even after review if there are any mistakes, both author & reviewer will be responsible.
- 3) Reviewer should spend more time in finding the mistakes not in giving the solution.

Q. What will you do while reviewing test case? OR Why we review test case? (*****)

- 1) I will look into the header of the test case & understand the req for which test case is written & then go to body of the test case & try to find a) Missing scenarios b) Wrong scenarios c) Repeated scenarios
- 2) I will check whether organized or not, so that when executed it should take less time.
- 3) I will check whether it is simple to understand so that when given to new TE he should be able to execute without asking any questions.
- 4) I will look into the header of the test case & try to find :-
 - a) All the attributes are covered or not.
 - b) Check whether content in all the attributes are covered or not.
 - c) Check whether test case format is according to standard defined in the project

TEST CASE REVIEW TEMPLATE :-

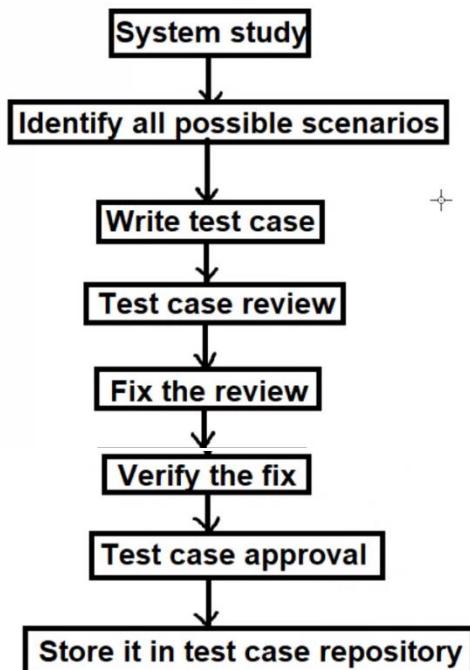
Test case review template will be prepared in test case management tool, MS Excel or MS Word.

Every TE should write review comments in Test case review template only.

Test case review template is not standard, it varies from company to company & project to project.

SI No	Test Case Name	Step No	Review Comments	Severity	Author Comments
1. CB_AT_FAN Text Field	Header Header	Requirement name is wrong	Major	FIXED	
		Test case type field is missing	Major	FIXED	
	10 20 50	Expected result is wrong	Critical	FIXED	
		Spelling mistake	Minor	NOT FIXED	
		Test case design technique not applied & coverage is not good.	Critical	FIXED	

Procedure to write testcase



1) SYSTEM STUDY :-

Read the requirement & understand the requirement, if you have any clarifications then you can interact with developer, business analyst & customer & get it clarified.

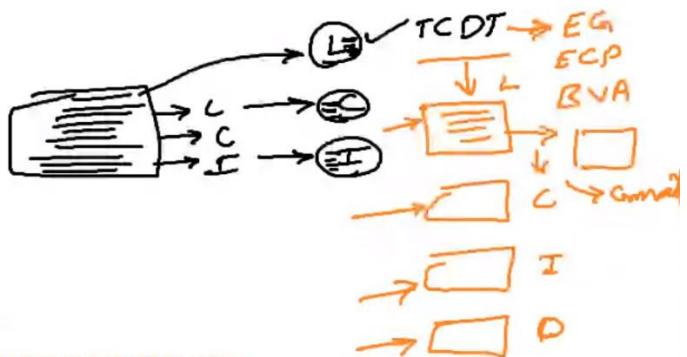
2) IDENTIFY ALL POSSIBLE SCENARIOS :-

1) Identify

2) Brain Storming session → Presentation

- a) Requirement/Features → Improves Product Knowledge
- b) Scenarios → Missing, Wrong & Repeated scenarios.

3) Measure the efficiency of Brain Storming session.



3) WRITE TEST CASE :-

- a) Group the related scenarios.
- b) Prioritise the scenarios within each group.
- c) Apply test case design technique.
- d) Use standard test case format.
- e) Document it.

4) STORE IT IN TEST CASE REPOSITORY :-

Test case Repository :- It is a centralised place where we store all the test cases.

Test Suite :- It is the collection of related test cases.

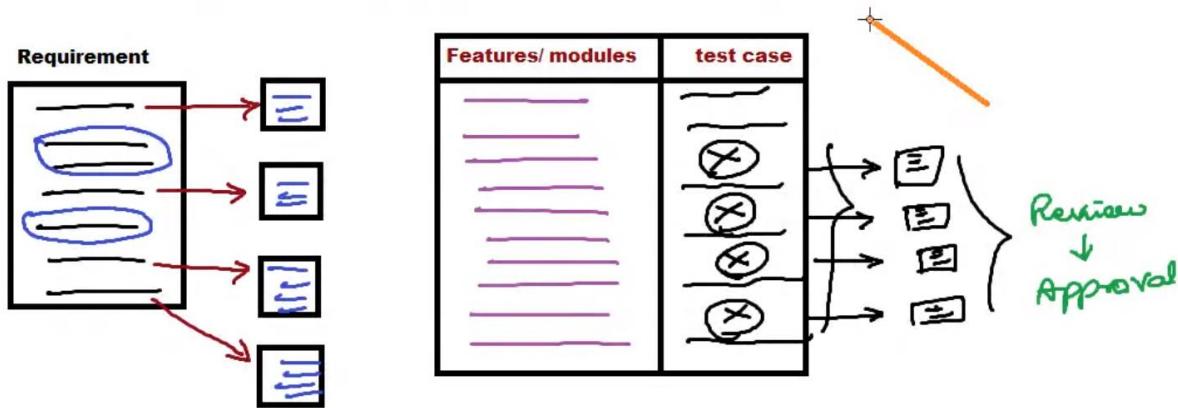
4)STORE IT IN TEST CASE REPOSITORY :-

Test case Repository :- It is a centralised place where we store all the test cases.

Test Suite :- It is the collection of related test cases.

TRACEABILITY MATRIX / REQUIREMENT TRACEABILITY MATRIX(RTM) / CROSS REFERENCE MATRIX (CRM)

It is the document which we prepare to make sure that every requirement has atleast one test case.



Sl No	Module Name	High level requirement	Detailed Requirement	Test Case Name
1.	Loan	1.1 Personal Loan 1.2 Home Loan	1.1.1 _____ 1.1.2 _____ 1.2.1 _____ 1.2.2 _____	CB_PL_Approval CB_PL_Rejection CB_HL_Approval CB_HL_Rejection
2.	Amount Transfer	2.1 FAN text field 2.2 TAN text field 2.3 Amount text field b/w 100-5000	2.1.1 should accept only 10 digit numbers 2.1.2 should accept only those numbers which is created by manager. 2.2.1 _____ 2.2.2 _____ 2.3.1 should accept only positive integer	

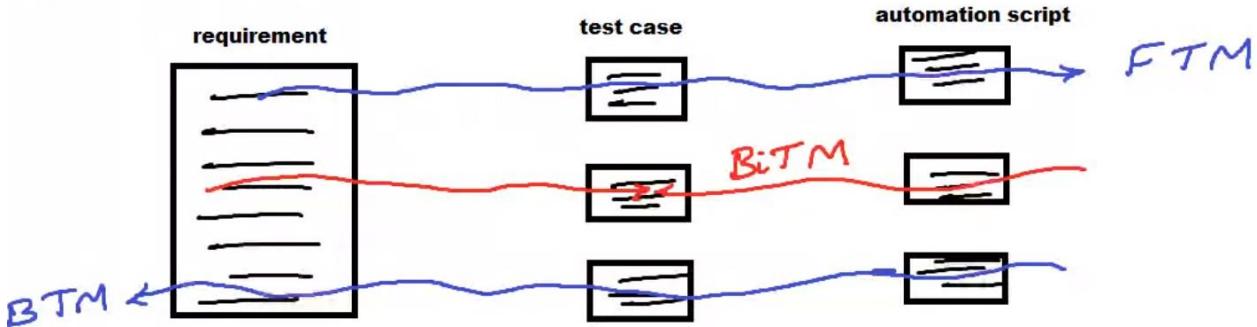
Advantages :-

- 1)It ensures that every requirement has got atleast one test case,which indirectly assures you that you are testing every feature atleast once.
- 2)It gives traceability from high level requirement till automation script name

Drawbacks :-

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

Types Of Traceability Matrix :-



1)Forward Traceability Matrix : Mapping from route document to derived document is called as F.T.M

Ex:-Mapping from requirement to test case & from test case to automation scripts.

2)Backward Traceability Matrix : Mapping from derived document to route document is called as B.T.M

Ex:-Mapping from automation scripts to test case & from test case to requirement.

3)Bi-Directional Traceability Matrix :If yo do both FTM & BTM then it is called as Bi-Directional Traceability Matrix.

Interview:-

1)If there is no requirement then how will you prepare traceability matrix?

Explore the application, understand the application, identify the scenarios, document it, write test cases. Once after writing, what is the proof that you have written test case for all the features, to ensure that we write traceability matrix.

Here we list all the features that are there in the product & map it to test cases.

Features	Test case Name
Login	-----
Forgot Password	-----
Remember Password	-----
Compose	-----
Attachment	-----
Inbox	-----
Sent Item	-----

What is the difference b/w traceability matrix & test case review?

Traceability Matrix

Test Case Review

1) Here we check every requirement has got atleast one test case.

2) Here we dont check whether every requirement has got all possible scenarios.

1) Here we check whether all possible scenarios are covered for a specific requirement.

2) Here we dont check whether every requirement has got atleast one test case.

Assume that last day when you are about to release the product to the customer, if there is a critical/blocker defect what will you do? Will you release product to the customer or not?

As a TE i am not decision maker, so i will list all the defects which are pending & the impact of the defect on the business & also i suggest whether to release or not & send it to PM.

It is the manager who should take a calculated decision & decide.

How do you ensure that your test coverage is good? OR How do you convince your customer/management/developer that you have tested everything?

My test coverage is good because my test cases are good, my test cases are good because i have followed a strict procedure.

Strict procedure means:-

1) I did a thorough system study because of which i had a deeper knowledge of the product, because of that i was able to find more number of scenarios.

2) I identified all possible scenarios then i did a brainstorming session, because of which i was able to find more number of scenarios.

3) While writing test case i applied test case design technique because of that there was an improvement in the coverage.

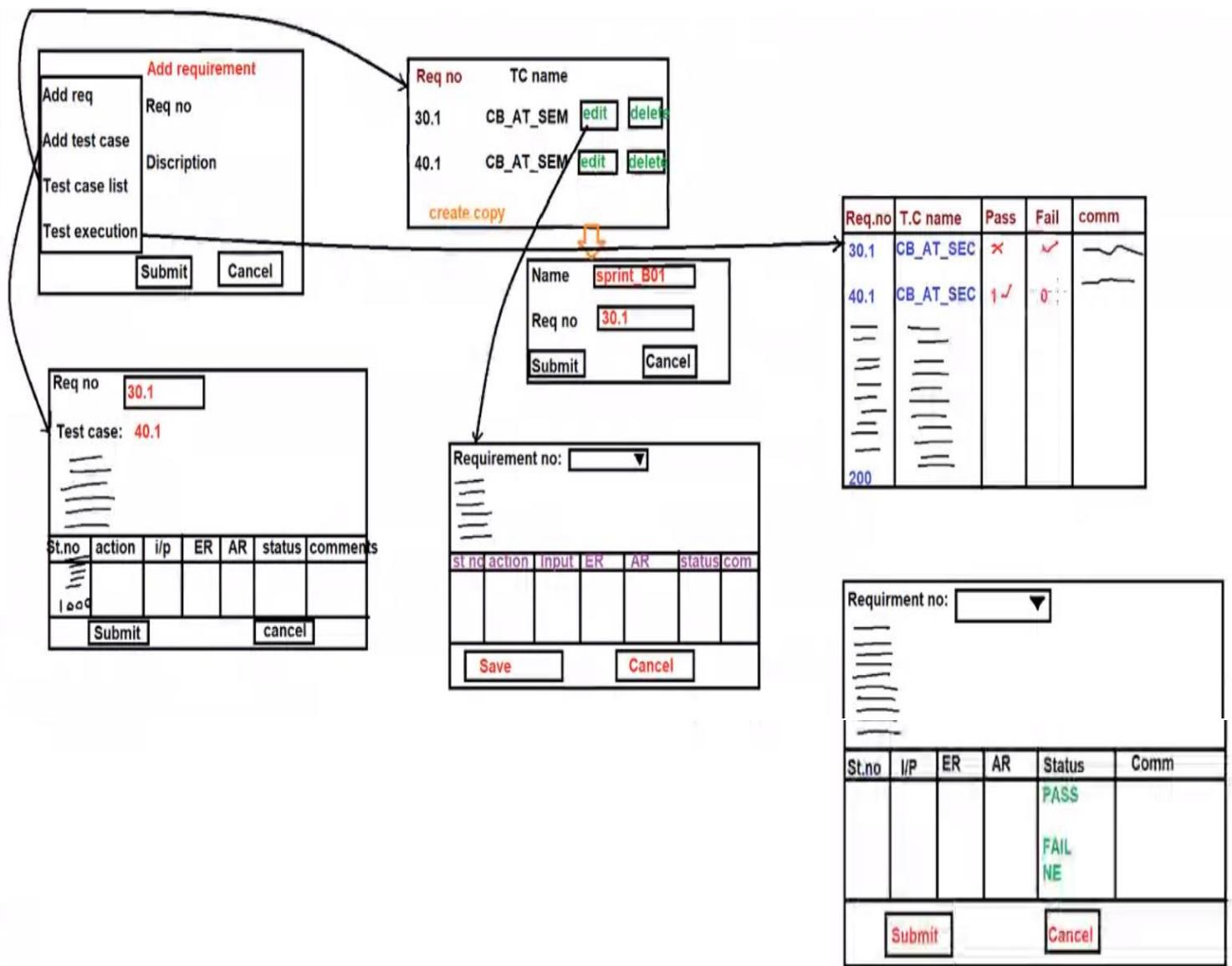
4) I got all my test case reviewed at that stage only, we got some more missing scenarios & because of that there was an improvement in the coverage.

5) While executing test cases i identified few new scenarios, i added them back into the test cases because of this there was an improvement in the coverage.

6) Also i performed Adhoc Testing because of that there was an improvement in the coverage.

7) I wrote traceability matrix & ensured that every requirement has got atleast one test case because of that there was an improvement in the coverage.

Test Management Tool



Globalisation testing

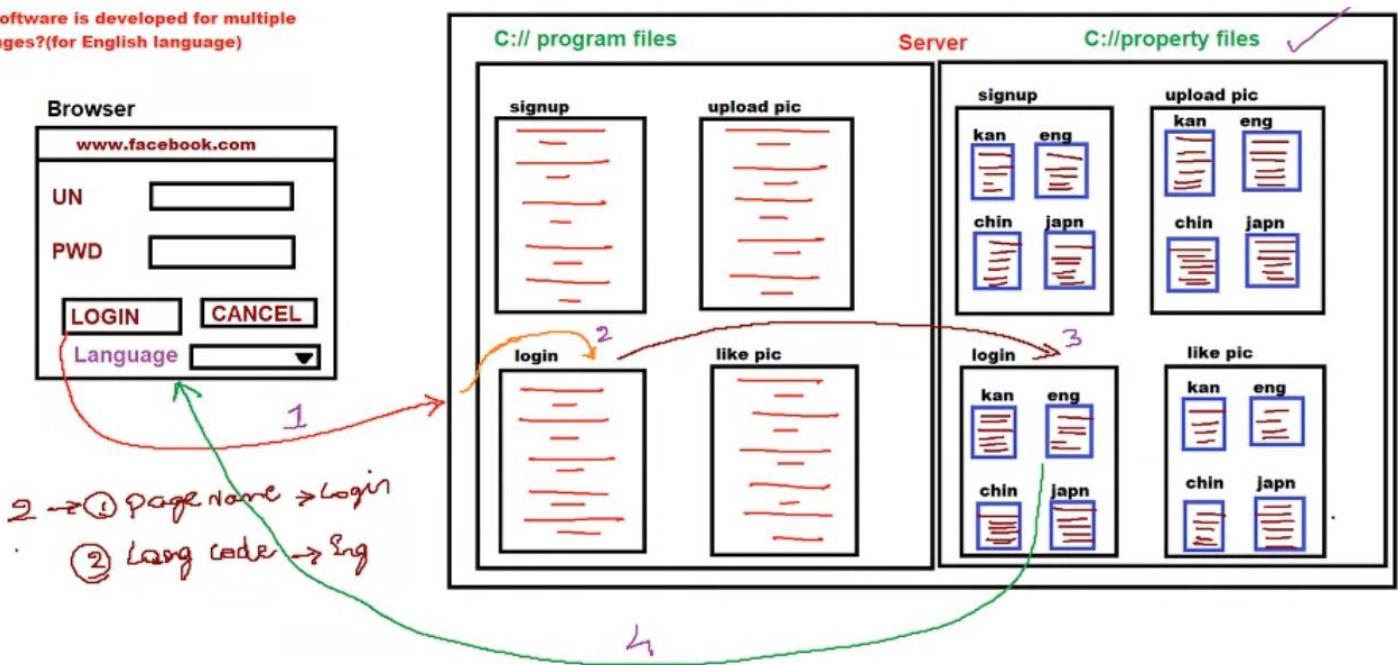
Developing the software for multiple languages is called Globalisation.

Testing the software which is developed for multiple languages is called as Globalisation Testing.

Types of Globalisation Testing:

- > Internationalisation Testing or I18N Testing
- > Localisation Testing or L10N Testing

How software is developed for multiple languages?(for English language)



Step 1: User will open browser & enter url, when he click 'go' request goes to server, request object contains 2 things 'Page Name' & 'Language Code'.

In this example Page Name:'Login' & Language Code:'English'

Step 2: Server is waiting for request as soon as request object comes in it goes inside & checks for page name.

In this example page name is 'Login' so it connects to login.jsp

Step 3: The program starts running & it has got 2 sections. The first section of the program receives the request object, goes inside it & checks for language code.

In this example language code is 'English' so it connects to english property file.

The connection has been established between property file & program file.

Step 4: The second section of the program copies the content from the property file & display it on the browser.

What is I18N Testing?

- > Testing the software which is developed for multiple languages is called as Internationalisation Testing.
- > Here we check whether content is in right language or not.
- > Here we check whether right content is displayed in right place or not.
- > Here we check whether features are broken with languages changed.

How to do I18N Testing?(for Chinese language)

- > Go to chinese property file.
- > Add prefix & suffix to the content.
- > Save the property file.
- > Open the application select the language (Chinese Language) corresponding page will be displayed.
- > I will check for the prefix if it is correct means content is in right language.
- > I will check for the suffix if it is correct means right content is displayed in right place.

Note: TE for their understanding purpose in order to test the software we give prefix & suffix for different languages & contents. This concept is called as Psuedo Translation.

What kind of defects we can find while doing I18N Testing?

- > Chances are there right language might not be displayed.
Ex: in gmail if we select language as kannada or telugu some contents are still displayed in english.
- > Chances are there right content might not be displayed in right place.
- > Alignment specification problem for the languages.
- > Tool Tip defect.

What is Tool Tip?

> When user points cursor on the image rectangular box will be displayed which explain about the image this is called as Tool Tip.

Tool Tip should be always displayed with the language which user has selected, if not it will be called as Tool Tip defect.

Localisation Testing or L10N Testing:

Testing the software to check whether software is developed according to country standards or country culture is called as Localisation Testing.

> Here we check for the currency/price format is displayed as per country standards or not.

Ex: India => Rs/Inr/

US => \$/USD

UK => Pound

Germany => Euro/

> Here we check whether date format is displayed as per country standards or not.

Ex: India => DD/MM/YYYY

US => MM/DD/YYYY

Germany => YYYY/MM/DD

> Here we check whether pincode format is displayed as per country standards or not.

Ex: India => postal code => 6 digit

US => Zip code => 5 digit

Germany => 5/3 digit

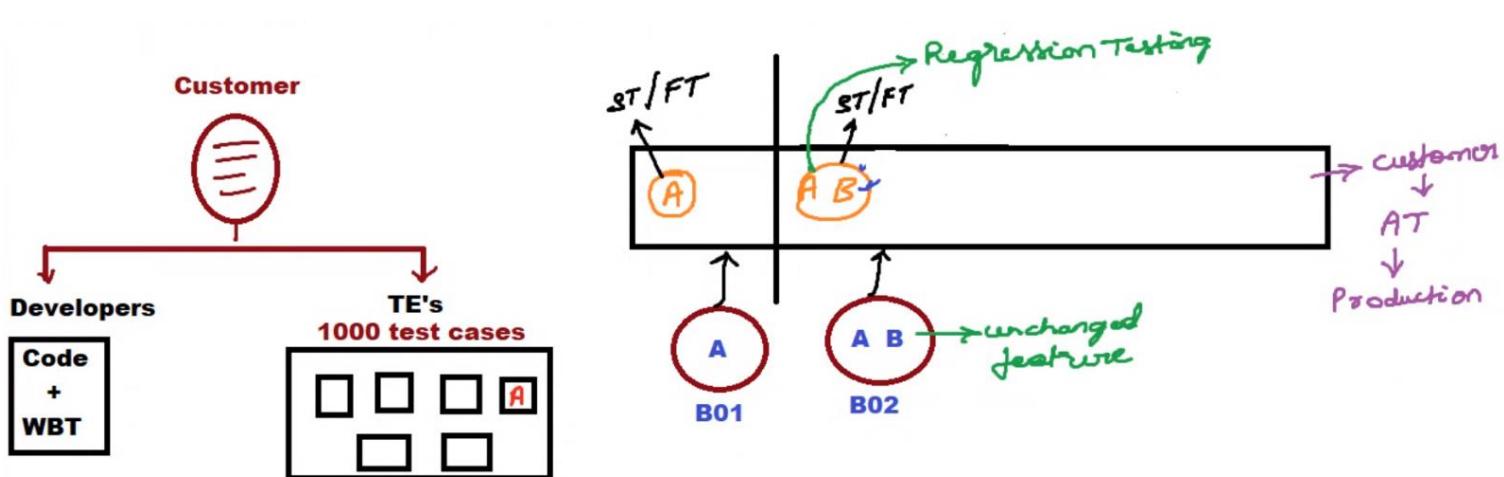
> Here we check whether the colour of the image is displayed as per country standards or not.

Regression testing

1. Testing the unchanged features to make sure that it is not broken or affected because of changes is called as Regression Testing.

Here changes can be addition, modification or removal of features & also it can be bug fixes.

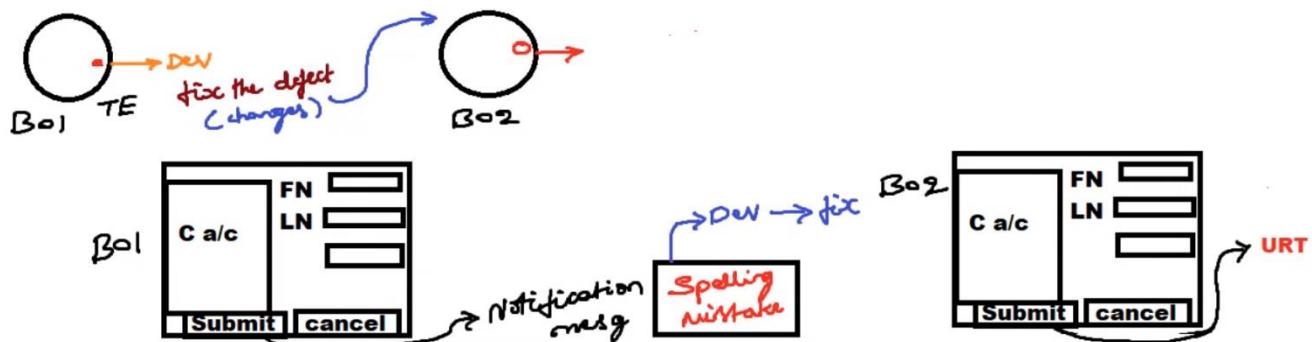
2. Re-execution of same test cases in different test cycles or release or sprint to make sure that changes are not introducing any defects in unchanged features is called as Regression Testing.



Types of Regression Testing:

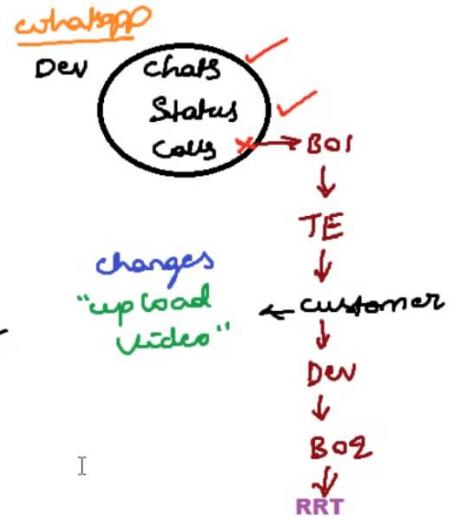
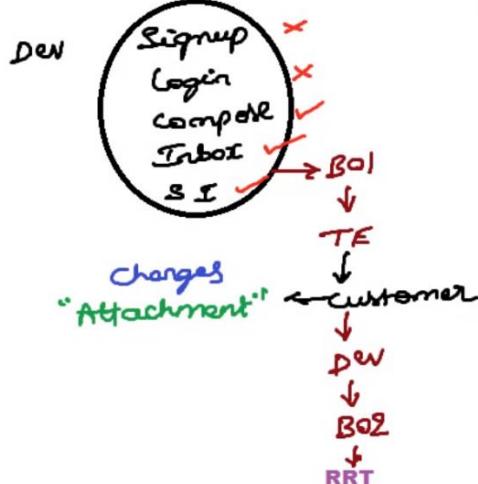
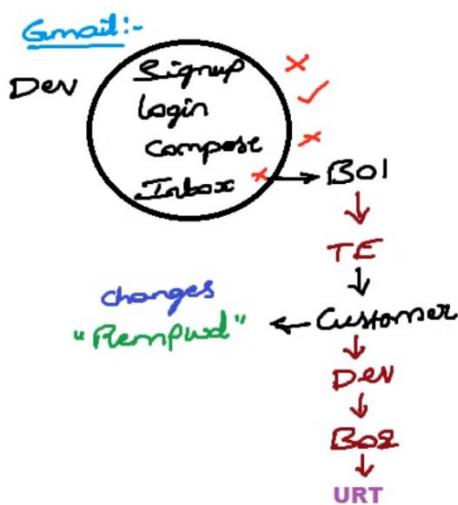
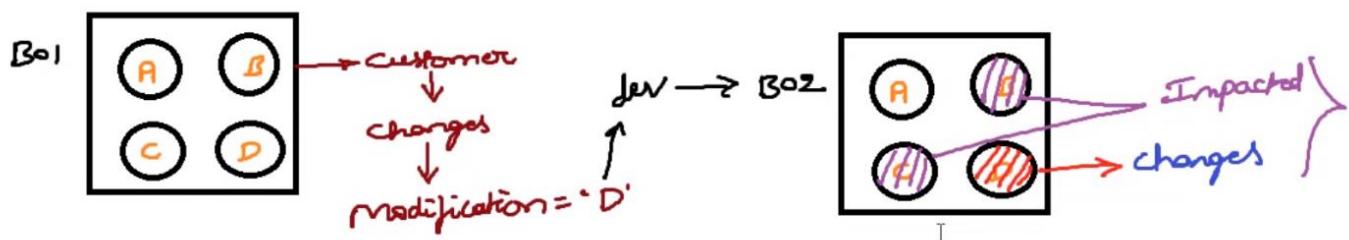
> Unit Regression Testing:

> Testing only the changes which is done or the bug which is fixed is called as Unit Regression Testing.



> Regional Regression Testing:

> Testing the changes & impacted region(affected areas) is called as Regional Regression Testing.



Q) What is impact analysis meeting?

> TE will interact with Sr.TE, TL, Sr.Dev, BA & Customer sometimes & discuss about the impacted areas & document the impacted areas, this is called impact analysis meeting.

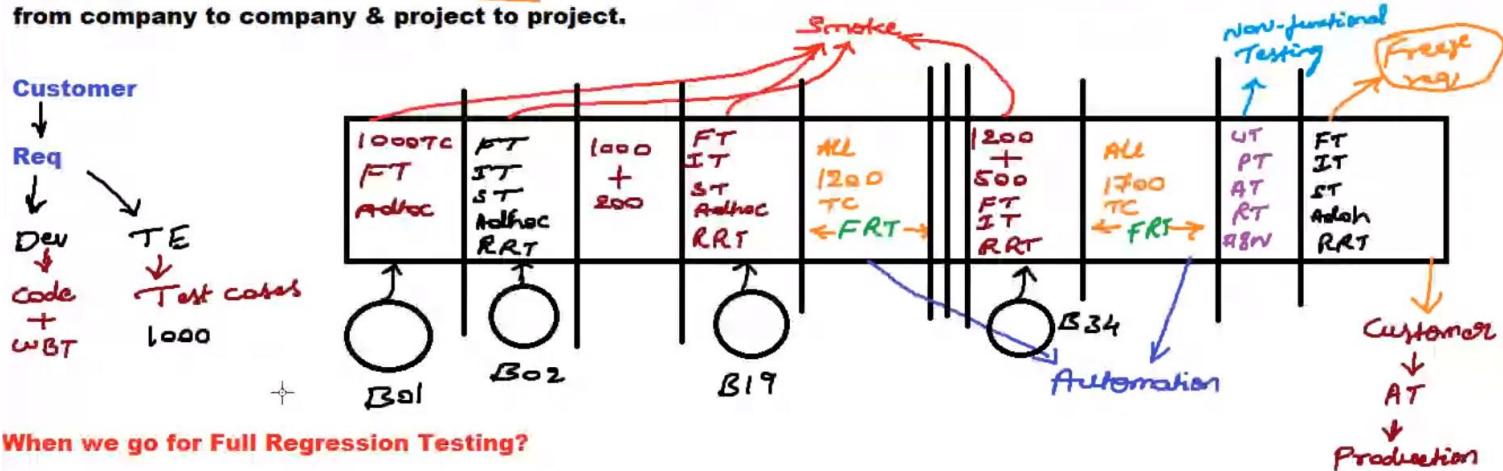
Q) How do you identify impacted areas?

- > Customer when he gives requirement he will only inform to developer & TE about the impacted areas.
- > BA when he converts CRS to SRS in the SRS there will be a section called impacted areas there BA will mention the list of impacted areas.
- > Developer will give the information about impacted areas because he is the person who is doing code changes & he will be aware of which all the areas code has been modified.
- > TE will interact with TL/TM/Sr.TE/BA, based on their product knowledge they will give list of impacted areas.
- > TE will conduct impact analysis meeting.

> Full Regression Testing:

> Testing the changes & all the remaining features of an application is called as Full Regression Testing.

Note: Some companies after every 10-12 cycles they go for full regression testing but it is not standard, it varies from company to company & project to project.



When we go for Full Regression Testing?

- > When the changes are more don't spend time in doing impact analysis then test the entire project by doing full regression testing.
- > When changes are done in the root of the product, test the entire product by doing full regression testing.

When we go for regression testing?

- > When changes are done to the s/w or when s/w is changed.
- > When the environment or platform is changed.

IQ) What is the difference between Regression Testing & Re-testing?

Re-Testing(Confirmation Testing) | Regression Testing

- | | |
|---------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------|
| > Whenever developer give the build check whether defect is fixed or not is called as Re-Testing. | > Testing the unchanged features to make sure that it is not affected or broken because of changes. |
| > Re-testing is done for failed test cases. | > Regression is done for passed test cases. |
| > Re-testing is planned. | > Regression is generic. |
| > Here we don't go for automation. | > Here we go for automation. |

When or Why we go for automation?

- > When the product is functionally stable.
- > When there are no blocker or critical defects.
- > When there more no. of regression test cases.
- > When there are no major requirement changes done by the customer.
- > Once after product is tested manually for 10-15 cycles we can go for automation.
- > Once after s/w is manually tested for 1 or 2 release.

Test plan

It is a document which drives future testing activities. It has got different sections like

- > Objective
- > Effort Estimation
- > Scope
- > Approach
- > Assumptions
- > Risk
- > Mitigation Plan / Contingency Plan / Backup Plan
- > Test Methodology
- > Test Schedule
- > Test Environment
- > Test Automation
- > Defect Tracking
- > Deliverables
- > Test Stop Criteria
- > Entry & Exit Criteria
- > Roles & Responsibilities
- > Templates

Objective: This section covers aim of preparing the test plan.

Effort Estimation: This section covers the estimation of how long it would take to complete the task also how many engineers needed to complete the task & cost of testing.

Scope: This section covers what are all the features to be tested & what are all the features not to be tested.

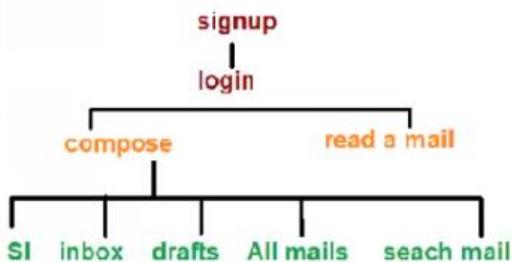
login compose Inbox	Help feedback support
---------------------------	-----------------------------

Approach: This section covers how we go about testing the product in future.

4.1: By writing high level scenarios

- > login to gmail click on compose fill all the details, saved & check in SI
- > Click on delete a mail & check it in trash

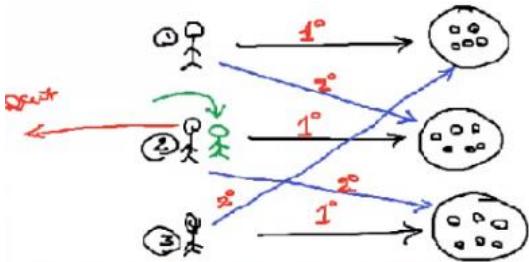
4.2: By writing flow graph



Assumptions: This section covers assumptions made while we are planning.

Risk: If any assumption fail that becomes risk.

Mitigation Plan: This section covers how do we face the risk.



> We are planning means we are assuming that we are going to do lot of task, based on that we promise the customer, because of few reasons if we are not able to do certain task it becomes risk. To face that risk we have a plan that is nothing but mitigation plan.

Ex 1: We assume that everybody will be there in the project.

Ex 2: If someone quits new fellow might miss lot of defects, that becomes risk.

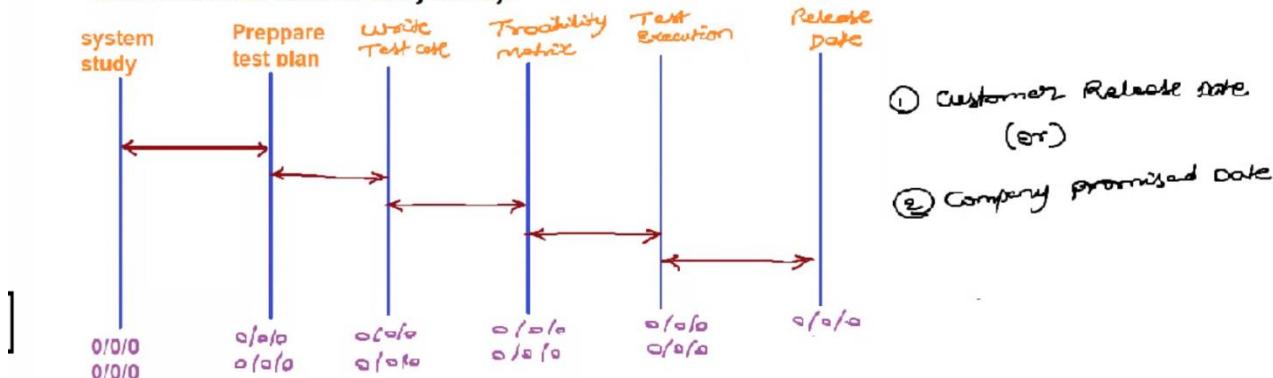
Ex 3: The mitigation plan is new engineer should refer test cases & test the product based on secondary owner help.

Test Methodology: This section covers what are all the types of testing that we are planning to conduct in future.

ST	CT
FT	AT
IT	RT
ST	UT
Adhoc	PT
Regression T	I18N
	L10N

Test Schedule: This section covers in future when exactly which activity should start & when exactly which activity should end.

Note: Here we fix dates for every activity.



Test Environment: This section covers how we go about setting up the test environment in future.

10.1 Procedure to install build

10.2 Hardware

- 10.2.1 Server Side
HP starcat 1500

10.2.2 Client Side

- 6 computers with following configuration
- 8GB
- 1TB
- motherboard
- Quad Core Processor

10.3 Software

10.3.1 Server Side

OS -> Win 10
Webserver -> Tomcat
Appserver -> Websphere
DBserver -> SQL

10.3.2 Client Side

OS -> Win7, Win8, Win 10, Win XP, etc
Browser -> Mozilla, Chrome, UC, IE, Edge, Safari, etc

Test Automation: This section covers what are all the features to be automated & what are all the features not to be automated & also complete automation strategy.

11.1 Features to be automated

Login, Compose, Inbox, Sl,....

11.2 Features not to be automated

Help, Calender, Contacts, FAQ,....

11.3 Automation Framework

Django

11.4 Automation Tool

Selenium



Defect Tracking: This section covers in future when we find defect how we should track it, it also covers what should be the

- a) procedure
- b) status
- c) severity
- d) priority levels

12.1 Procedure to track

12.2 Severity

Blocker
Critical
Major
Minor

12.3 Priority

P1
P2
P3

12.4 D.T.Tool

Bugzilla

Deliverables: This section covers what are all the outputs or document that has to be given by the testing team at the end of test life cycle.

14.1 Test Case

14.2 Test Plan

14.3 Tracability Matrix

14.5 Test Execution Report

14.6 Release Note

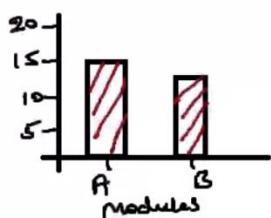
14.7 Graphs & Matricis

Release Note:

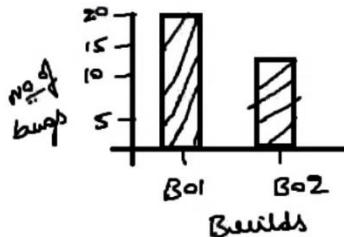
- > Along with the product we release 1 note to customer, that is called release note.
- > It consists of
- > List of open defects / known defects still that are there in the product.
- > List of the platform in which product is tested.
- > List of platform in which the product is not tested.
- > List of new features, added, modified / removed.
- > No. of defects found in previous release & fixed in current release.
- > Installation steps & version no.

Graphs:

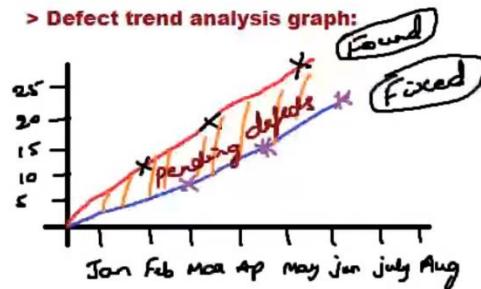
> Defect distribution graph:



> Build wise defect distribution graph:



> Defect trend analysis graph:

**Matrixis:**

> It is a past data which is gathered between 2 parameters(test engineer & no. of defects), because of which we can do proper analysis & take decision for future testing activities.

> Defect distribution matrixis:

Module	critical	major	minor
A	30	60	90
B	15	30	45

> Build wise defect distribution matrixis:

Build	critical	major	minor
B01	40	80	200
B02	30	60	90

> Test efficiency matrixis / Productivity matrixis:

Efficiency = $\frac{\text{No. of defects found by TE}}{\text{No. of defects found by TE} + \text{No. of defects found by customer} + \text{No. of defects found by enduser}}$

X 100

$$\frac{\text{No. of defects found by TE}}{\text{No. of defects found by TE} + \text{No. of defects found by customer} + \text{No. of defects found by enduser}} \times 100$$

TE name	No. of defects found by TE	No. of defects found by customer	No. of defects found by endusers	Efficiency
Puneeth	60	10	5	$\frac{60}{60+10+5} \times 100 = 80\%$ 75
Sunny	20	35	30	$\frac{20}{20+35+30} \times 100 = 23.5\%$ 85

Test Stop Criteria: This section covers when exactly we should stop testing.**Q) What is the test stop criteria or When do we stop testing?**

> We stop testing in 2 situations i.e., product quality should be good or it should be bad.

> Good means:

> All the features requested by the customer should be ready.

> All the end - end business scenarios should work fine.

> There should be 0-blocker & 0-critical defects & if there are few bugs left out it should be within the limit set by the customer.

> You should have tested the product in the environment similar to production environment.

> Bad means:

> If there are too many blocker & critical defects.

> If many end - end business scenarios are not working fine.

> If it is crossing the budget & also schedule.

Entry & Exit Criteria:

- > **Entry Criteria:** It is the list of criteria that should be met to start the activity.
- > **Exit Criteria:** It is the list of criteria that should be met to say that activity is over.

> Entry Criteria for FT:

- 1. WBT should be done.
- 2. Build should be installed properly.
- 3. Test case should be there.
- 4. Test data should be available.
- 5. Resources should be available.

> Exit Criteria for FT:

- 1. Percentage of test case executed should be 90%.
- 2. Percentage of test case pass should be 85%.
- 3. It should not cross more than 10 critical defects.
- 4. It should not cross more than 40 major defects.
- 5. It should not cross more than 80 minor defects.

> Entry Criteria for IT:

- 1. It should be met with exit criteria of FT.
- 2. WBT should be done.
- 3. Build should be installed properly.
- 4. Test case should be there.
- 5. Test data should be available.
- 6. Resources should be available.

> Exit criteria for IT:

- 1. Percentage of test case executed should be 95%.
- 2. Percentage of test case pass should be 90%.
- 3. It should not cross more than 5 critical defects.
- 4. It should not cross more than 30 major defects.
- 5. It should not cross more than 60 minor defects.

Roles & Responsibilities: This section covers what each engineer should do in different stages of test life cycle.**> Roles & Responsibilities of TM:**

- 1. He will write & review test plan.
- 2. He will interact with test team, development team, PM & also customer sometimes.
- 3. Handels issues & escalations.
- 4. He will approve release date.

> Roles & Responsibilities of TL:

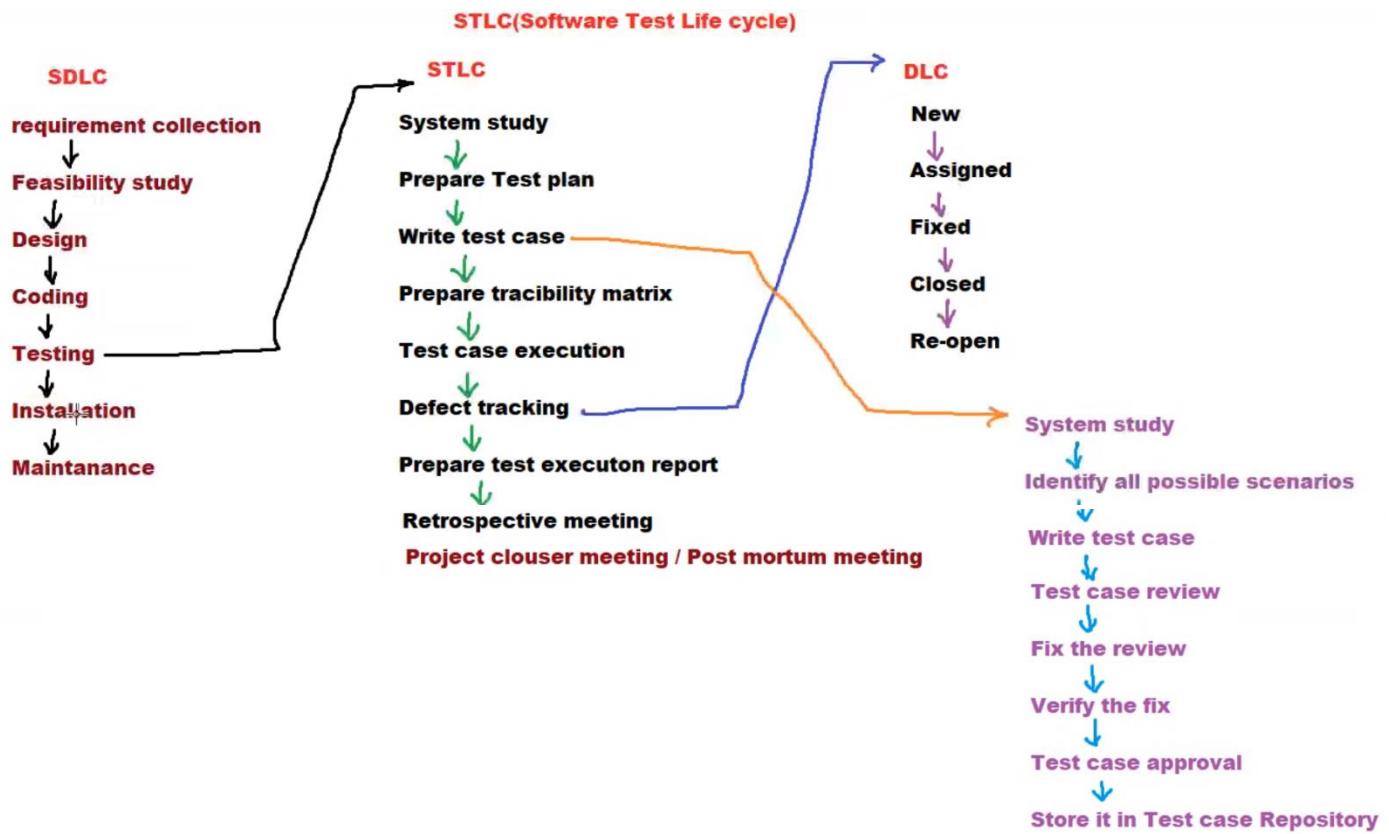
- 1. He will write & review test plan.
- 2. He will allocate work to TE & make sure that they are going to complete the task within the schedule.
- 3. He will consolidate the reports which are sent by every TE & communicate with test team, development team, PM.
- 4. He will conduct impact analysis meeting.

> Roles & Responsibilities of TE:

- 1. He will review test plan.
- 2. He will write test case for his features.
- 3. He will review test cases.
- 4. He will conduct all types of testing.

Templates: This section covers format for all the documents that you are planning to prepare in the entire test life cycle.

- 17.1 Test Case
- 17.2 Defect Report
- 17.3 Test Plan
- 17.4 Tracability Matrix
- 17.5 Test Execution Report
- 17.6 Tets Case Review



It is step by step procedure or standard procedure to test any software.
It is part of SDLC.

What is STLC (or) What procedure you will follow if I will software to test (or) Stages of STLC?

> It is the step by step procedure to test any software.

> **System Study:**

It means here we read the requirement & try to understand the requirement. If we have any queries interact with developer, BA or customer & get it clarified.

> **Prepare Test Plan:**

Once we understand the requirement we should prepare test plan. Test plan is a document which drives all future testing activities. This is where we decide

- 1) How many engineers are needed to complete the testing.
- 2) What each engineer should do in each stages of testing.
- 3) What are the types of testing that we have to conduct in future.
- 4) What are the features to be tested & not to be tested.
- 5) What should be the testing approach.
- 6) When exactly we should start any activity & end the activity.

> **Write Test Case:**

Once planning job is done we start writing the test case.

This activity has got different stages like

- 1) System Study
- 2) Identify all possible scenarios
- 3) Write test case
- 4) Test case review
- 5) Fix the review comments
- 6) Verify the fix
- 7) Test Case Approval
- 8) Store it in test case repository

> Prepare Tracability Matrix:

Once test cases are ready what is the proof that we have written test case for all the modules. To ensure that we prepare tracability matrix.

Tracability Matrix is a document which we prepare to make sure that every requirement has got atleast 1 test case.

> Test Execution:

This is the stage wherein we test the product for 3-7 cycles if it is agile & 40-60 cycles if it is non-agile.

This is where we conduct all types of testing & find defects, also help the product to improve the quality.

This is where TE are productive to the organisation.

> Defect Tracking:

We are executing the test means definitely we are going to catch lot of defects. Every defect that we are trying to catch should be tracked in a very organised way i.e., called defect tracking.

> Test Execution Report:

At the end of every test cycle we prepare test execution report.

It is a document which covers

- 1) How many test cases are there
- 2) How many test cases are executed
- 3) How many test cases are not executed
- 4) How many test cases are pass
- 5) How many test cases are fail
- 6) What is the passing percentage
- 7) What is the fail percentage

We send this report to the customer in last test cycle & that is the end of the project or release

from customer point of view. But from project

point of view we have got 1 more activity i.e., retrospective meeting.

> Retrospective Meeting:

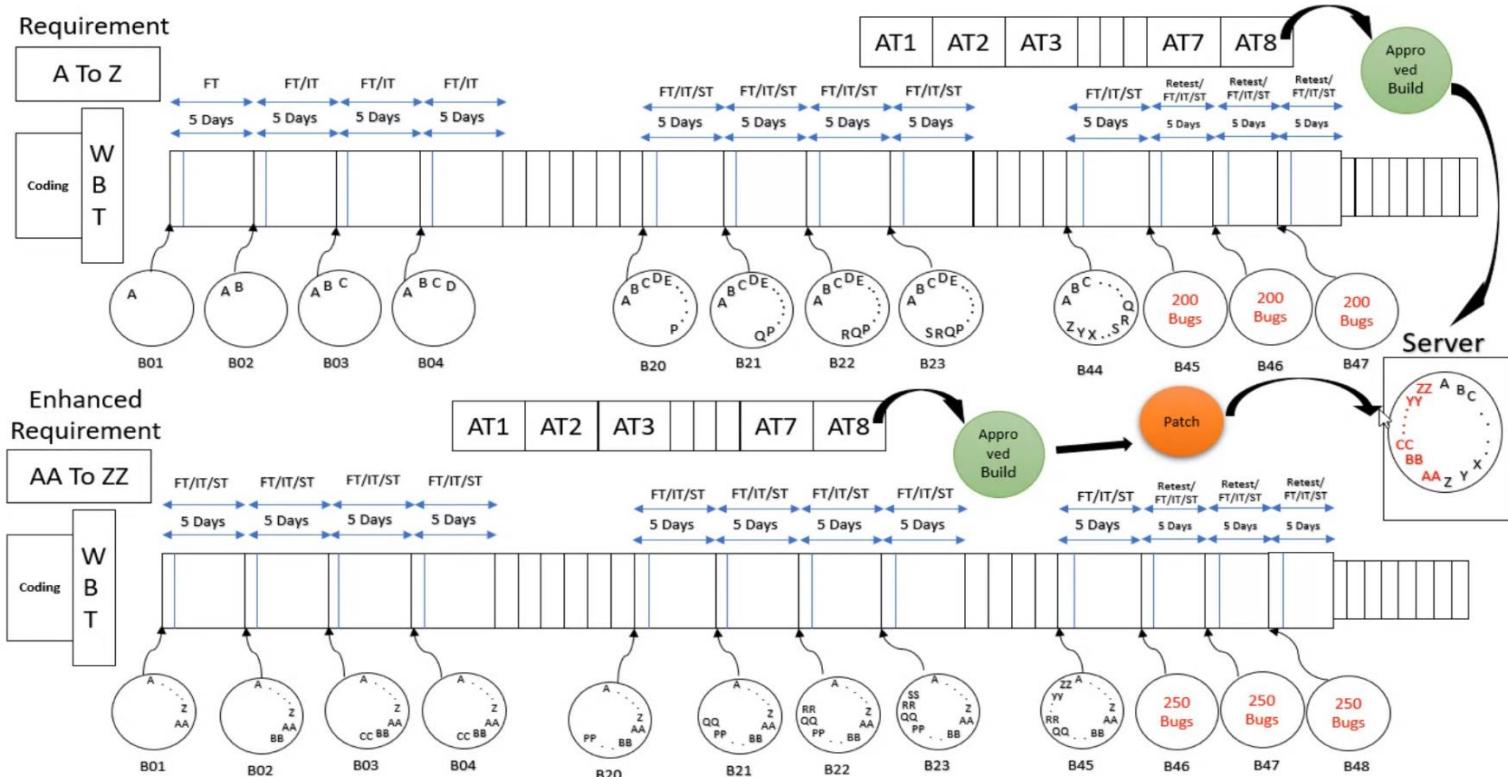
Here entire testing team meets & discuss about the list of achievements & mistakes (good process & wrong process followed). We document all the mistakes & achievements & this document is called as **Retrospect Document**.

In next release or sprint or project at planning stage we will open the old retrospect document & plan it in such a way that old mistakes are not repeated & good activities are once again adopted.

When we do retrospective meeting at the end of the meeting we will realise that the number of mistakes that we have done is very less.

This is where we continuously improve the test life cycle.

Traditional Model

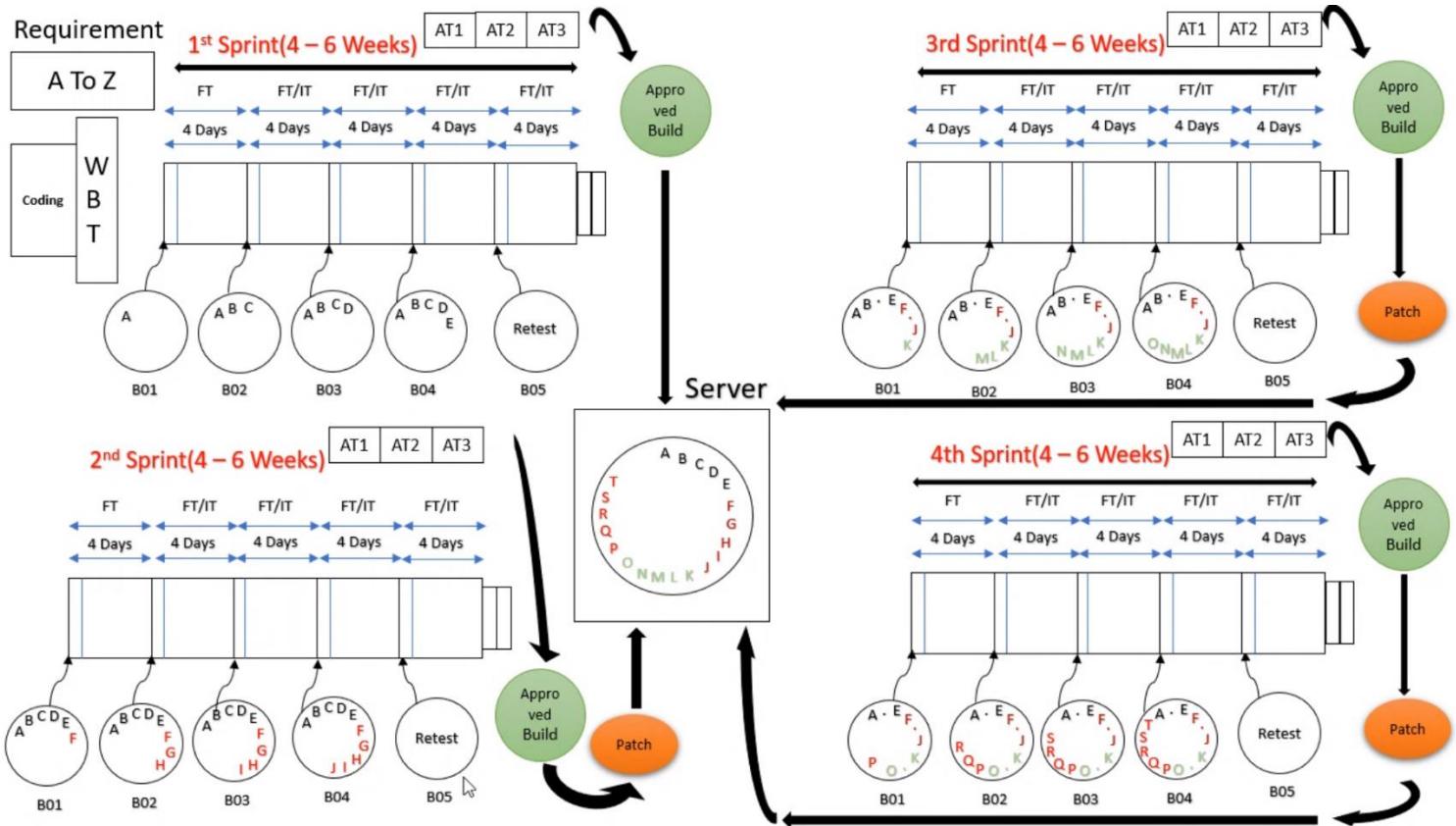


Drawback of traditional way of Software Development

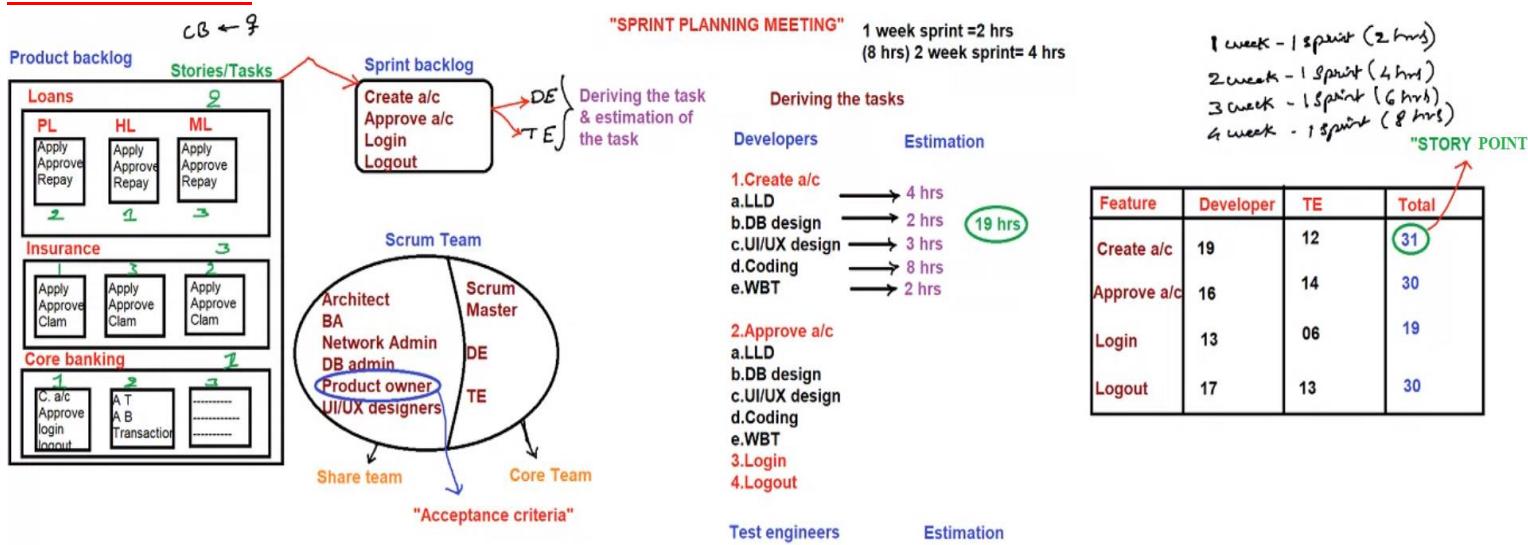
Drawbacks :

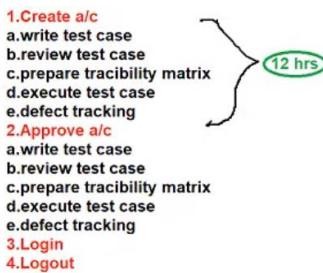
- Customer gave requirement on day one, will not be in touch till the product is delivered.
- One release might take anywhere between 10 to 16 months, in this time period customers business might change.
- Wont make people responsible.
- Customer invest money on day one should wait for return for very long time.

Introduction to Sprint



AGILE MODEL





1. What is agile?

Agile is a model where we develop the software in an incremental and iterative process.

*They come up with model in order to overcome the drawback which was there in traditional model.

*Here they build large product in short term cycle called sprint.

Note: Types of agile (don't mention in flow)

- 1.Scrum model / process
- 2.XP (extream programming)
- 3.Crystal clear
- 4.Lean & Kanban
- 5.FDD(Feature Driven Development)/TDD(Test Driven Development)
- 6.ASDM(Adoptive Software Development Method)
- 7.DSDM(Dynamic Software Development Method)

2.Scrum: It is a process used in agile to develop the software or application.

Note: 3.Scrum team:

- a)It is the group of engineers working towards completing committed features or stories.
- b)Generally scrum team will have 5-12 members.
- c)It includes share team & core team members.
- d)Core team includes scrum master,developer & TE.
- e)Share team includes architect,product owner,UI/UX designers , network admin ,DB designers.
- f)Scrum master leads the entire scrum team & he facilitates everyone to complete the task.

4.Product Backlog:

- a)It is the prioritized list of stories(requirement)that must be develop in complete project.
- b)Generally customer , product owner, Architect , scrum master will be involved while building it.
- c)Generally stories in product backlog need not be indetail.

5.Sprint backlog:It is the list of stories or task commited by scrum team that must be dilver within 1 sprint.

6.Sprint planning meeting:

- a)Here entier scrum master sit together & pull the stories from product backlog.
- b)Scrum master will assign each story to developers & TE's.
- c)Now each engineer will derive the task to be completed to build each story.
- d)Each engineer estimates time taken to complete their each task i.e, they derive "story point"
- e)Generally sprint palnning meeting should be completed within 8 hours (2 hours per week sprint)

[Following are the roles played different people in sprint palnning meeting]

1)Scrum master:This complete meeting is driven by scrum master.

His prime role is to facilitates the complete meeting & co-ordinates between all stakeholders(relevant people)
Example:Getting architects support to developers

getting product owners support to TE's to understand the requirement .

2)Product owner:He clarifies if any questions are there relateed to stories.

He will set acceptance criteria for each stories.

3)Development engineer:

- >He will derive the task of building each story .
- >He prioritises which story to be build first & later in the sprint.
- >He derives the story points (estimation for each task)

4) Test engineer:

> He will derive the task to be completed to test each feature.

ex: create account

 write test case

 review test case

 prepare traceability matrix

 execute test case

 defect tracking.

7. Acceptance criteria:

a) It's criteria which has to be met in order to move the product for production or customer.

b) It is generally set by product owner or customer

c) Generally they set this in the beginning of the sprint i.e., in sprint planning meeting.

 ex: create account : it should go to pending approval page

 Approve account : user should be listed in incometax list.

 user should be there in RBI list.

8. Story point:

It is the estimation of how long task would take to complete.

or

How long story is going to take for developing & testing.

9. Stand-up meeting:

a) Here entire scrum team means.

b) This meeting is completely done in front of scrum master.

c) here every engineer should explain

 1) what they have done yesterday

 2) what are the impediments or hurdles faced yesterday.

 3) what are the activities that he is expecting in order to complete today's job.

 4) what are impediments done he is expecting in order to complete today's job.

d) Scrum master tries to solve certain impediments right there in the meeting. If it takes too much of time, then he notes it down in impediment backlog & solves it later.

e) Generally this meeting should be conducted in the beginning of the day.

f) Here everybody should standup in the meeting so that they talk to the point.

10. Retrospective meeting:

a) Here entire scrum team means sometimes customer also joins & discusses about all achievements (good process followed) & mistakes (wrong activities performed) & document it & that document is called as retrospective document.

b) When the next release or next sprints starts while doing sprint planning meeting we refer this document & plan it such a way that good activities are once again adopted & old mistakes are not required.

11. Sprint review meeting:

a) Here entire scrum team meets at the end of the sprint.

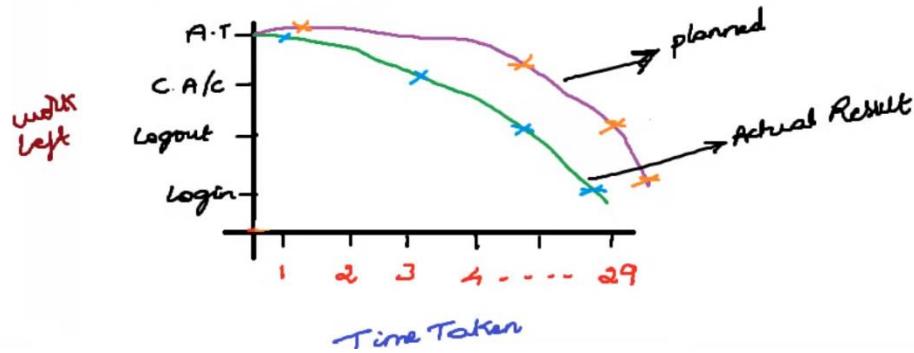
b) The engineers give a demo of whatever they have built to product owner.

c) Product owner will tell what is done & what is not done.

d) Also they discuss about how to plan for next sprint.

12. Burn down chart:

It is graphical representation of work left vs time taken.



13.Story board / White board:It is the list of stories or task contains the work which is in progress,pending & completed.

14.Spill over:There are certain features or stories which you are not able to build in this sprint & planning to postpone it for next sprint, that is called as spill over.

15.Chiken:Chiken is the one who observes & try to understand how the sprint is going on & he will not be doing any task.
ex:Customer,shareholders,director,VP,CEO etc.

16.There are 4 scrum ceromonies in agile they are:

- a)Sprint planning meeting
- b)Stand-up meeeting
- c)Sprint review meeting
- d)Sprint retrosepctive meeting

17.Here in agile literally use a word called,

Done,Done,Done(3dones) i.e,

- 1)WBT will be done by developers
- 2)BBT will be done by TE
- 3)AT will be done by customer.