→ S/W testing is platform independent.

→ S/W is the business need of the customer.

→ Ex: ICICI bank need a S/W. So ICICI bank is the customer (or) client for a S/W company. It goes to a S/W company for its business need. Then the S/W company will ask for its needs. Needs are to handle more no. of users.

① Login —
- User name – 5 alphabets
- PSW – 5 alphabets
- OK
- Cancel } → For enable.

② A/C Balance.

③ Amount Transferred

④ Loan

⋮

Logout.

Then the S/W company will develop the S/W according to the customer's requirement with using of any kind of languages.

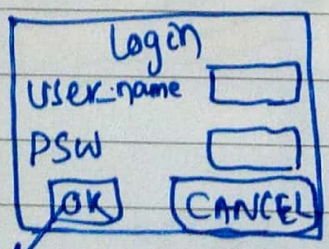
Developer

Programming languages like
- C
- C++
- C#
- JAVA
- .Net
- = } codes

Testing [Application by Test Engg.]

1. www.icici.com

2



Homepage

Scanned with CamScanner

If any kind of errors (or) bugs found out the T.E immediately report to the developer. Developer fix the bug & send it to the T.E for retest. This process continuous till the application is stable (or) bugfree (or) working according to the requirement.

→ Manual testing means process of testing the application as per the client's requirement.

→ It is called so, because the above process of testing by the test engineer manually.

→ Manual testing is done at 1st of all testing application.

→ To do a proper testing, the requirements should be clearly understand.

→ Testing depends on person to person's understanding of the requirements.

→ Automation Testing is done only with the help of some tools (or) individual s/w's.

* It reduces monotonicity i.e repeatition of a work
* Better accuracy
* Time saving.

→ Different types of tools are.

① QTP : Quick Test proffesional needs VB Script Here Support (or) Service is available Designed by HP.

Scanned with CamScanner

② Selenium : It needs core Java
It is open source i.e free
Support is not available.

③ Load Runner
④ SQL.

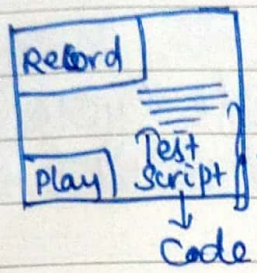
Programming Languages	Scripting Languages
C	VB
C++	JAVA
C#	Phython
JAVA	Shell } Important
.NET	Perl } Important
HARD	Easy
Complex	Simple
Runs in Server	Runs in Browser
Used for developing the application (or) S/w	Validate/Testing the S/w.

→ Ex: One Raja Proposed a Rani. Rani agreed in the proposal. Raja used to call Rani at 2 AM & used to say a single song to sing as Rani is a good singer. After few days rani get irritated & there's a break up in her relation.

So to maintain the relation the solution is raja should record the song so that he can listen as per his desirement. The the mobile through which the song can be recorded is a tool which is treated as a virtual rani.

Scanned with CamScanner

Recording



→ Record various things !

① Mobile (QTP)
② Singer (TE)

→ Advantages of Automation Testing is
① Time Saving
② Greater Accuracy
③ Reuse of the Script.

→ Automation testing is used for repeated testing.

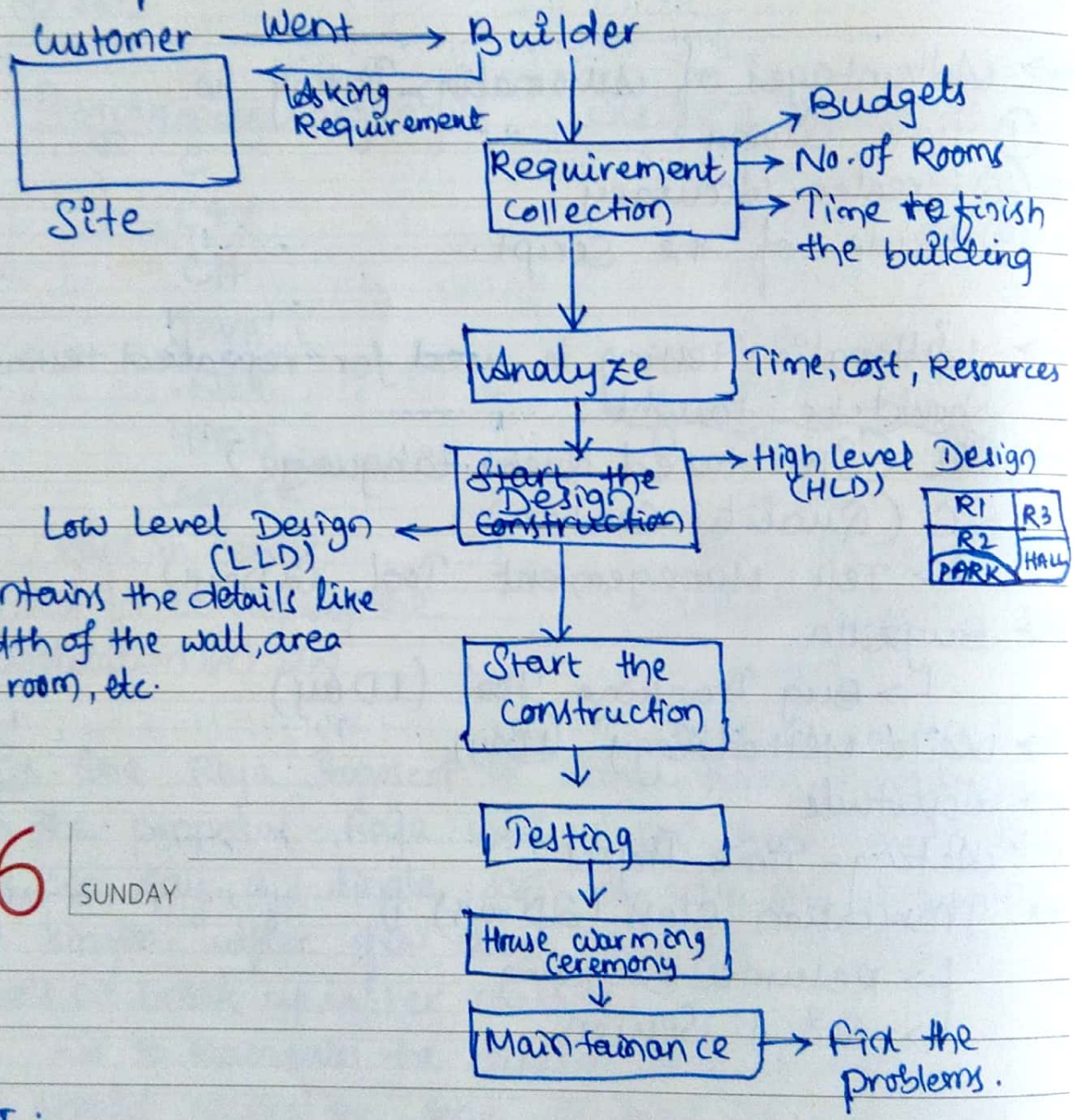
Should be taught:
→ SQL (Structured Query Language)
→ QC (Quality Center)
↳ Test Management Tool (2 Days)
→ Bugzilla
↳ Bug Tracking Tool (1 Day)
→ Agile Methodology (1 Day)
→ Aptitude
→ Acting - Time Project
→ Translation class (2 Days)
↳ Resume Building
↳ Q & A Session.

Scanned with CamScanner

Q. What is SDLC ?

SDLC means S/w Developement Life Cycle.

→ Ex: Suppose a person needs to build a house for him in a site.



06 SUNDAY

NOTE:

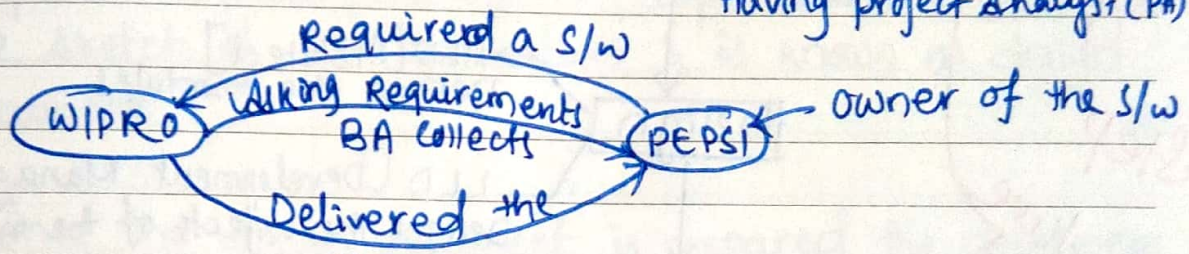
In HLD the modules are designed such as for an Gmail page, Login, Compose, Inbox, Sent items, Outbox, Logout where as in LLD the objects of modules are designed such as for login page; username, psw, ok, cancel, etc.----

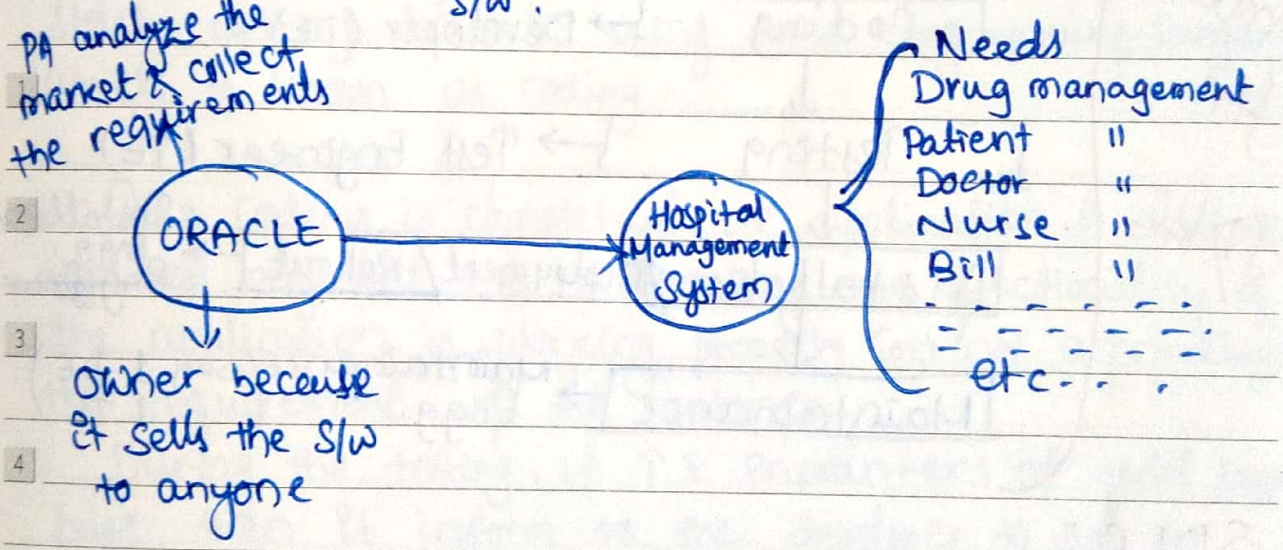
Scanned with CamScanner

→ Types of S/w company.

① Service Based (WIPRO, Infosys, CTS) → Having Business Analyst (BA)
② Product Based (Oracle, Microsoft) → Having Project Analyst (PA)

] HP, IBM, etc.

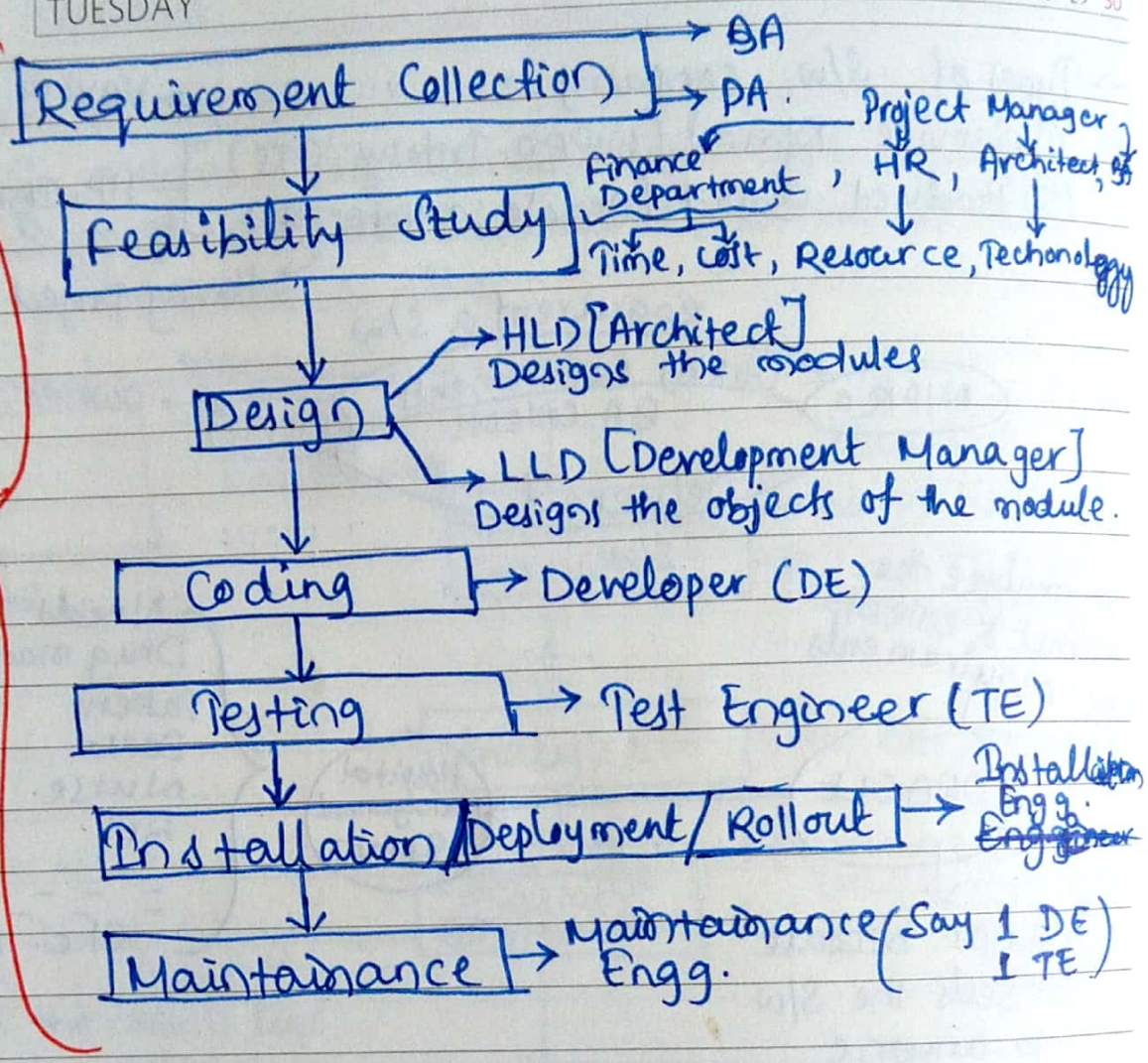




Ex: A person want to open a shop in Uganda & he decides to create a S/w for his shop for different business needs. He went to TCS which is in India. So TCS will follow the following ~~mo~~ flow chart.

Scanned with CamScanner

Step By Step Procedure for SDLC



SDLC:

→ It is the step by step procedure to develop a S/W

→ It consists of various stages like, Requirement collection, Feasibility study, Design, Coding, Testing, Installation & Maintainance.

→ In requirement collection we collect the business needs of the client & make a proper document.

→ In Feasibility study phase, a set of team sit & analyze the project is durable (or) not. It happens

once the requirements are collected properly.

→ Once the project is decided that it is duable, we prepare a blue print of the application which is the sketch of the application & is known as design document.

→ Once the design document is prepared the developers start writing the code using some programming languages, which is known as coding.

→ Once coding is completed i.e application is developed, the test engineer starts checking the functionality of the application is working properly (or) not according to the requirement of the customer.

During the testing, if T.E encounters of find some bugs then it inform to the developer to fix the problem.

The above is continuing till the application is stable (or) bug free & works properly according to the customer's requirement.

→ Once the S/w is developed, tested & works according to the client's requirement, it is handed over to the customer in the form of installation.

→ After installation, customer starts using the S/w & he may find some issues which can be fixed ~~by~~, retested

Scanned with CamScanner

& handed over to the customer as soon as possible which is done under maintainance stage.

This is the end of the SDLC.

NOTE:

Once the S/w is handed over to the customer, they may get some bugs for the following reasons.

1. Installation problem
2. The customer may not know how to use the S/w & he may use in the wrong way.

→ Ex: Suppose we need to go from Banglore to Goa. The media can be,

Bus, Train, Flight, car, Bike, cycle, walk

But we prefer the journey according to the Time, cost, comfort.

Like this when we develop a S/w we use some models.

Models of SDLC (For Quality S/w)

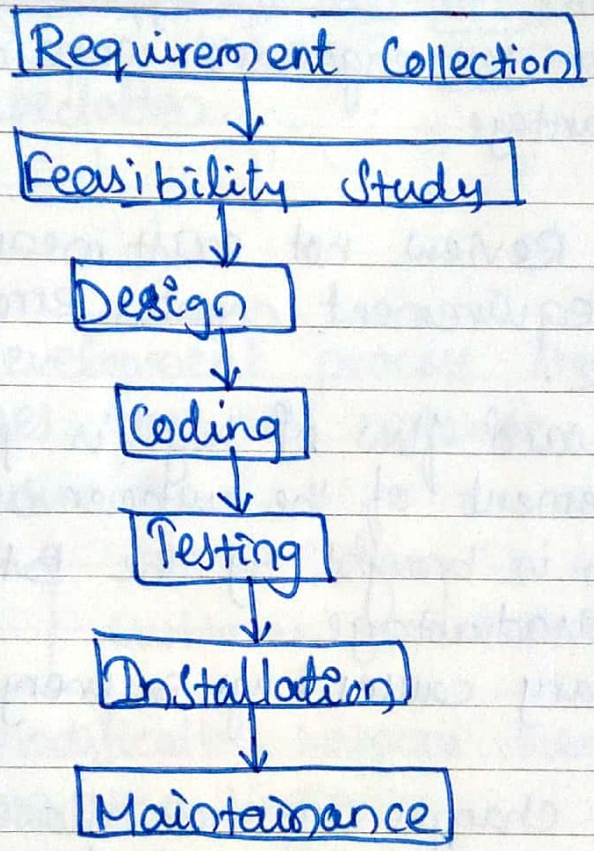
1. Water-fall Model
2. SPIRAL "
3. Prototype Model
4. Verification & validation (V & V) Model
5. Hybrid Model
6. Agile Methodology.

Scanned with CamScanner

16.4.14

Models of SDLC:

① Water-Fall Model:



→ It is the basic model of SDLC because all other models are derived from this model.

→ Execution happens sequentially because o/p of one phase is the i/p of other phase.

→ No parallel deliverables means no two works or two phases ~~executated~~ executed simultaneously.

Disadvantage
→ It is very time consuming because after execution of one phase the next phase is being executed. It is

a disadvantage.

Disadvantage

→ Developers used to test the application because ~~earlier~~ earlier test engineers were not there. It is a disadvantage.

Important

→ Requirement Review not exist means wrong collection of requirement creates errors (or) bugs in every stage.

1. Here down ward flow of bugs is possible. Here requirement of the customer is not cross checked which is brought by the BA. It is a disadvantage.
2. Wrong documentary causes bugs in every stage.

→ Requirement changes are not allowed. It purely depends on the application.

→ No back tracking because we can't move back.

Since back tracking is not allowed i.e once the requirements are implemented we don't allow any changes. It is called as waterfall model.

Suppose a customer want to change its business needs because of the following reasons.

Scanned with CamScanner

① Business : If there is a rapid growth in his business.

Ex: FB ≡ [ ] machine

② Technology : let say Logbook ≡ [ ] machine, [ ] mobile

migrated beoz of the technology as well as user expectation.

③ User Expectation.

NOTE:

Once development process starts, we never allow any changes from the customer.

Ex: Healthcare S/w

Code [ ] 70% Testing is over customer needs change (or) modification

If the modification happens then the TE should again from the begenning.

NOTE:

Q. Which type of application used in waterfall model?

(or)

Which type of application, is suited for waterfall model?

Ans: Whenever it is lifer critical (all healthcare application) (or) machine critical application we go for waterfall model because requirement changes should not be allowed for such kind of application in between the project.

Scanned with CamScanner

Eg: 4 Friends are there. One felt ill & gone to doctor. Doctor checked him by machine & told that the guy is suffered in cancer. Similarly other 3 will go after that & the machine checked & reported that they are also suffered in cancer & all committed suicide. The doctor also checked himself by the machine & report was cancer. Then he tested himself manually & got the report no cancer. This is life critical.

Lunching of missile: If the missile targeted to its origin place instead of targeting to the original target then it is machinary critical.

NOTE:

① Apart from waterfall model, requirement changes are allowed in all other models.

② Apart from waterfall model, TEs test the application in all other models.

Scanned with CamScanner

18.4.14

Drawbacks of water-fall Model:

Needs Excell Requirements

Customer → Application

Total 2 Years

Needs 6 months → ① Module-A → cells (Rows & Columns)

6 months → ② Module-B → Operations in Cells

6 months → ③ Module-C → Merge the cells

6 months → ④ Module-D → Macros (Calculations having Formula)

If we will use the waterfall model then the customer will get the required application after 2 years which is a very long time period & in between the customer will need to change the requirements according to the business needs, technology & user expectation.

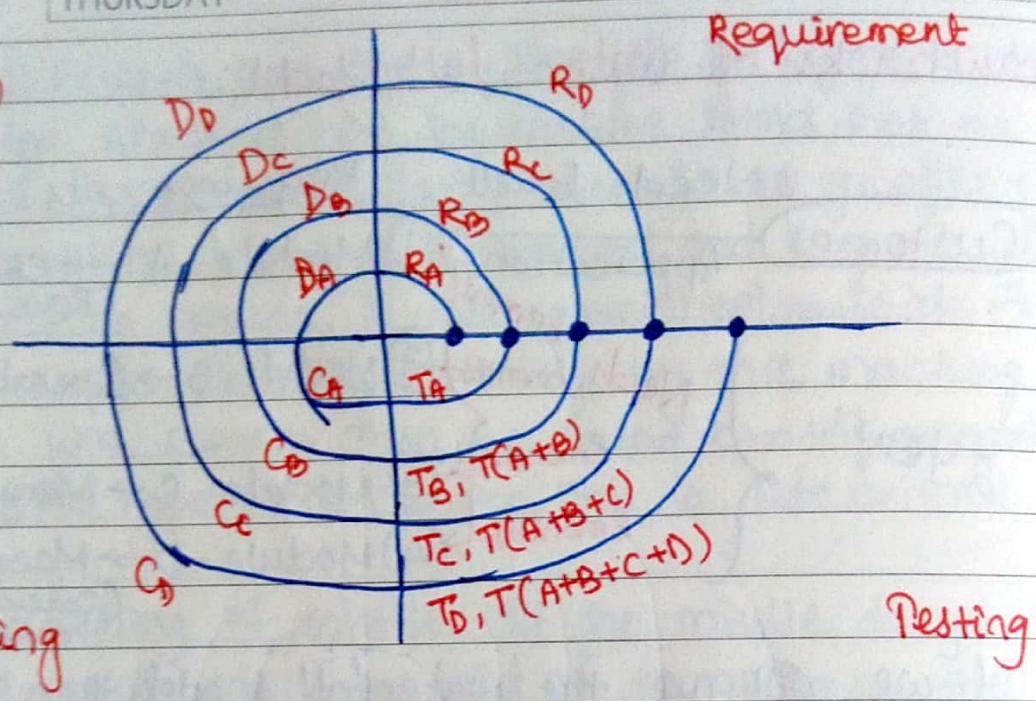
But it will be possible, that the client will use the basic module which is module-A. So at 1st module-A is developed, tested & installed. This can happen only iff the modules are dependent on each other.

② Spiral Model / Iterative Model / Cyclic Model:

→ Generally we go for this model whenever the modules are dependent on each other.

→ In this model we develop the application module by module & handed it over to the customer

Scanned with CamScanner



Advantages:

→ Customer can use the S/w at very early stage. Here the customer get the basic module for using the application

→ Flexible for changes i.e at any instant of time the coding can be upgraded (or) changed according to the customer's need.

Disadvantage:

→ No parallel Deliverable.

→ There are no requirement review.

During the development of the client need to change the requirements then the developer team will decide about that.

Scanned with CamScanner

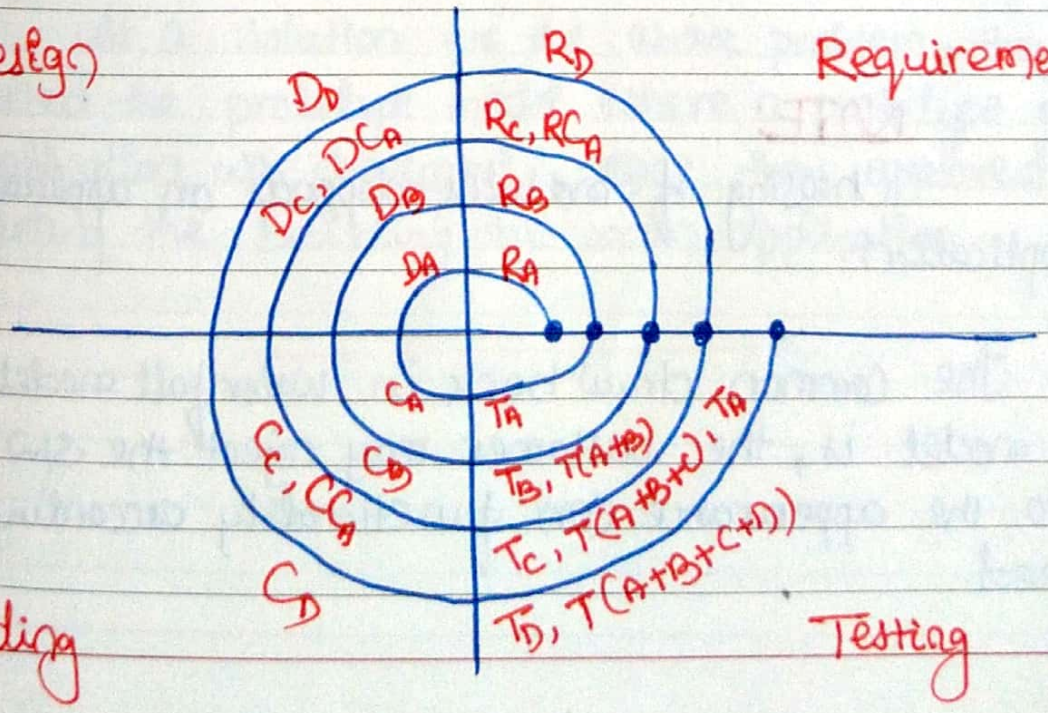
Ex: If the S/w company collected the requirement for Module-C. At that time if the customer told the BA for the modification in module-A. At 1st the team will watch either the change is minor change (in this change the change in module A won't hamper (or) effect the other modules & the current module will be developed) (or) major Change (If the change in one module effect the next module).

If the change is the Minor change:

→ In this case, we accomodate the changes with the next module of the same cycle i.e we don't implement extra cycle for minor changes.

→ Changes are entertained along with the new model.

→ Minor Change won't effect the further modules.

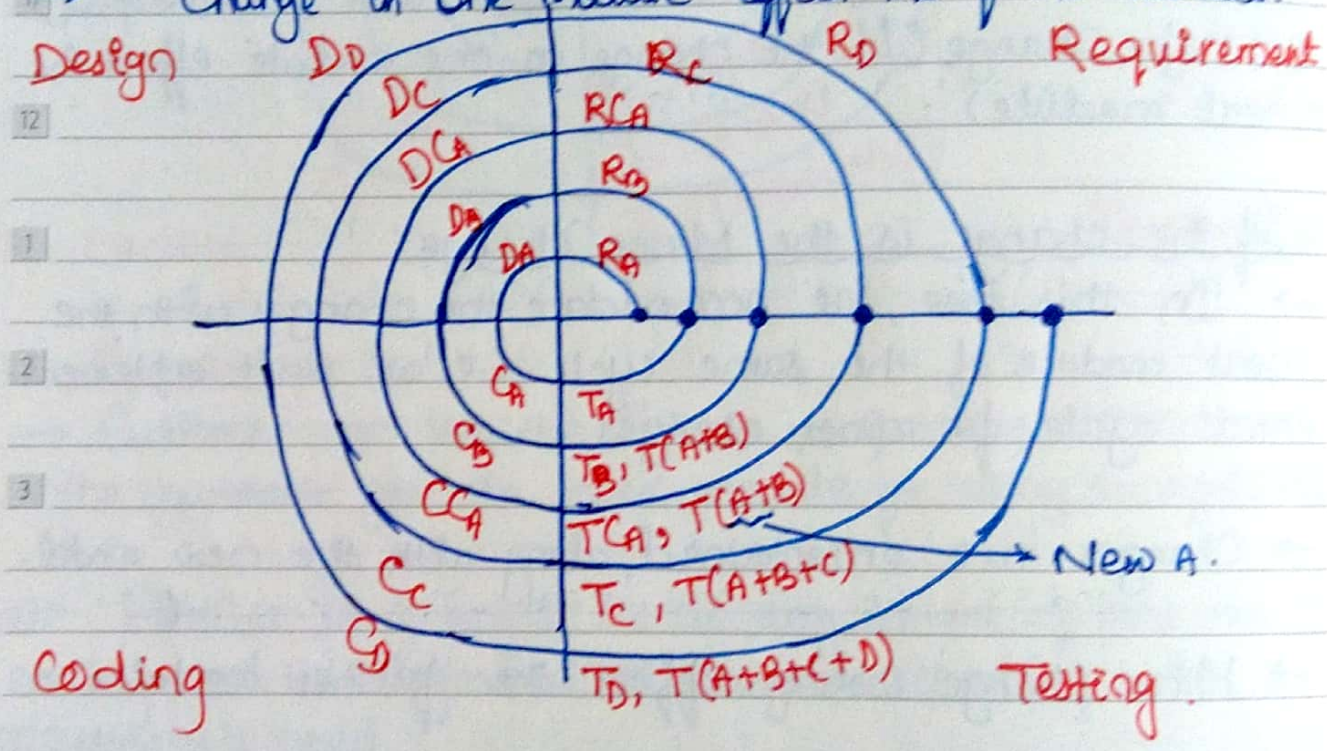


Scanned with CamScanner

<u>If the change is the major change:</u>

→ In this case we go for one more extra cycle since these are the major changes & it may effect the existing functionalities.

→ Change in one module effect the further modules.



20 SUNDAY

<u>NOTE:</u>

Choosing of modules depends on application to application.

The common draw back in waterfall model & spiral model is, the customer may reject the s/w due to the appearance (or) functionality according to his need.

Scanned with CamScanner

Ex: Suppose a girl working in abroad & her familey member asked her the requirement in a boy to whom she will get marry. Suppose the girl won't see the boy before & while she is in mandap she saw him & rejected because the boy is fat & looking ochward. So here the solution is the girl should see the photo of the boy which is almost all 90% same as real. Here the photo is the sample (or) the prototype.

So to avoid the customer rejection the S/w company should design a sample (or) prototype according to the requirement which will be developed.

NOTE:

1. The main draw back of waterfall & spiral model are customer rejection was more since they never had an idea how the S/w looks like after it is developed.

2. As a solution for the above problem, they started the prototype model where a prototype of the application was developed & once they approved we starting the building the main application.

Scanned with CamScanner

21.4.14

Very Imp

⑥ Agile Methodology:

→ This model is used 95% in the market for SDLC.

→ Requirements are of two types:

ⓐ CRS : Customer Requirement Specifications

* It is the 1st Stage
* It is given by the customer.
* Here only the business needs are documented.
* It is the high level requirement document.

Customer
Username
PSW
OK
- - - - - -
- - - - - -
- - - - - -
- - - - - -

Here basically modules are mentioned only.

ⓑ SRS : System Requirement Specifications

* After CRS document prepared, BA Converts the CRS document to SRS document.
* It is the low level requirement document.
* Here the step by step business needs are mentioned in a document.

① Username:
ⓐ Contain 5 char
ⓑ 4 alphanumeric
② PSW:
ⓐ - - - - -
- - - - -
- - - - -

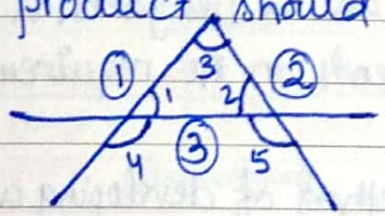
Details of each module is prepared in a document.

Scanned with CamScanner

Ex: Suppose a customer mentioned his requirements:

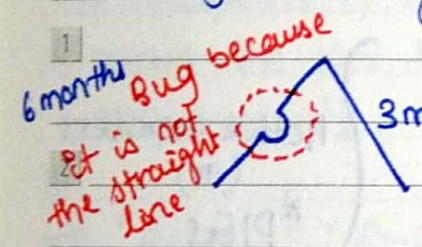
① My product should contain 3 straight lines.
② One line should bisect two lines.
③ Product should contain 5 inner angles.

The product should look like the below.



①, ②, ③ → Straight lines
1, 2, 3, 4, 5 → angles.

This has given to the BA.

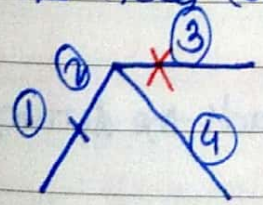


3 months + time taken for bugfree

As a result, the final design is

The above project is rejected because it is not according to the customer's need. It happened due to non-involvement of the customer.

When customer involved in the project frequently, he came to know whether the project is going according to his need (or) not.



The 3rd phase is rejected due to customer involvement because the design was not according to the customer's need.

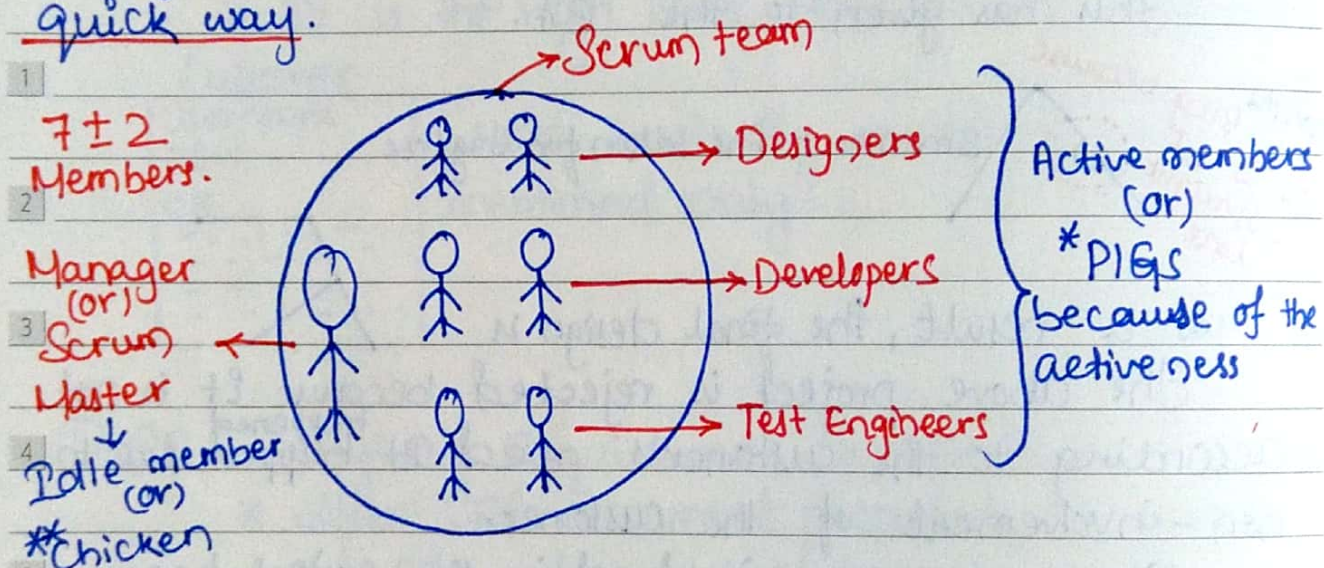
→ Here there is a frequent involvement of the customer so that the product is a robust product.

Scanned with CamScanner

→ Here we need expert designer, expert developer, expert Test engineers, expert manager & highly qualified peoples.

→ It normally contains 7±2 members in a team which is called as "Scrum Team" means 2 Designers, 2 developers, 2 TEs & 1 manager (It depends on the requirement).

→ Agile means it is a method of developing a S/w in quick way.



→ "Scrum" is a process to develop a S/w in iterative & increamental way.

→ "Scrum Master" is the person who leads the Scrum team.

NOTE:

Requirements are also called as "User Story".

2013

* PIG: If we leave a PIG then anywhere then it'll go near to the drain.
** Chicken: If we leave a chicken anywhere then it'll behave as idle.

Scanned with CamScanner

50 requirements
↓ Pick by Scrum master.
10 requirements
↓ Explain to developer.

Customer	Customer	Customer	Customer	Customer
1st 10 requirements 15 days 1st Sprint.	10 req. / 15 Days 2nd Sprint	10 req / 15 days 3rd Sprint	10 req / 15 days 4th Sprint	10 req / 15 days 5th Sprint.
Involvement	Involvement	Involvement	Involvement	Involvement

→ "Sprint", it is otherwise known as "time slots". It means the time duration required to develop certain set of requirements.

→ "Release": Once after product is stable we handover the product to the customer. This process is called as release.

Disadvantage:

If any member leaves the project, then the project gets failure.

Types of Meetings:

① **Project Kick-off Meeting:** It is a type of meeting in agile which is done before the starting of the project. It is done by BA, Scrum master with all high designated peoples.

② **Sprint Planning Meeting:** This is a meeting with the team itself along with the scrum master before developing the application.

Scanned with CamScanner

in each Sprint.

③ Daily Standoff Meetings: It is a kind of meeting with the scrum masters & the team members to discuss about
ⓐ How much work finished yesteday?
ⓑ what works will be done today?

④ Reterospect / Post mortem Meeting: It is the type of meeting done once ~~of~~ after the product is released. Here the team discuss:
ⓐ What went wrong in the project?
ⓑ What to improve in the next project?

Product Owner:

It is the person (or) organization who is handling the requirements.

Sprint Banned:

Whenever the ~~th~~ requirements are dependent on each other the project work continues w:r:t the sprint. But when the requirements are brought from different modules ~~are~~ or the requirements are not dependent on each other at that moment we ban the sprint & the process is called as sprint banned.

Sprint Backlog:

The requirements which are going to be developed in a sprint is called as sprint backlog.

Scanned with CamScanner

Carry forward:

The requirements moved from one sprint to other sprint is called as carry forward.

Completed 9 req./15 days	1 req. left	
1st Sprint		2nd Sprint.

Product Backlog:

The requirements given by the customer is called product backlog.

50 req. $\xrightarrow{\text{To be developed}}$ Product backlog.

Interview Questions:

① What is Sprint?

*② What is the team size of your Current project? (7 - 8 members)

*③ What is the project duration? (Say 1 year → 250 working days)

*④ What is your Sprint duration? (10 - 15 days)

*⑤ How many sprints you spent? ($\frac{250}{15}$ = 16, 16-18 sprints)

⑥ What is sprint backlog?

*⑦ What are the types of meetings you involved at?

⑧ What is agile methodology?

⑨ Explain agile process.

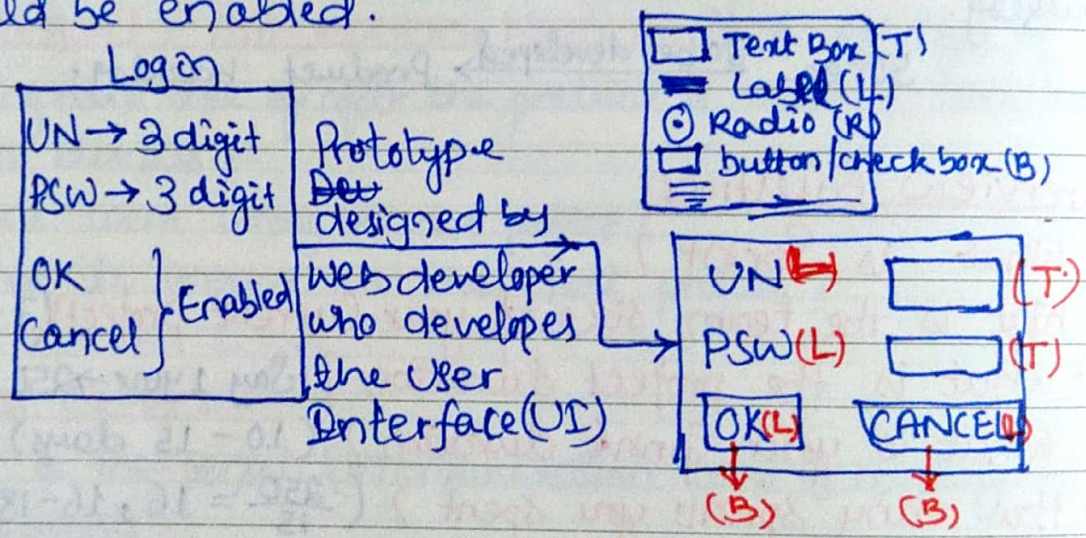
* → For Experianced people.

Scanned with CamScanner

③ Prototype Model :

→ Here one dummy/prototype/sample model is shown to the customer. If that prototype is as per the customer's requirement then that project is approved otherwise it is being rejected.

→ Ex: Suppose a customer needs a login page having User Name (UN), Password (PSW), OK & Cancel button which should be enabled.

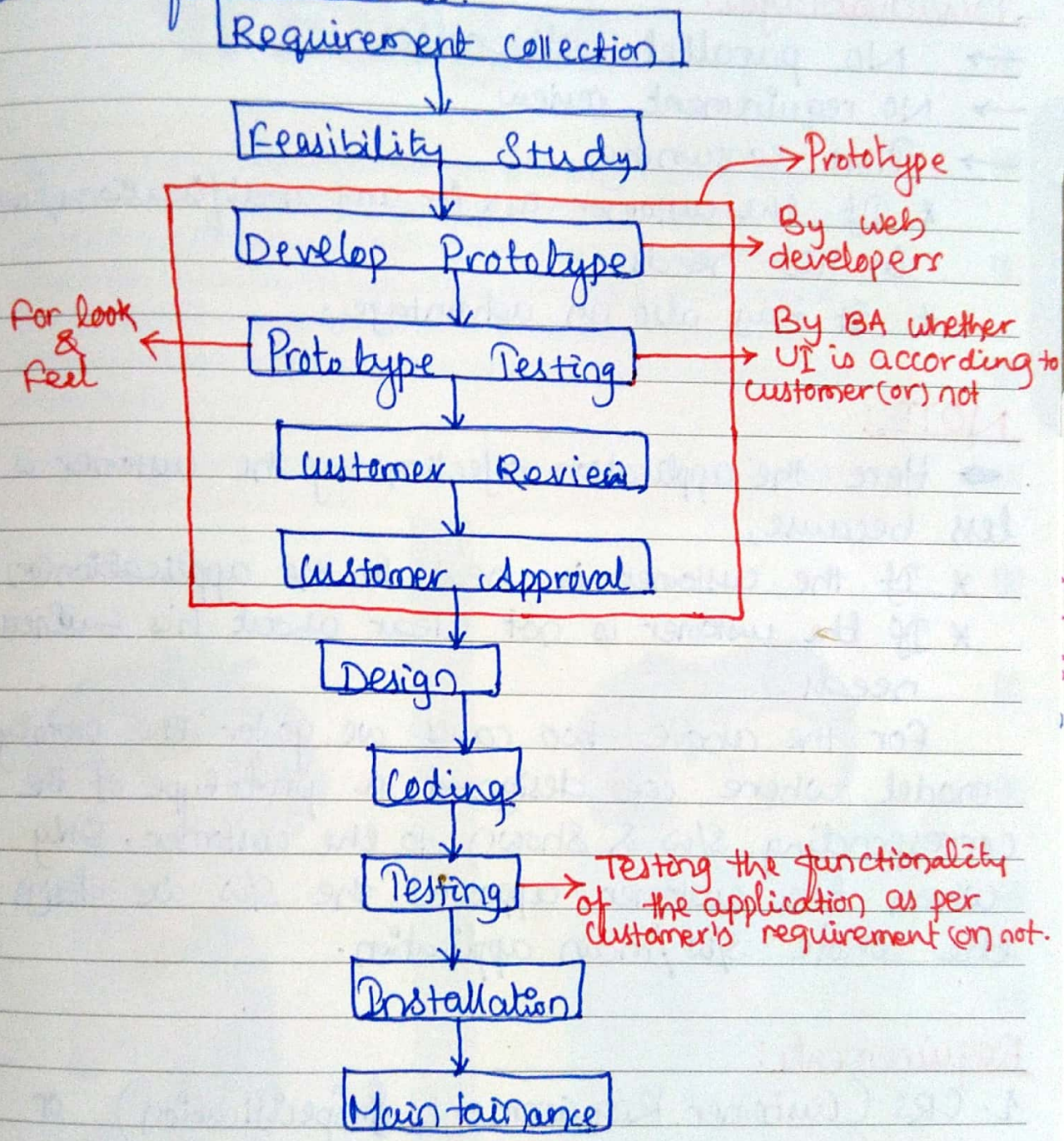


→ Here the customer rejection becomes less because the customer will decide that the s/w what is going to be developed that is in accordance to the customer need (or) not.

→ Still here the customer rejection possibility is there because the customer is unknown about the complete functionality of the s/w.

Scanned with CamScanner

→ Here the flow model is.



Advantages:

→ Customer Satisfaction is exist.
→ Customer rejection possibility is less
→ Customer can feel his application at very early Stage
→ Customer review exist.
→ Prototype can be reused in H.L.D & L.L.D.

Scanned with CamScanner

Disadvantages:

→ No parallel deliverables
→ No requirement review
→ Time consuming
* If the customer ask for any modifications/changes in his needs.
* It may also an advantage.

NOTE:

Here the application rejection by the customer is less because,
* If the customer is new to the application (or) S/w
* If the customer is not clear about his business needs.

For the above two cases we go for the prototype model where we designed a prototype of the corresponding S/w & shown to the customer. Only when the customer approves the S/w we design the whole S/w / main application.

Requirements:

1. CRS (Customer Requirement Specification) or BRS (Business Requirement Specification)

This is written by the BA in business language which is a high level document.

Ex: Req. for Gmail.

1. User able to register
2. User able to login
3. User write mails.
4. User view mails.
5. User save mails
6. User logout

Scanned with CamScanner

② SRS (System /S/W Requirement Specification) or
FS (Functional Specification)
It is documented by the BA itself which is in details. It is a low level document.

1. Login
   1.1 UN → Text Box
   1.1.1. UN → Accept 3 digit
   1.2. PSW → Text Box
   1.2.1. PSW → Accept 3 digits
   1.3. OK → Button → Enabled
   1.4 Cancel → Button → Enabled
2. Compose
   2.1. To
   2.2. CC
   2.3. BCC

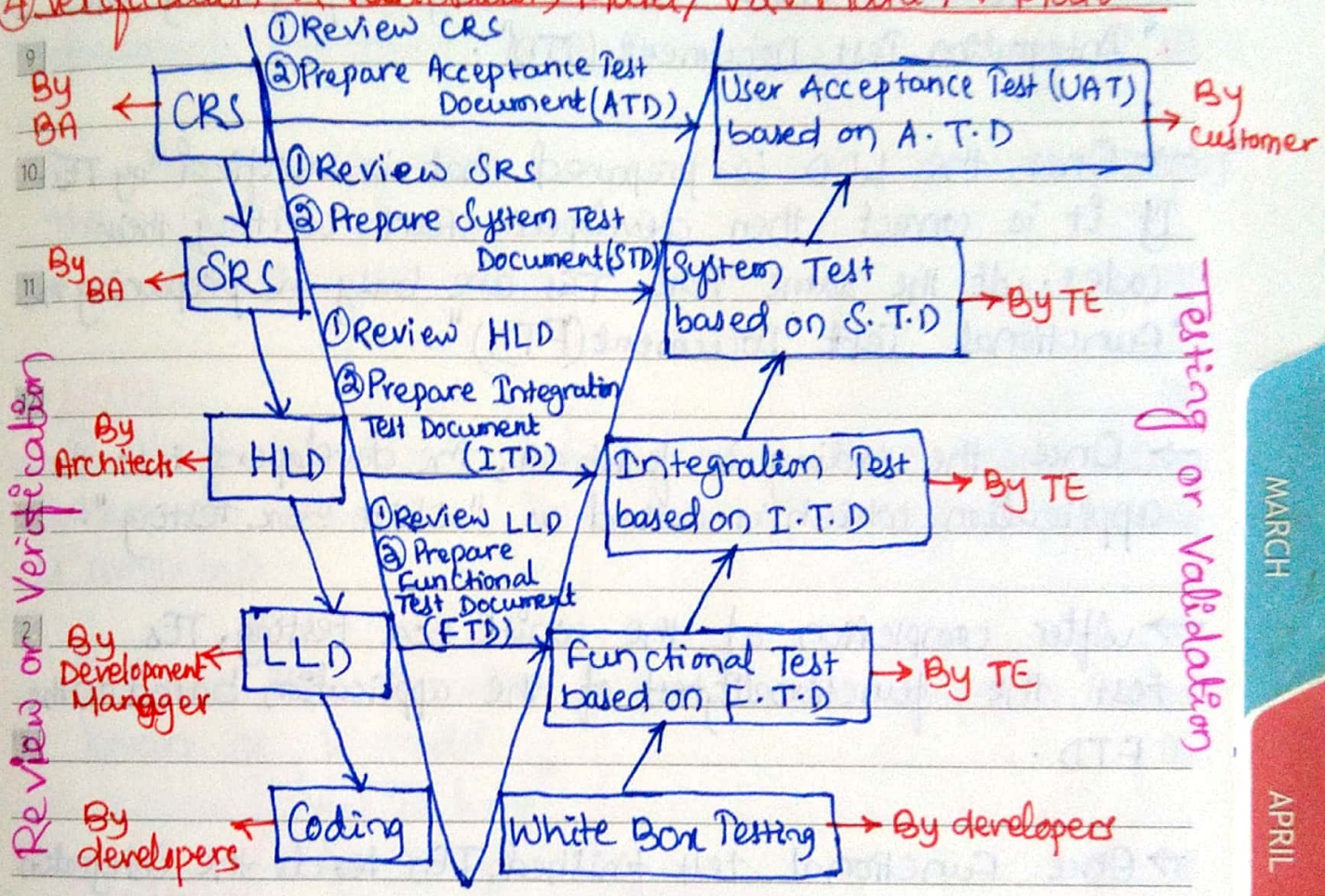
   ---
10. Logout

Characteristics of Good Requirements: (for SRS)

① In details
② Easy to understand because simple language is used for the documentation.
③ Proper flow means the modules as well as the objects of the modules should be in proper order / proper manner.
④ Measurable means the requirements are properly countable.

Scanned with CamScanner

④ Verification & Validation Model / V&V Model / V Model



→ Here once CRS brought by the BA the TEs verifies the CRS document. If it is correct/bugfree then BA will prepare SRS. At the same time TEs will prepare the "Acceptance Test document (ATD)".

→ Once SRC prepared by BA, TEs will review the SRS. If it is correct, then the architect prepare the HLD & at the same time TEs busy in preparing the "System Test Document (STD)".

→ After completion of HLD that is being verified by the TEs. If it is bugfree then LLD will be prepared

Scanned with CamScanner

& at the same time TEs are busy in preparing the "Integration Test Document (ITD)".

→ Once the LLD is prepared that is verified by TEs. If it is correct then developers start writing their codes. At the same time TEs are busy in preparing the "Functional Test Document (FTD)".

→ Once the coding is finished, the developers test the application which is called as "white box testing".

→ After completion of the white box testing, TEs test the functionality ~~test~~ of the application based on the FTD.

→ Once functional test finished, TEs ~~tes~~ do the integration test based on the ITD.

→ When integration test is over, the TEs will do the system testing based on the STD.

→ Once the system testing is over, customer will check or test his application as per the acceptance test document.

In this model, the test engineers are hired when a project came for develop.

Scanned with CamScanner

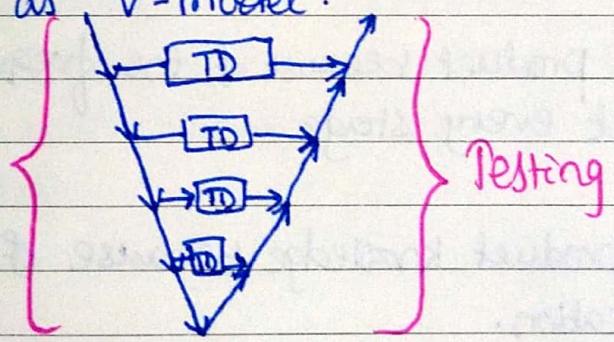
In this model the TEs review & prepare the test documents in each stage till coding.

→ The test document is nothing but the containing of the : ⓐ Test plans.
ⓑ Test cases.

NOTE:

① Verification / Review always takes place before coding where as Validation / Testing takes place after the coding is completed.

② Since the execution happens in V-shape it is also known as V-model.



③ This model works in 2 steps.
ⓐ All the verification activities takes place
ⓑ Once the S/W is developed we start the validation activity.

Hence it is called as Verification & Validation model.

④ V-model is best suited for large & Complex applications.

Scanned with CamScanner

Advantages:

→ Parallel Derivarables Exist because two phase are working simultaneously.

Ex: During preparation of SRS, TEs are busy in preparing the Test Document.

→ Requirement review is exist at every stage.

Ex: Review of CRS

① User login ② User create mail ③ User Read mail ④ User Delete mail ⑤ User view mail	ⓐ Here 3 & 5 are having same meaning: Conflict Requirement ⓑ Missing Req.: Before Login, user must register.

→ Having a robust product because of the frequent review of requirements at every stage

→ TEs have good product knowledge because of the frequent review of the application.

→ Testing process becomes easy.

Disadvantages:

→ Costlier: All the developer as well as TEs hired for the project in the early stage.

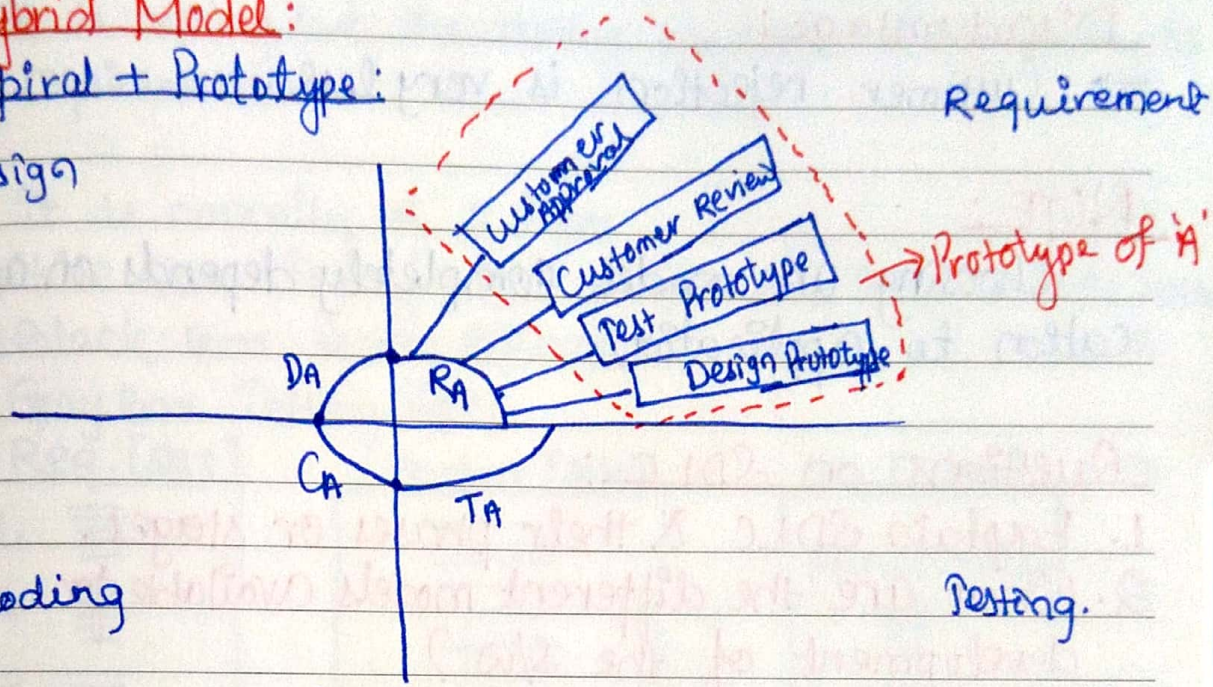
→ Lots of rework may come if the customer asks for changes.

Scanned with CamScanner

25.4.14

⑤ Hybrid Model:

@Spiral + Prototype:



→ Here at 1st requirements of module-A will be collected. Then the prototype of module-A will be developed. If the prototype is approved by the customer then the design, coding & testing process will continue. This process will continue till all the modules will be prepared.

Advantages:

→ Customer can use the S/w at very early stage.
→ Customer satisfaction is exist.
→ Flexible for changes
→ Customer rejection possibility is less
→ Customer can feel his application at very early stage
→ Customer review exist.
→ Prototype can be reused.

Scanned with CamScanner

Disadvantage:

→ Customer rejection is very less.

NOTE:

Choosing an model completely depends on application to application.

Questions on SDLC:

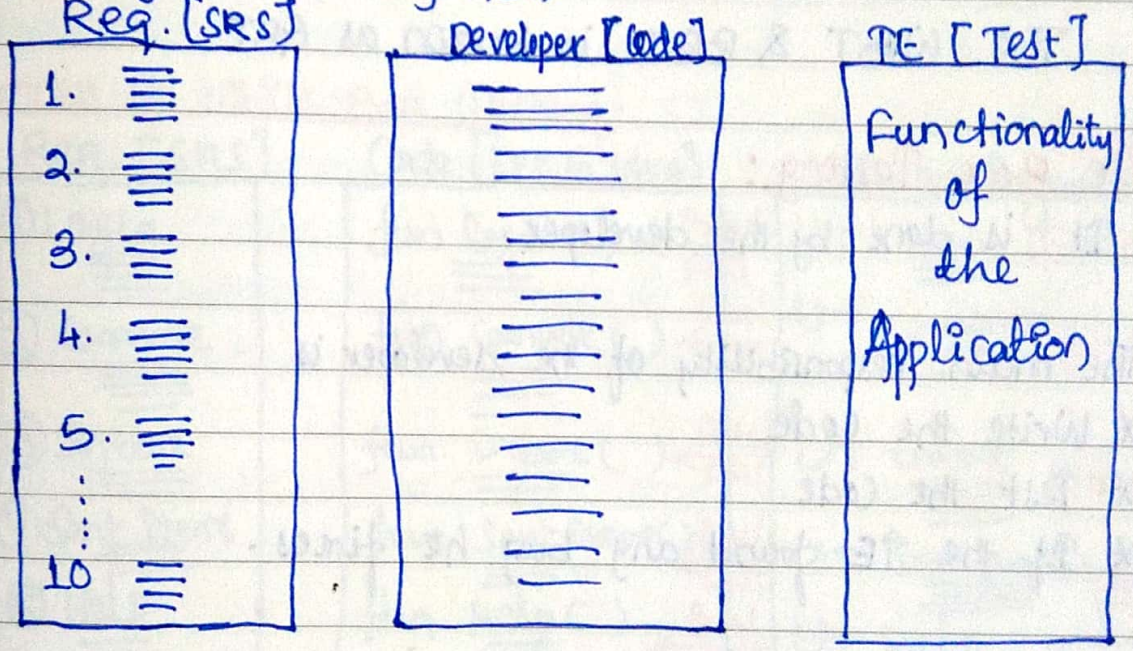
1. Explain SDLC & their phases or stages.
2. What are the different models available for development of the S/W?
3. Explain V-model.
4. Why it is called as V-model or V&V model?
5. Explain the advantages & disadvantages of V-model
6. What is a hybrid model?
7. Which all models can be combined to form hybrid model? (1. Spiral + Prototype
2. Prototype + V)

Scanned with CamScanner

TESTING

→ It is nothing but the testing the application according to the requirements.

→ It is normally of 3-types
① White Box Testing (WBT) or Glass Box Testing or Unit Testing.
② Black Box Testing (BBT)
③ Gray Box Testing (GBT)



→ White box testing

* It is a type of testing done by the developers where they check each & every line of the codes before handing over the application to the TEs.
* It is known as white box testing since the code is visible for the developers during the testing process.

Imp
→ Black Box Testing (100% interview questions)

* It is a type of testing done by the TEs where they check the functionality of an application works according to the customer requirements.

Scanned with CamScanner

* It is known as black box testing because the codes are invisible to the TEs during the test process.

→ Gray Box Testing

Generally it is done by the person who has the knowledge on both the coding as well as testing.

or

The WBT & BBT is known as GBT.

White Box Testing:

→ It is done by the developer.

→ The main responsibility of the developer is
* Write the code
* Test the code
* If the TE found any bug he fixes.

→ Code is nothing but the set of instructions or using of syntaxes or simply variables combined with functions

Code

int a char abc float abcd	Variables
main ( ) Fun add ( ) Fun sub ( ) Fun mul ( ) Fun ( )	Functions

Calculators

1	2	3	+
4	5	6	−
7	8	9	×
0	C	=	/

1+2 = −1 (wrong call of function)

Scanned with CamScanner

Drawbacks of manual testing of codes:

① It is time consuming because developer takes a lot of time to test line by line of the coding part. If there are a huge nos. of codes then it takes more time.

② No accuracy: It means developer may leave some codes during the testing.

③ Tedious Process

Process of white Box Testing:

Req. [SRS]	Code [Let in JAVA]	Unit Test Program (UTP) (Execute whole)
① Login	fun login( ) ← Pass/Fail	UT Login (Execute)
② Compose	fun compose( ) ← Pass/Fail	UT compose (Execute)
③ Inbox	fun inbox( ) ← Pass/Fail	UT inbox (Execute)
④ Sent Items	fun Sent Items( ) ← Pass/Fail	UT Sentitems (Execute)
⑤ Help	fun help( ) ← Pass/Fail	UT help (Execute)
⑥ Logout	fun logout( ) ← Pass/Fail	UT logout (Execute)

→ In UTP we can either execute function by function or we can execute the whole UTP. when all the pass result will come from all functions then the code is correct.

→ Normally we use unit testing tools for checking the code is correct or not. This is language specific means for different programming language we use

Scanned with CamScanner

different tools. Ex: for JAVA the tool is JUNIT, similarly for .NET the tool is NUNIT etc. By the help of this tool, execution takes very less time.

NOTE:

1. Whenever bugs found by TEs, then it is fixed in the main program & we should not change the UTP. But whenever the requirements changed at that time change in UTP is must & the UTP must be in accordance to the customer requirement.

2. Always the UTP is reviewed & approved by otherperson to become bug free in the UTP itself.

3. The developers pass the program to the TE whenever all the results by the UTP is pass.

4. We should not answer about the UTProcess untill & unless it is being asked.

Unit Test Process:

→ Once the coding is completed developers need to check whether the written code is executed properly or not which is done in white box testing.

→ During white box testing, if the process will be done in manual, then it is very time consuming, accuracy is not maintained as well as it is a tedious process.

Scanned with CamScanner

→ Hence the white box testing is done with the help of UTP.

→ The developers write a separate set of programs known as UTP which ~~is~~ helps to test the main program.

→ Generally UTP are written with the help of ~~as~~ Unit testing tools.

→ The main advantages of UTP are,
ⓐ Time Saving
ⓑ Accuracy is maintained
ⓒ This program can be reused.

Scanned with CamScanner

26.4.14

Types of White Box Testing:

① Path Testing
② Condition testing
③ Loop Testing.

NOTE:

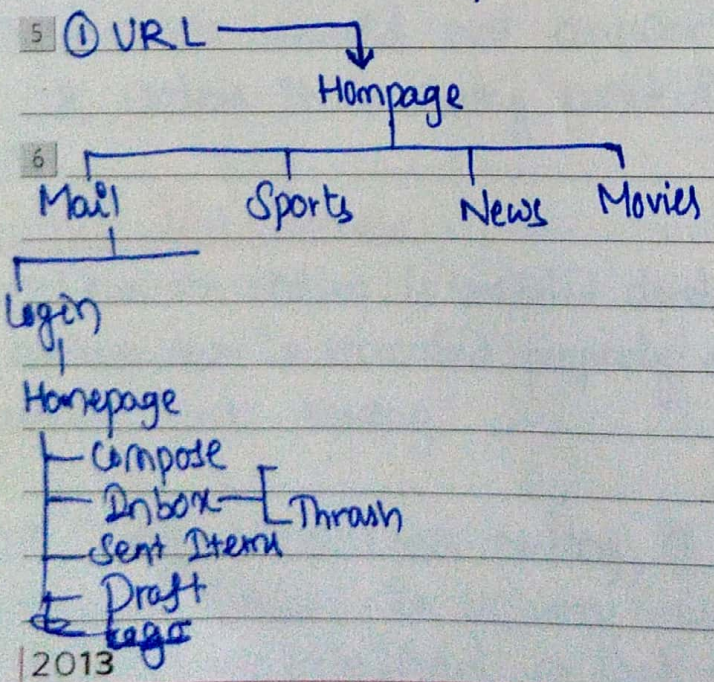
WBT is done w.r.t codes of the program.

1. Path Testing:

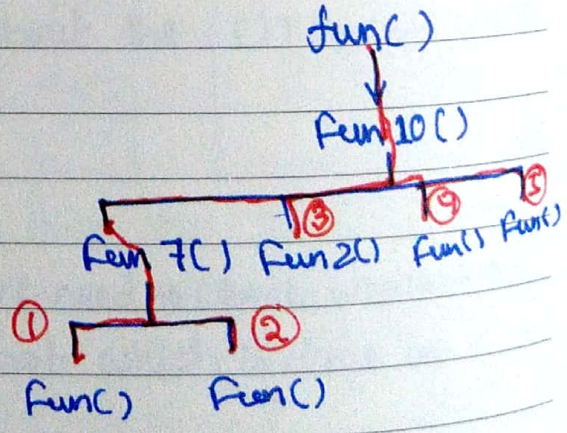
→ Here the developers check the flow of execution of the program i.e the main fun.? calls the proper/correct sub fun.? or not.

→ They also check the total no. of paths in the program.

Ex: Suppose the TE test the path.
Lets take an example of Yahoo



If the Developer test the path



Here there are 5 paths.

Scanned with CamScanner

→ Here normally the developers concentrating the checking the flow of execution of the function or the proper calling of the fun.?

NOTE:

The developers do the path testing w:r:t the code where as the TE do the path testing w:r:t the application.

② Condition Testing:

→ Here the developers check if the written code is working both for true as well as false condition.

Ex: For login Page

if UN/PSW is valid
display homepage } For True condition

else
if UN/PSW is invalid
display error message } For False Cond.?

③ Loop Testing:

→ The developers define many iterations or cycles or loops in a program.

→ The developers cross check if the written program is successfully executed for the total no. of loops specified in the program or not.

→ If the condition is satisfied then program

Scanned with CamScanner

should stop execution else continue till last loop then stop execution.

Ex: Case I : if cond.^n satisfied
break;
Case II : if cond.^n satisfied
break;
Case III : if cond.^n satisfied
break;

Here the cases will be checked one after another if one case becomes failed or else it will come outside & execute all the otherlines.

NOTE:

① In loop testing, developers checks if the written program is successfully executed for the total no. of loops mentioned in the program.

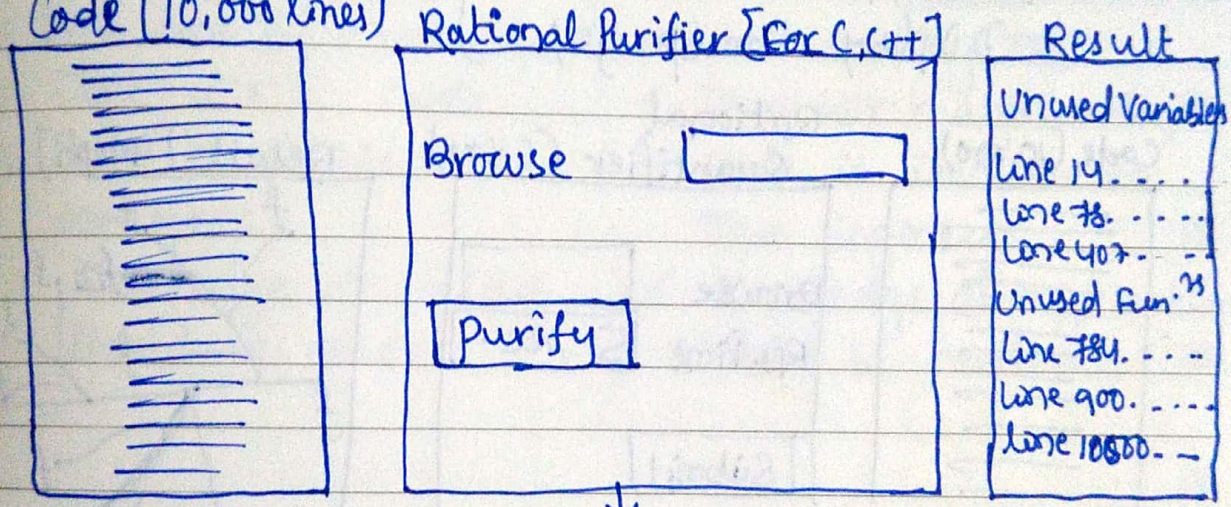
② The developers do all the above testing. Along with all the testing, the developer checks/tests the memory as well as response time of the program also.

Memory:

In this type of developer checks all the unwanted lines in a program & deletes them to reduce the size of the program.

Scanned with CamScanner

Ex: Let Nokia want to make a handset where it wants to put a game with 2GB memory area. He gone to EA sports to built a game. So EA sports should built the required game which should be occupie the within 2GB. If it takes more than 2GB then the game can't be installed. So EA sports should concentrate on the memory of that game.



Here the program is browsed & checks what are the unused things in the program so that memory occupied by the lines in the program is more. Then the developers checks manually if they are truely unused then he reduces the size by deleting all those codes.

Response Time:

Here the developer checks the execution time of the program using some tools.

Scanned with CamScanner

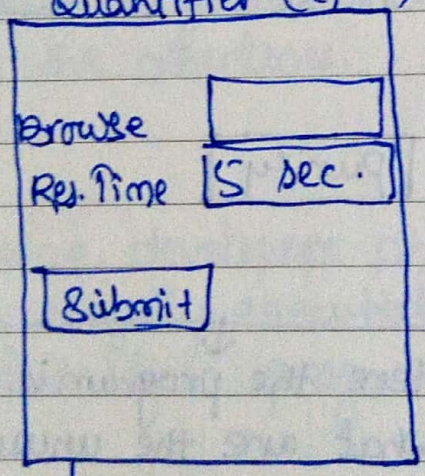
Ex: If 100 no. of users used ~~this~~ the login page & 99 peoples are entering the correct UN & PSW & 1 user enters invalid UN/PSW then it takes a longer time to respond if the code is written as below,

```
If UN/PSW is invalid
    Display error message;
else
    If UN/PSW is valid
        Display homepage;
```

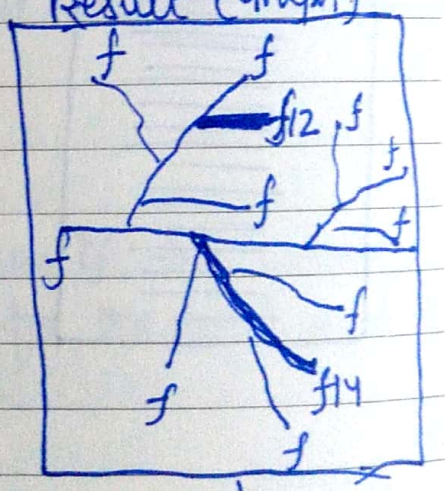
Executed line by line
→ False cond.?
→ True cond.?

Code (10,000)

Rational Quantifier (C, C++)



Result [Graph]



It checks which fun.? takes more time from the given response time & that is detected & shown in the graph as a thick lines.

The thick lines indicates that the fun.ns which are taking more time to respond.

Scanned with CamScanner

Questions on White Box Testing:

1. What is WBT?
2. Explain WBT Process.
3. Explain types of WBT.
4. What is the difference between memory & response time testing done by the developers?
5. What is the difference between WBT & BBT?

Ans:

WBT	BBT
* It is done by the developers.	* It is done by TEs.
* Done w.r.t code	* Done w.r.t application
* Code is visible to developers.	* Code is invisible to Test Engineers.
* Knowledge of coding should be needed.	* Knowledge of application should be needed.

Scanned with CamScanner

Testing :

→ Checking the functionality of an application according to the customer requirement is known as "testing."

→ Testing the application with the intension of finding the bugs or defects is known as "testing".

→ Validating the S/w against the requirement document is known as "testing".

NOTE:

Before starting the testing we should have

(a) The requirement document

(b) we should know the test URL.

(c) we should know the respective ~~as~~ assigned modules.

Scanned with CamScanner

29.4.14

Application Installed

Test Server

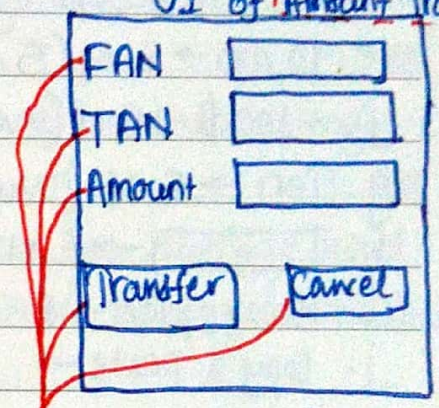
Requirement (SRS) → Coding | WBT | Testing

Customer — Developer — TE

Ex: Requirement

5. Amount Transfer
  5.1. FAN → Text Box
    5.1.1. FAN → Accept only 10 digits
  5.2. TAN → Text Box
    5.2.1. TAN → Accept only 10 digits
  5.3. Amount → Text Box
    5.3.1. Amount → Accept max. 4 digits
  5.4. Transfer → Button → Enabled
  5.5. Cancel → Button, Enabled

UI of Amount Transfer (Module)



Objects or Components — Should be tested

Where FAN → From Account Number
TAN → To Account Number.

1. Testing for FAN:
a. 1234567890 → Accept
b. 1234567899 → Error Msg. (EM) (A/C Not Exist)
c. Blocked A/C → EM
d. Alphanumeric → EM
e. Less than 10 digit → EM
f. More than 10 digit → EM
g. Same as TAN → EM
h. Copy & Paste → EM

2. Testing for TAN

Same as FAN

Scanned with CamScanner

3. Testing for Amount:

a. 9999 → Accept (4 digit)
b. 999 → Accept (3 digit)
c. 97 → Accept (2 digit)
d. 1 → Accept (1 digit)
e. 10,100 → E.M (5 digit)
f. -100$ → E.M (-ve no. & Special characters)
g. Ten → E.M (Characters)
h. [ ] → E.M (Space)
i. Backspace →
j. Copy & paste →

k. 0
l. 99.99
m. [ ]99
n. 099

No Assumption.

o. Amount more than Balance → EM

4. Testing for Transfer: Enter valid FAN, TAN, Amount date Click on 'TRANSFER', we should get a confirm message like "Amount Transfer Successfully".

5. Testing for Cancel: Enter any value for FAN, TAN, Amount & click on 'CANCEL', if all the boxes are cleared then it's ok.

Here in the above case we checked the functionality of each component of the given module.

Functional Testing / Unit Testing / Component Testing:

Checking each & every component of a given module or each & every module of an application individually/independently is known as functional testing.

Scanned with CamScanner

NOTE:

1. Whenever we are checking one component, the data/value in the other components must be +ve valued.
2. Points to be remembered while testing,
   a. Always start with +ve testing i.e start entering the +ve data for testing.
   b. Continue testing with -ve testing after +ve testing i.e enter the -ve data/ i/ps.
   c. During testing process never check for similar kind of values i.e don't perform over testing / exhaustive testing.
   d. During testing process, if the requirements are not clear, we should never assume the requirements in a wrong way where as get it clearified from the concerned person.
   e. Always stop testing whenever we feel the "maximum test coverage" is achieved i.e all the possible value i.e both +ve & -ve values are covered.

30.4.14

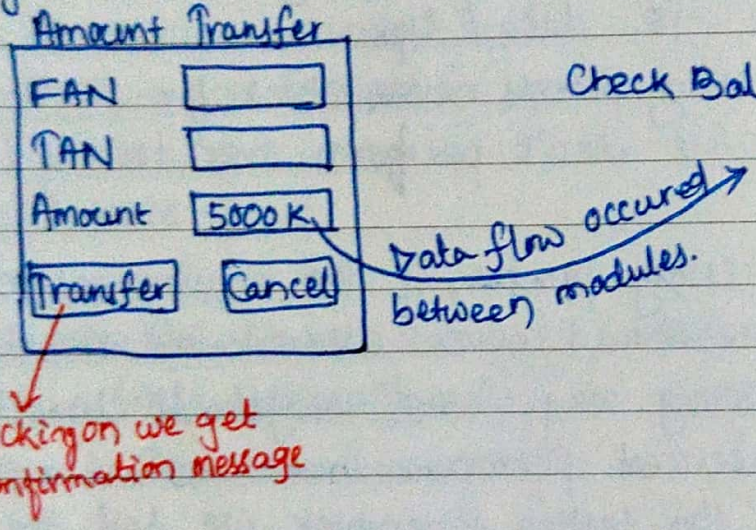
Integration Testing:

→ Checking the dataflow between the modules or the interface between the modules is known as integration testing.

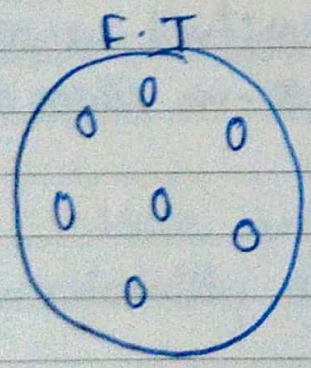
→ In this type of testing we check if the data between the dependent modules are transferred or not.

Login as User 'A' — Login as User 'B'

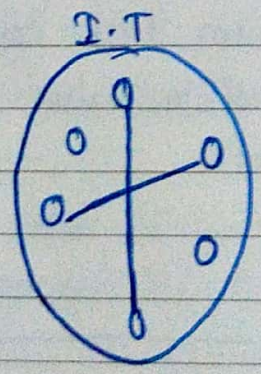
Ex:



Check Bal = Prev. Bal + Recieved Bal. (5000K)



Bugs are more



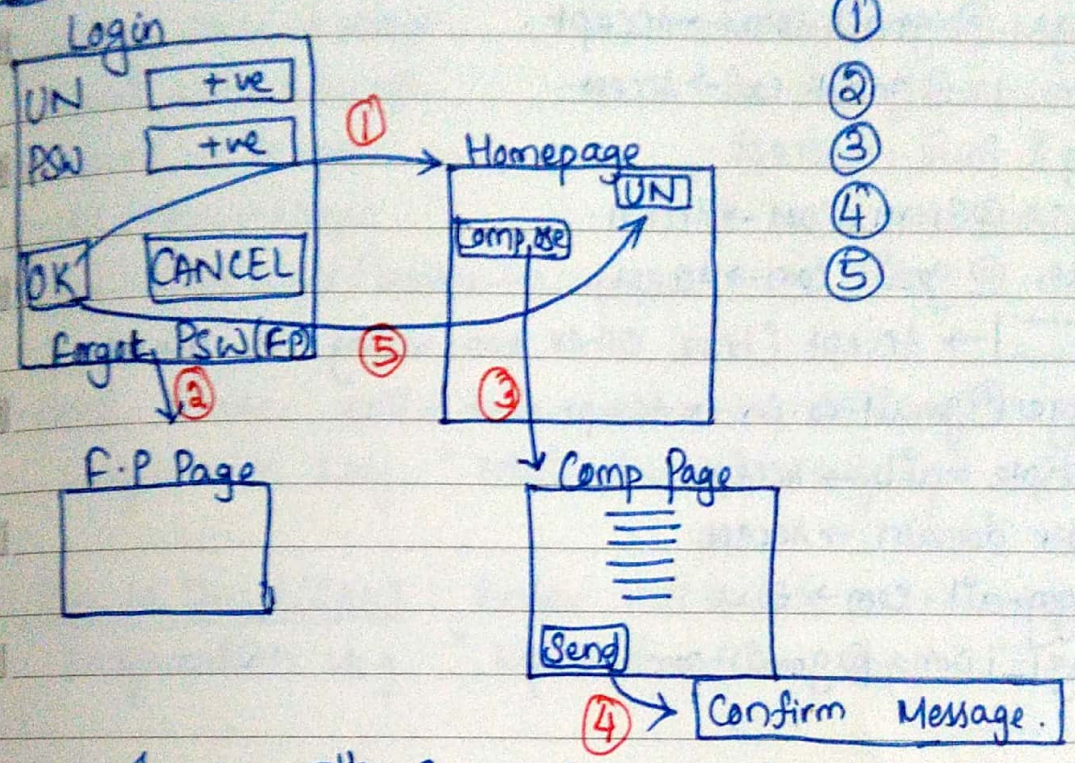
Bugs are less

NOTE:

During testing 1st Functional testing is must then only other testing process will be continued.

Scanned with CamScanner

Ex: Testing Scenario:



Scenario	Type of Testing
1	F.T
2	F.T
3	F.T
4	F.T
5	I.T

In the 5th Scenario the UN which is the date in login page is flown to the ~~homp~~ homepage.

Ex: Functional Testing for Gmail:

Gmail

1. Login
2. Compose
3. Inbox
4. Sent Item (S.I)
5. Draft
6. Thrash
7. Spam
8. Chat
9. Help
10. Logout

Checking the functional test Scenarios for Compose

1. For TO, 2. For CC, 3. For BCC

a. Dinga @ gmail. com → Accept.
b. Dinga_12 @ gmail. com → Accept.
c. Copy & Paste → Accept.
d. DINGA @GMAIL.COM → Accept.
e. 4585 @ gmail. com → Accept.
f. [ ] → Accept (bcoz other two having +ve values)
g. dinga @gmail.co.in → Accept.
h. multiple mails → Accept.
i. Other domain → Accept
j. dingagmail.com → E.M
k. Ding [ ] Dong @gmail.com → EM

4. Subject: 200 max. Char

a. Enter 200 char → Accept.
b. One char → Accept.
c. [ ] → Accept.
d. Copy & Paste → Accept.
e. Backspace → Accept.
f. URL / Link → Accept.
g. more than 200 char → E.M
h. Copy & paste images → E.M

5. Attachment:

a. Browse
b. Different file format
c. Different sizes of files.
d. Select multiple files
e. Remove files
f. Drag & Drop file
g. View Attached file
h. Browse from Different locations
i. Attach opened file
j. Time of attachment.

Scanned with CamScanner

6. Body → 2000 max. chars

a. max 2000 chars → Accept
b. min 1 char → Accept
c. Copy & paste → Accept
d. Blank → Accept.
e. URL / Link → Accept
f. Copy & paste a file / images → Accept.
g. Edit text → Accept
h. Formatting → Accept.
i. Add Signatures → Accept.
j. more than 2000 chars → E.M.

7. Send: Enter valid data, click on SEND
Confirmation Msg. "Mail Sent Successfully".

8. Save as Draft (SAD): Enter valid data, Click on S.A.D
Confirmation Msg. "Mail Saved in Draft".

9. Cancel: Enter data, Click on CANCEL,
All the fields must be reset / clear.

Scanned with CamScanner

1.5.14

NOTES RECORD THE LIST OF THINGS YOU MIGHT WANT TO DO.

Integration Testing for Gmail:

NOTE:

Don't test randomly.

Gmail

1. Login
2. Compose
3. Inbox
4. S.I
5. Draft
6. Thrash
7. Spam
8. Chat
9. Help
10. Logout

1. Login: Enter UN & PSW in Login page, Click on OK & check in homepage for UN is displayed or not.

2. Compose:
   * Compose mail, Send, Check in S.I of Sender & Inbox of Rx.
   * Compose mail, Send to yourself, Check in inbox of Sender.
   * Compose mail, Send to an invalid ID (valid format), check mail in inbox of sender.
   * Compose mail, Click on S.A.D, Check in draft.
   * Compose mail, Close window, check in draft.

3. Inbox:
   * Forward a mail from Inbox, Click on send, Check in S.I & Inbox of Rx.
   * Reply a mail from inbox, click on send, check in S.I & inbox of Rx.

Scanned with CamScanner

* Move a mail, check in Spam.
* Delete a mail from Inbox, Check in Thrash.

* Download a file, Check in saved place (Functional Testing) because file is not saved in any module of Gmail.

4. Sent Items (S.I)

* Forward a mail from S.I, Check in inbox of Rx & S.I of Sender.
* Select a mail from S.I, S.A.D & Check in Draft.
* Move a mail from S.I, Check in spam.
* Delete the mail from S.I, check in thrash.

5. Draft:

* Select the saved mail from draft, Forward, Check in S.I of Sender & inbox of Rx.
* Select a mail from draft, Forward, S.A.D & check in draft.
* Delete the mail from draft, Check in thrash.

6. Thrash:

* Restore a mail from thrash, Check the mail in the module from where it has been deleted i.e from Inbox, from S.I or from draft.

* Delete the mail from thrash & Check in thrash (F.T)

7. Spam:

Scanned with CamScanner

8. Chat:

* Chat & Check in chat history.
* Chat with offline user & check in inbox.

9. Help:

No. I.T for help.

10. Logout:

No I.T for logout.

* Click on Logout, check the login page (F.T).

NOTE:

* Points to be remembered while doing I.T

a. Entire independent modules should be working fine i.e functional testing should be completed.

b. Knowledge on the dataflow between the modules is very much necessary to do I.T.

c. Always start I.T module by module orderly so that we can cover more no. of scenarioes & we don't miss out any scenario.

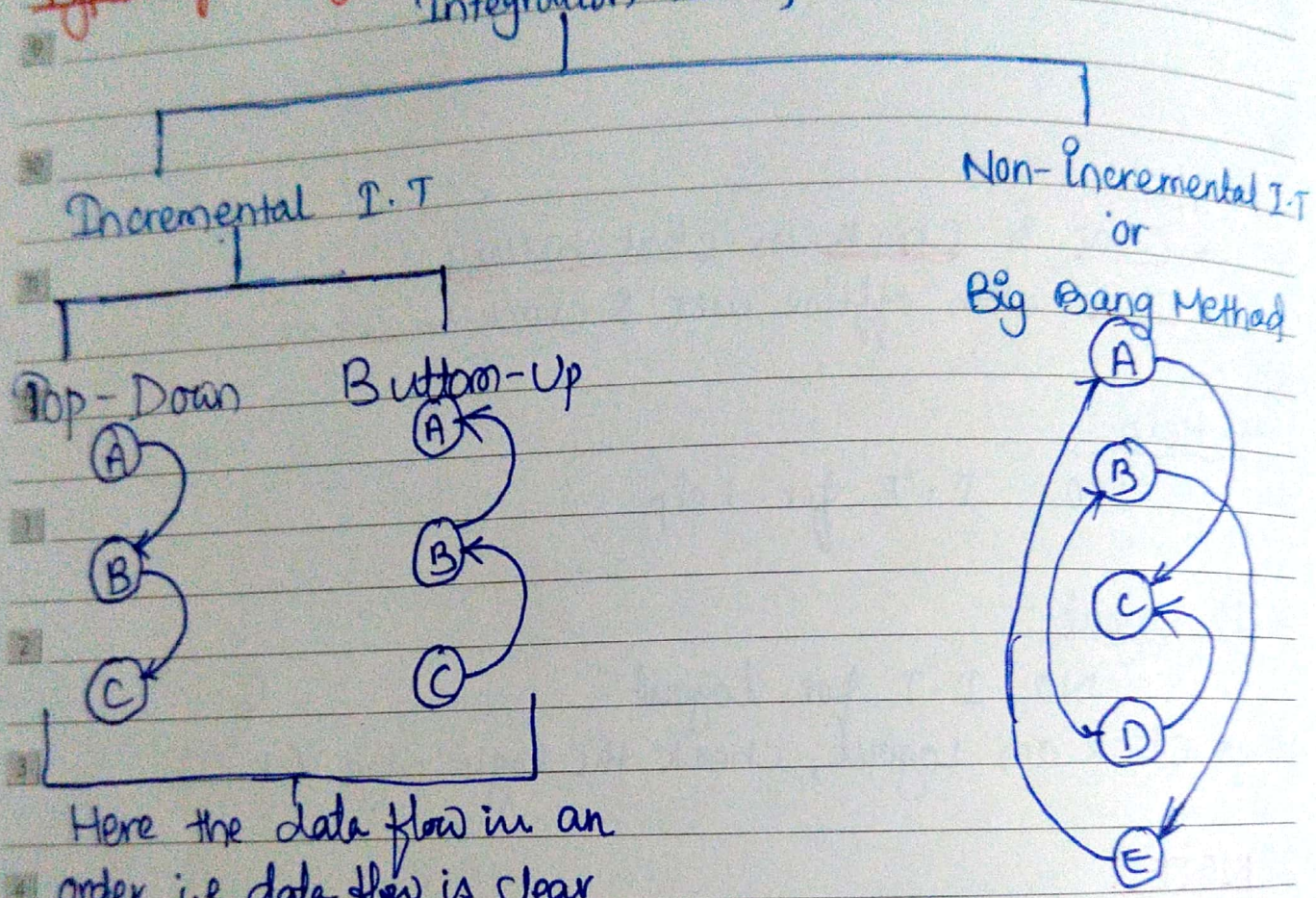
d. F.T on every modules is compulsory where as I.T exist only if there is a dataflow between the modules.

Scanned with CamScanner

2.3.14

04 MARCH MONDAY

Types of Integration Testing:



Here the data flow in an order i.e data flow is clear

Ex: Online Shopping, Booking a ticket from IRCTC.

Here data flow is in random order i.e data flow bet.^n the modules are not clear.

Ex: Gmail.

(1) Incremental I.T:

→ We go for this approach whenever the data flow between the modules is clear.

→ There are two approaches under this,

a. Top-down Approach

In this case, the modules are added one by one from top to the bottom & we check the

data flow in the same order.

b. Bottom-Up Approach:

In this case we add the modules from bottom to up/top & checking the data flow between the modules in the same order.

2) Non incremental I.T:

We go for this approach whenever the data flow between the modules or the relationship between the modules is not clear.

Ex: Incremental I.T

IRCTC

Registration ↵
Login ↵
Home page ↵
Select Train ↵
Book a ticket ↵
Payment ↵
Logout ↵

Data flow in an order / Incremental

Non-incremental I.T

Gmail

Login
Homepage
Compose
Inbox
S.I
Thrash

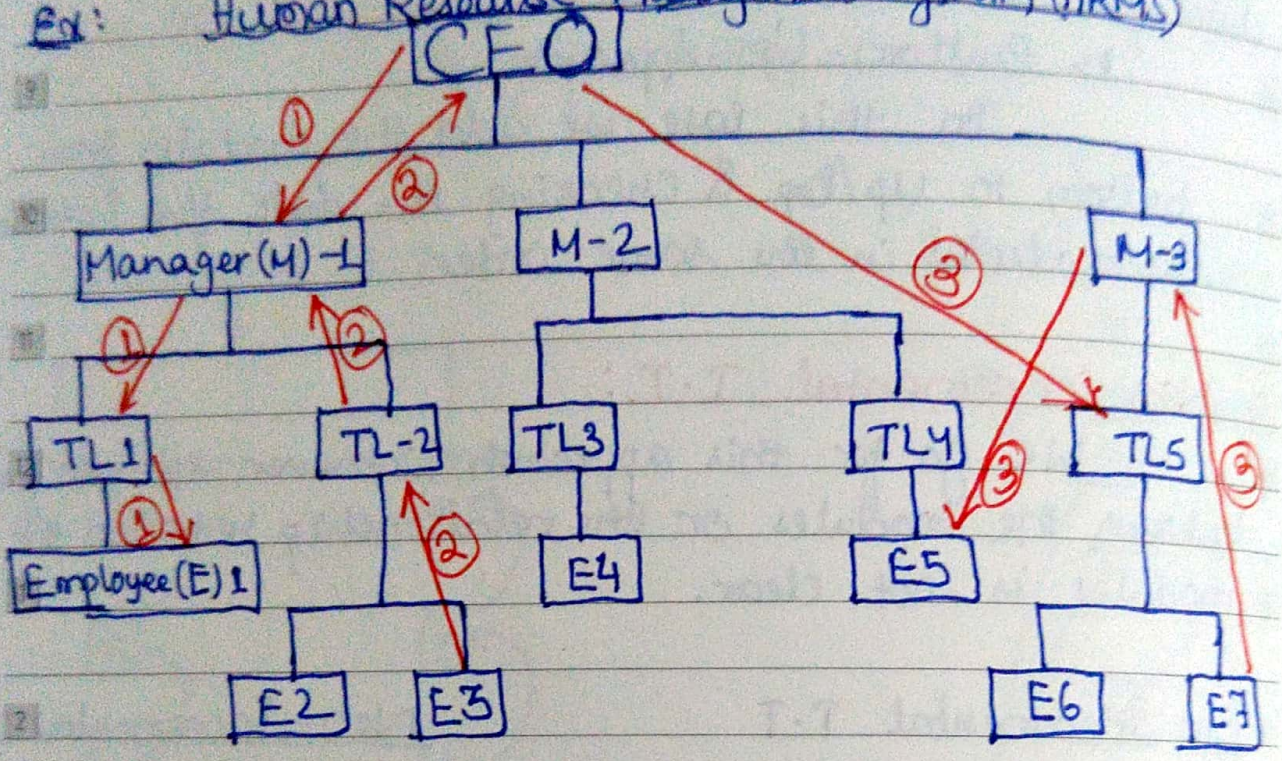
Data flow not in order / random → non-incremental.

NOTE:

Combination of Incremental & Non-incremental Integration Testing is called as "Sandwich Testing".

Scanned with CamScanner

Ex: Human Resource Management System (HRMS)



1, 2, 3 : are the cases.

Cases	Types of Integration Testing
1	Top - Down Incremental Integration Testing
2	Buttom-Up Incremental Integration Testing.
3	Non incremental Integration Testing.

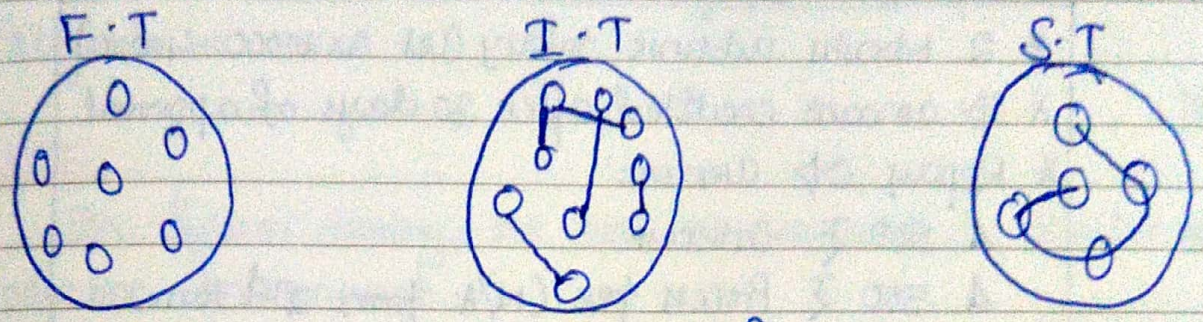
In the above application both the incremental & non-incremental integration testing happens which is called as sandwich testing.

NOTE:

Don't speak about the types of I.T unless & until asked.

Scanned with CamScanner

System Testing:



F·T	I·T	S·T
Modules are tested independently	Data flow bet^n the modules are tested	End-End Flow is tested.

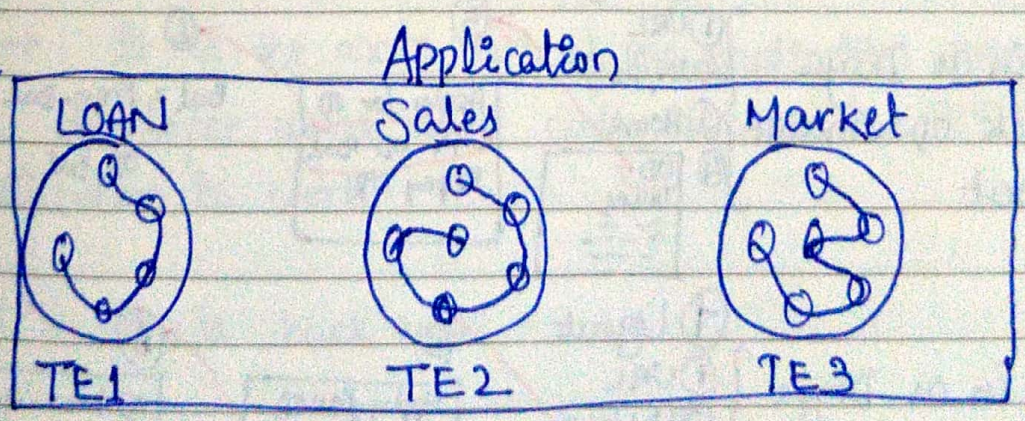
→ Checking the end-to-end flow of an application is known as system testing.

→ In this type we navigate through all the modules of an application & check the application as a whole system.

NOTE:

F·T takes more time as compared to other testing methods.

Ex:



NOTE:

Every test engineer performs F·T, I·T & S·T w.r.t their modules because all the submodules are tested end-end in the main module.

Scanned with CamScanner

6.5.14

Example of S.T

Let 6. OD (over Draft) Feature

* 2 Months advance Salary (Let 2x20000=40000)
* OD amount credited after 30 days of approval
* Repay OD amount
  - 500 ₹ interest
  - 500 ₹ Process fee (N/A from 2nd time

Scenario-1

1. Login as dinga
2. Apply for OD
3. Logout

① URL
② UN/PSW
③ Home
④ OD, Loans, Sales
⑤ OD page: Apply for OD, Check OD Bal, Repay OD
⑥ Apply for OD: Emp ID, Sal, OD Amt, Apply, Cancel
Confirmation Message
⑦ Logout.

4. Log in as Manager
5. Approve OD of Dinga
6. Logout

Wait for 30 days.

① URL
② UN/PSW
③ Home
④ OD, Loans, Sales
⑤ Apply for OD, Chk OD Bal, Repay OD, Approve OD
⑥ Dinga 40K, ---- 50K, ---- 60K, Approve, Reject
⑦ Logout.

7. Login as Dinga
8. Check OD Amount
9. Logout

① URL
② UN/PSW
③ Home
④ OD, Loans
⑤ Apply for OD, Chk OD Bal, Repay OD
⑥ Bal = Prev. Bal + OD Bal
⑦ Logout

10. Log in as Dinga
11. Repay Amount
12. Logout

① URL
② UN/PSW
③ Home
④ OD
⑤ Apply for OD, Chk OD Bal, Repay OD
⑥ OD Amt 40000, Int. 500, P.F 500, 41,000, Repay
Conf. Msg.
⑦ Logout

Scanned with CamScanner

Scenario-2

Repeat 1 to 11 & Pay 40,500 at 11

12 Logout

NOTE:

In System testing, the test environment is similar to the production environment.

Parts of Application:

1. User Interface (UI):

→ It acts as the interface bet.n the user & the application.

→ Simply, it can be called as pages.

→ It is front end.

2. Business Logic (BL):

→ It deals with the codes.

→ It is the main part of the application where the developer write some codes to make the application to work.

→ It is back end.

3. Data Base (DB):

It is the backend of an application where all the user informations are stored in the form of table.

Scanned with CamScanner

NOTE:

Whenever the things are visible it is called as frontend & the things which are invisible is called as backend.

There are two types of UI.

UI

G.U.I (Graphical User Interface)	Command Line Interface (CLI)
→ It is user friendly.	→ It is not user friendly
→ It is not secured.	→ It is ~~highly~~ secured
→ Normally used in PCs.	→ Used in Server always
→ Ex: Windows.	→ Ex: UNIX

NOTE:

In server we install/use Unix since it is secured.

Types of Application:

1. Stand Alone Application.
2. Client-Server Application
3. Web Application: (Recent market based on)

Scanned with CamScanner

1. Stand Alone Application:

It is a S/w which is completely installed on end users machine & only one user can access at a time.

NOTE:

Generally Stand Alone Applications are ".exe" files.

Ex: MS office, Antivirus, Media player, VLC player, OS.

Advantages:

→ Faster in access

→ Security
  ⓐ Hacking is not Possible
  ⓑ Virus is not affected.

Disadvantages:

→ Single user can access.
→ Installion is required
→ End user's system memory utilized.
→ Data Sharing is not possible.
→ Maintainance is tough.

Ex: Suppose a S/w developed by an Indian company. It is good marketed so that many customers (lets say 1000) from U.S.A, U.K & U.P bought the S/w. Suppose the client got an issue. So the company will leave the employee to fix the issue. In this way he spent a lot for the on site employees. In this way the company will be closed in fewer days.

Scanned with CamScanner

9.5.14

2. Client-Server Application:

→ In this the entire S/W is divided into two parts where one part is installed at end user's / Client's machine & the other part is installed at the server machine.

→ The client & Server communicate with each other for the application to work.

Advantages:

→ Multiple user can access.

→ Data sharing is possible.

→ Faster in access : It depends on bandwidth of internet.

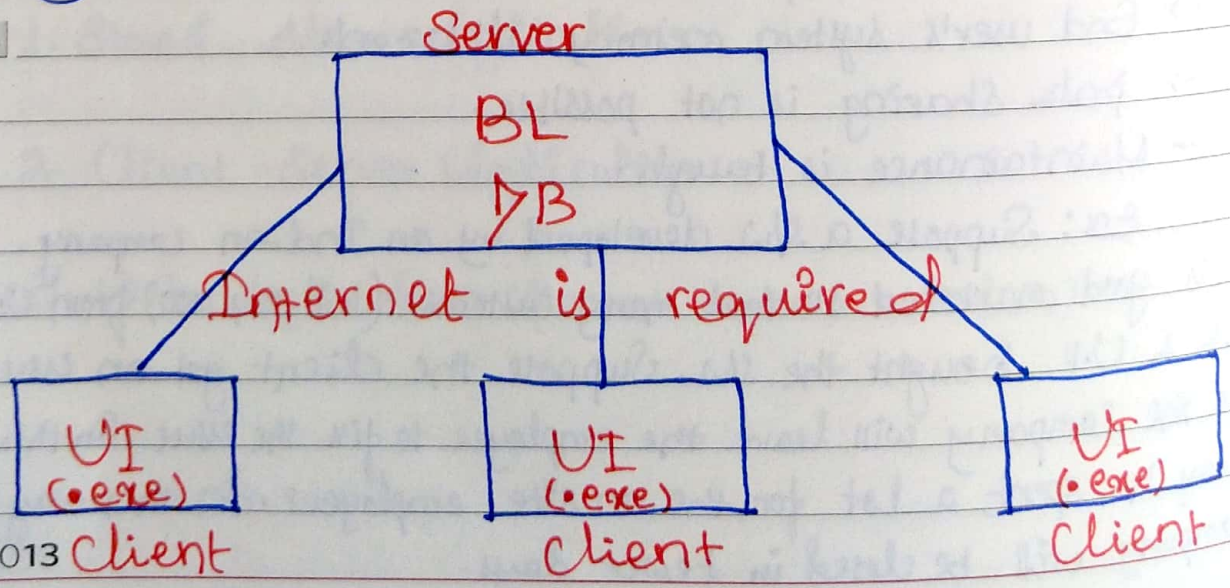
→ Security

ⓐ Hacking $\xrightarrow{\text{Sol.?}}$ Security Testing.

ⓑ Virus $\xrightarrow{\text{Sol.?}}$ Antivirus.

Disadvantage

→ If the server fails, no user can access.



Scanned with CamScanner

Ex: Gtalk, Yahoo messanger, Skype.

3. Web Application:

It is a S/w where the entire application is installed at the server & the end user can access with the help of the URL.

Advantages

→ Faster in access : completely depends on bandwidth of internet.

→ Data sharing is possible.

→ Multiple user can access.

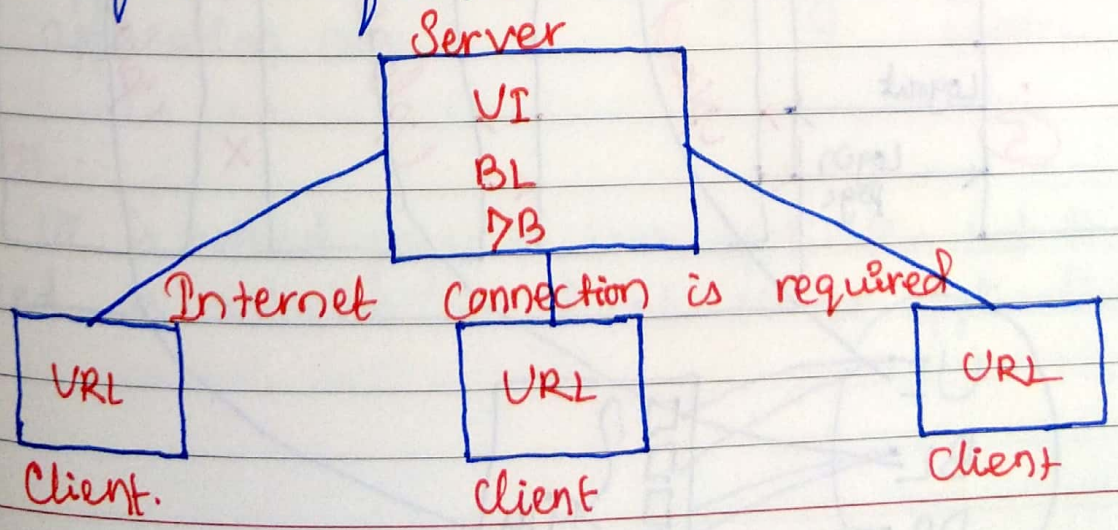
→ Security :

a. Hacking $\xrightarrow{\text{Sol.?}}$ Security Testing.

b. Virus $\xrightarrow{\text{Sol.?}}$ Antivirus.

Disadvantage:

→ If server fails, no user can access.



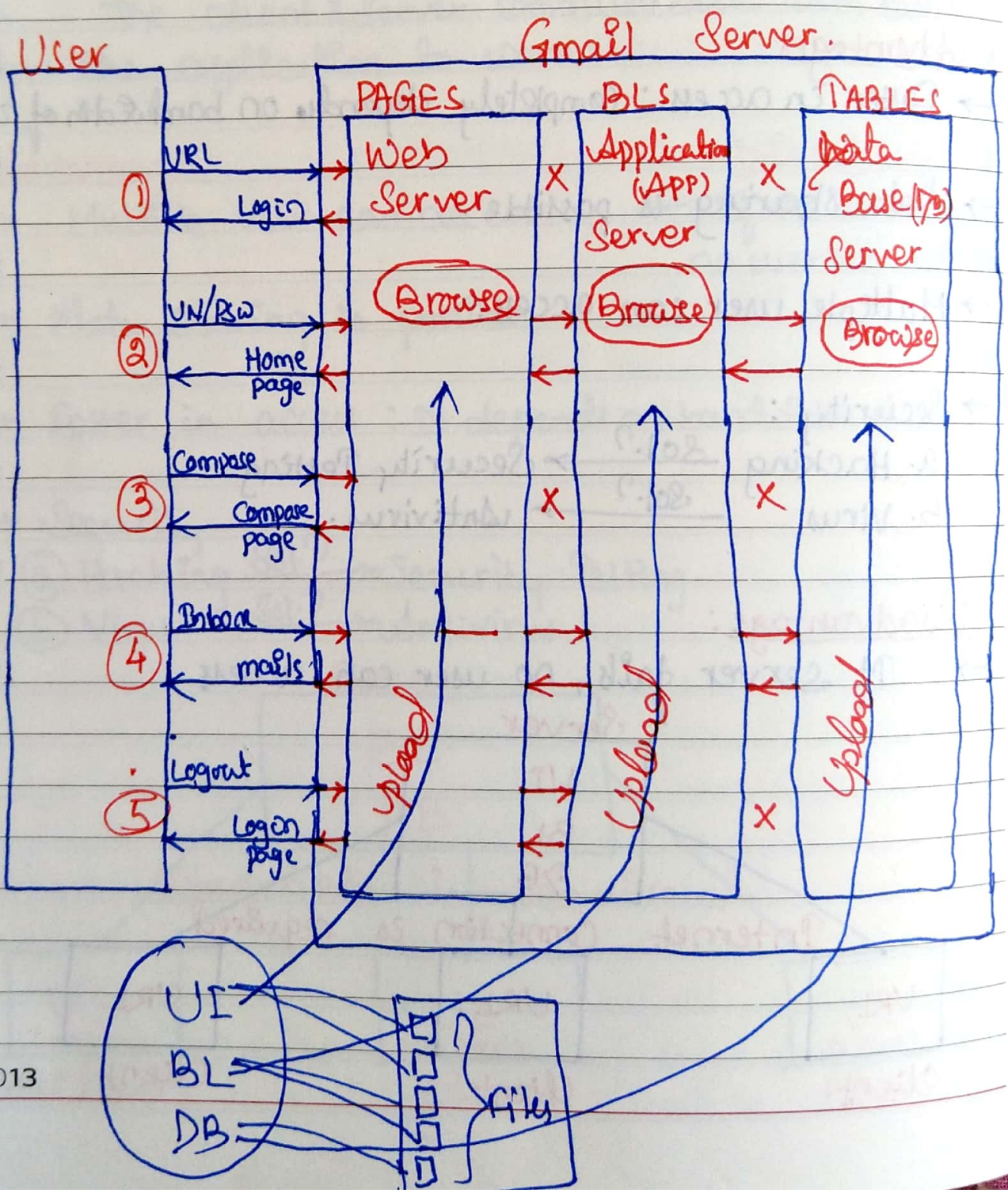
Scanned with CamScanner

Ex: Yahoo, Gmail, Facebook.

NOTE:

Browsers are stand alone application.

Web Application Architecture:



Scanned with CamScanner

Advantages :

→ Faster

→ Load is less.

NOTE :

1. Generally web applications consist of S/w like webservers, Application server & DB Server.

2. a. All the pages related to the applications are uploaded on to the web server.

b. All the business logic functions related to the application are installed on to the application server.

c. The entire tables related to the applications & the user data is stored in the DB server.

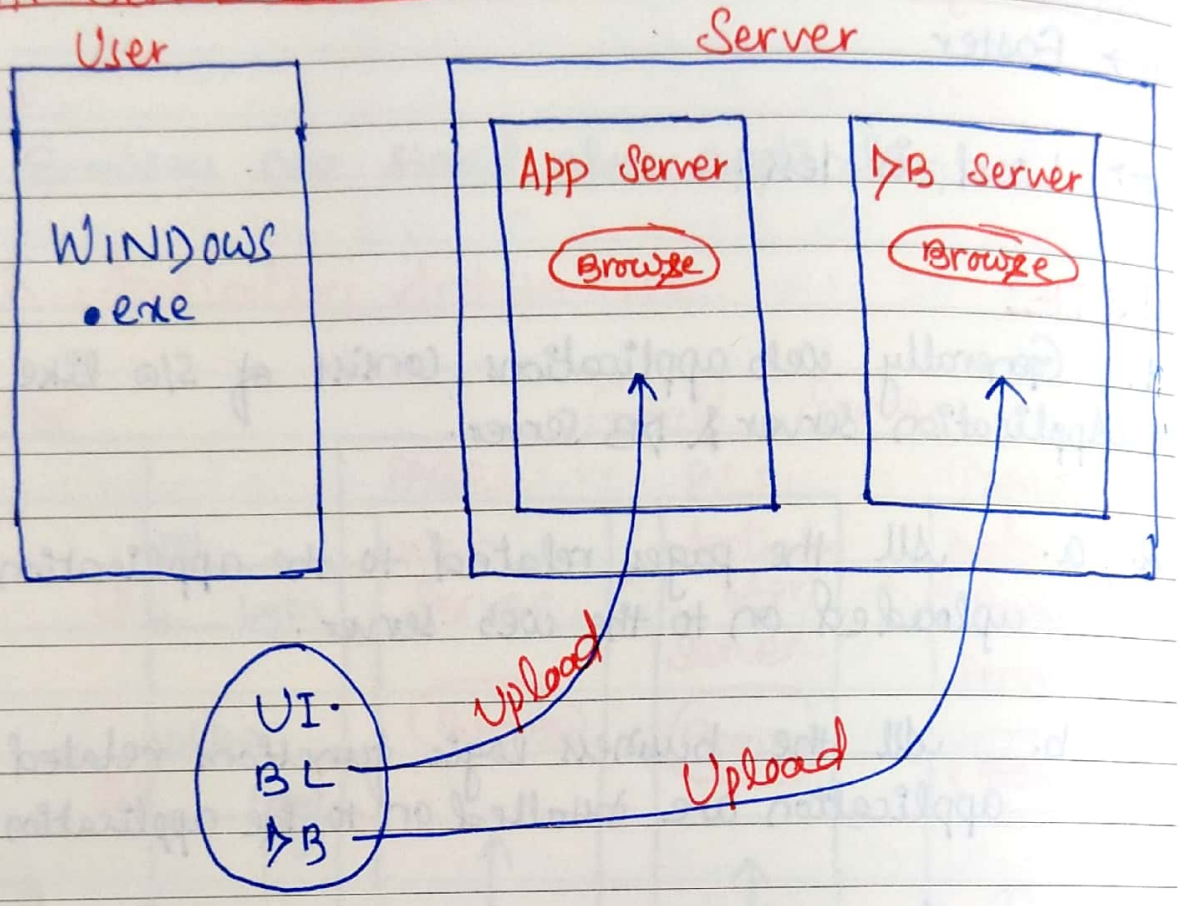
Ex: Online calculator is related to both web server & application server. Similarly logout of an application also.

NOTE :

UI is called pages in web application but it is called as window in client-server application.

Scanned with CamScanner

Client-Server Architecture:



Assignment:

Find the list of web servers, App servers & DB Servers.

Ans: Web Servers

1. Apache
2. Apache Tomcat
3. MS's Internet Information Services (IIS) windows server.
4. Nginx web server.
5. Lighttpd
6. Jigsaw
7. Klone
8. Abyss.
9. Oracle web tier
10. Xitami (X5)
11. Zeus
12. Sun Java System.

Scanned with CamScanner

Application Server

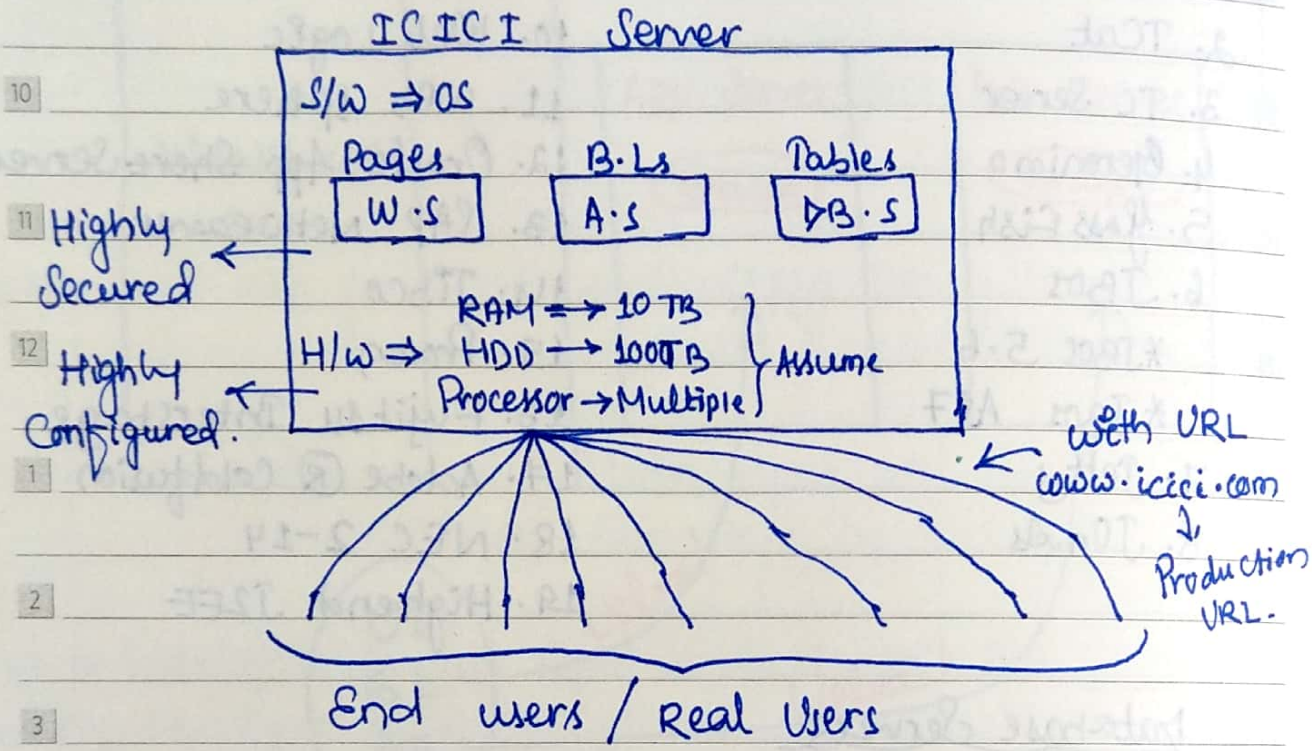
1. Tomcat
2. TCat
3. TC Server
4. Geronimo
5. Glass Fish
6. JBos
   * JBOS 5.6
   * JBOS AS7
7. Jetty
8. JOnAs
9. Resin
10. Web Logic
11. web sphere
12. Oracle App ~~Shere~~ Server
13. SAP Netweaver
14. Tibco
15. Proven
16. Fujitsu Interstage
17. Adobe ® Coldfusion
18. NEC 2-14
19. Highend J2EE

Database Servers

1. MS SQL
2. ORACLE
3. Informix
4. DB2
5. Sybase
6. Postgress
7. MySQL.

Scanned with CamScanner

Production/Real/Live Server/Environment :



→ Production environment is an environment (or) server where entire stable application is installed & accessed by the real users with the help of production URL.

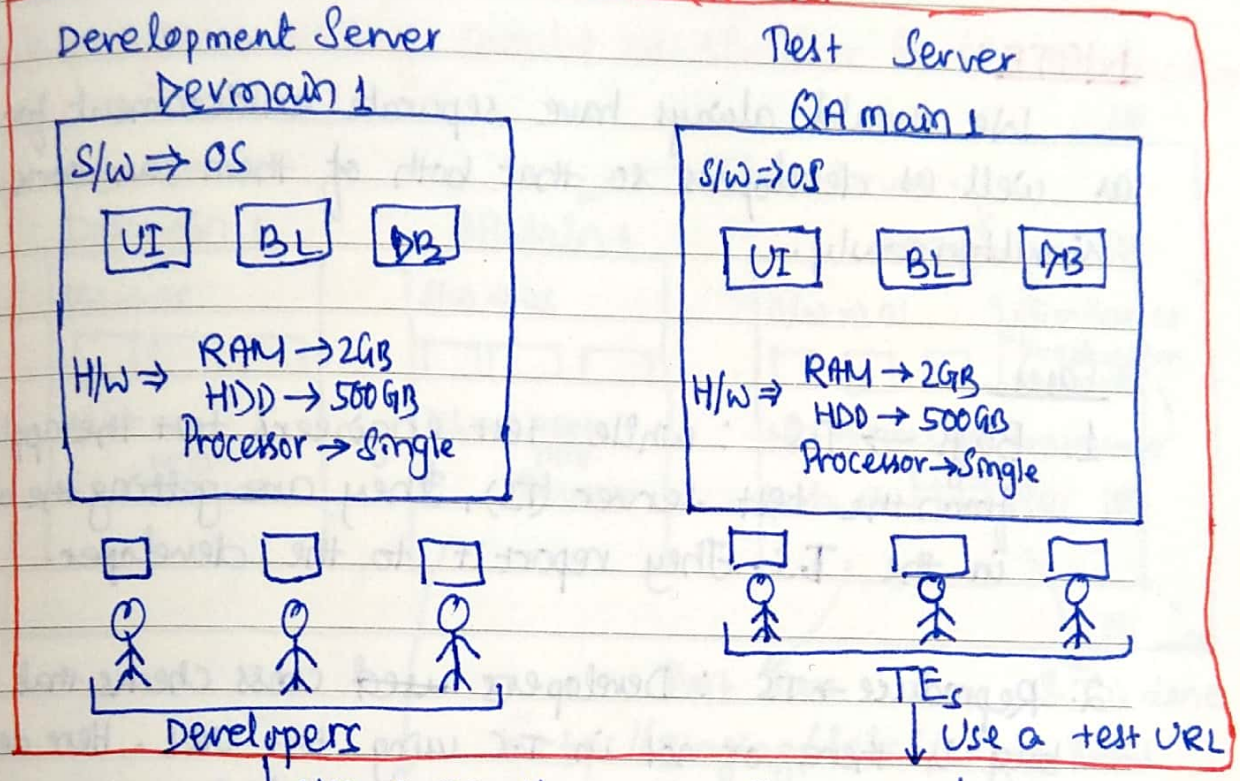
→ Production environment is highly configured always since it is checked by 'n' no. of users.

→ Production environment is always secured since the real users data is stored in the server.

→ Real users always access the application with the help of production URL.

Scanned with CamScanner

WIPRO (Assume)



↓ Use a DEV. URL

http://Server_name/proj_name/comp._name.local

Ex: http://devmain1/icici/wipro.local

↓ Use a test URL

http://Server_name/proj_name/comp.name.local

Ex: http://qamain1/icici/wipro.local.

Ex: Let ICICI need a product & gone to wipro. Wipro asked for the requirements & ICICI gave him. After that developers use a development server ~~to ent~~ where UI, BLs & DBs are there. They upload all the files in the corresponding places so that can work. Next the application is loaded in the server of the TE which is called as test server. TEs start testing the functionality of the application. If they found any bug they report to the development team. All the developers & TEs are assigned with a system with a correspondent URL.

Scanned with CamScanner

URL : Uniform Resource Locator.

NOTE:

We should always have separate environment for TE as well as developers so that both of them can work simultaneously.

Cases

1. Bugs → T.S : while test engineers test the application from the test server (TS), they are getting the bug in the T.S. They report it to the developer.

2. Reproduce → TS : Developers ~~used~~ cross checks that the bug is there or not in T.S using test URL. Here developers are not authorised to modify any data.

3. Fix → Development Server (DS) : Developers modify the BL & fixes the bug as per the requirement in the D.S. This BL is ~~agin~~ again loaded to the T.S.

4. Retest → T.S : Here the modified application is retested in T.S.

5. Stable → T.S : The application is tested untill it's stable

6. Installed in production server.

NOTE:

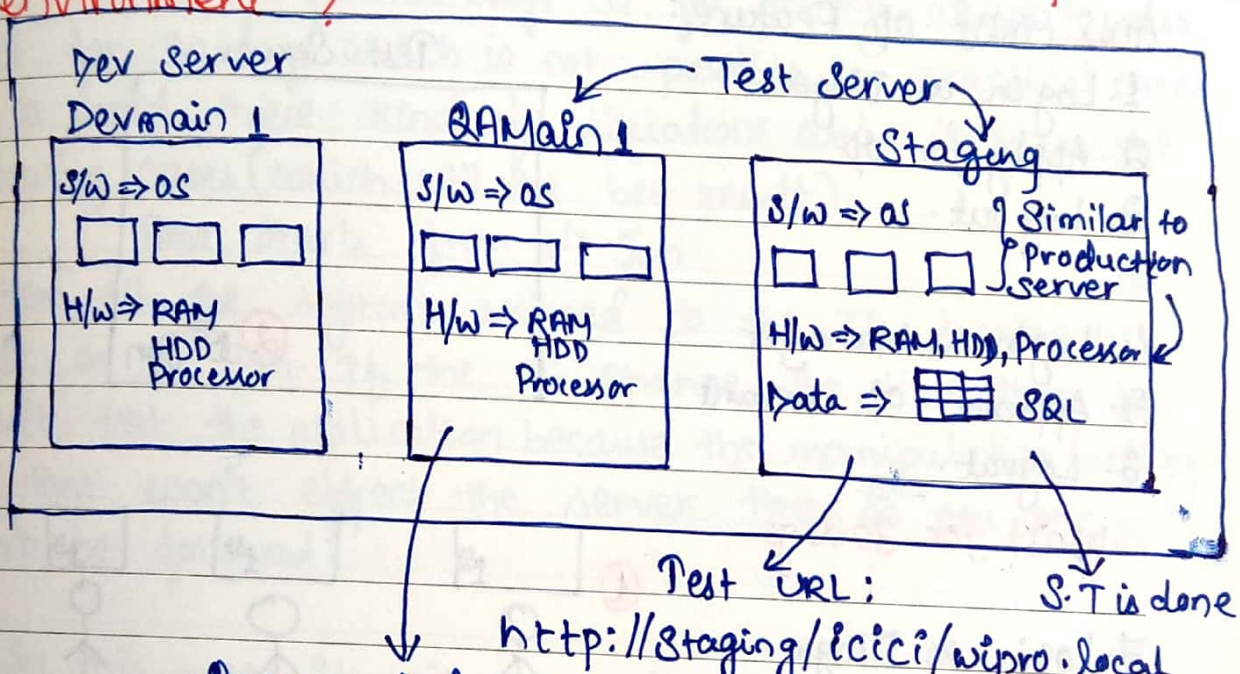
1. DS is used by the developers for writing BL & bug fix.

2. TS is used by the TEs for testing the application & used by the developers for reproducing the bug.

Scanned with CamScanner

Q. Why system testing should be similar to the production environment ?

Ans:



One round of F.T & I.T done

Once the application is tested in a smaller environment & directly moved onto the higher environment or highly configured environment, we may face some issues. To avoid this we do one round of end-to-end testing which is called as system testing on an environment which is similar to production environment.

NOTE:

SUNDAY 24

Normally in QAmain1 test server one round of F.T & I.T is done where as in staging server the system test is done.

Scanned with CamScanner

Q. How can we avoid the waiting cases during testing?

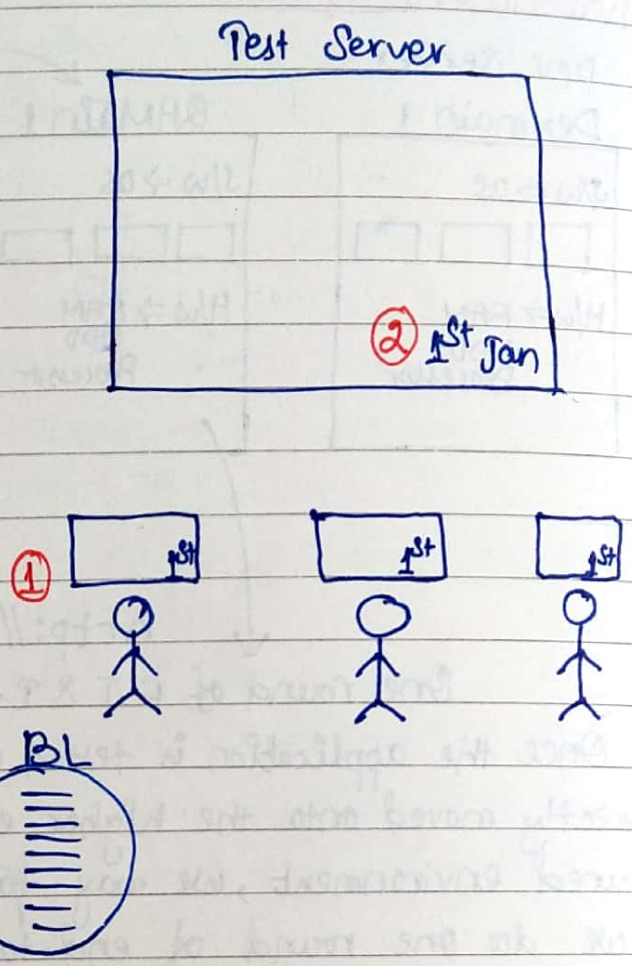
Ans: End of OD Feature:

1. Log in as Dinga
2. Apply for OD
3. Log out.

4. Login as Manager
5. Approve OD amount
6. Logout.

Wait for 30 days

7. Login as Dinga
8. Check OD Balance
9. Logout



Data Base (Collection of Tables)

* Login →

	UN	PSW	email
R1	A	–	–
R2	B	–	–
R3	C	–	–

* A/C Summary
* Loans
* Sales
* OD →

4

	User	Current Date	Apply Date	Approve Date
R1	Dinga	1st	1st	1st
R2	Manga	1st	1st	1st
R3	–	–	–	–

Comparision takes place in between these two columns for approving the OD.

Scanned with CamScanner

From / In the above example to get the op amount we've to wait for 30 days. So for testing we've to also wait for 30 days which is not possible in practical cases. So to avoid these kind of situations we've to follow the following cases (which will the best result)

Time starts from 1st Jan

① Here all the system assigned to the TEs having the date as 1st Jan. If the TE change the time then he can't test the application because the manipulation / setting in time won't effect the server time. So this one can't be followed.

② In this case it will be possible that we can change the date in the server, so that the TE assigned with wait module can test but it may affect to all other TEs. So this is also not prefferred.

③ Here if we'll change the BL as for 2sec. It may work but it may not work same for 30 days. So we've to again wait for 30 days. This is also the failure one.

④ Here in this case there is always a comparision between current date & approve date.

Let current date = Approve date = 1st Jan

Now current date will change as per the day change but the approve date remains same. When the current date will be 31st Jan, it has taken 30 days & OD amount is credited in the customer's account.

Scanned with CamScanner

Here we can change the current date by firing the SQL query,

Update set current date = 31st Jan where UN='DINGA';

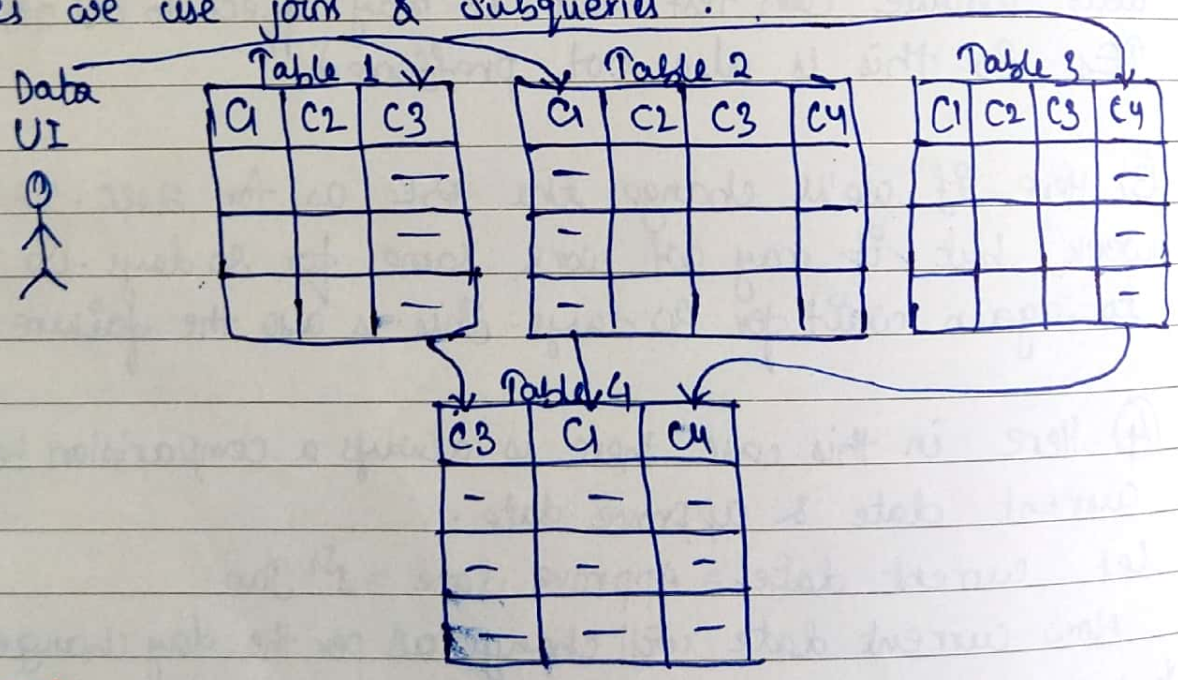
so that it won't effect the others & we can continue with out corresponding testing process.

Q. What is the use of SQL in testing?

Ans: (I) Many times during testing process we may have to wait for a particular duration. These cases we use SQL for updatation.

(II) Many situations we need to cross check the data created in the UI is properly stored in the backend or not. These cases we go for SQL queries.

(III) Sometimes we need to fetch data from different tables & see the result in one table to make the process easier. These cases we use "joins" & "Subqueries".



NOTE:

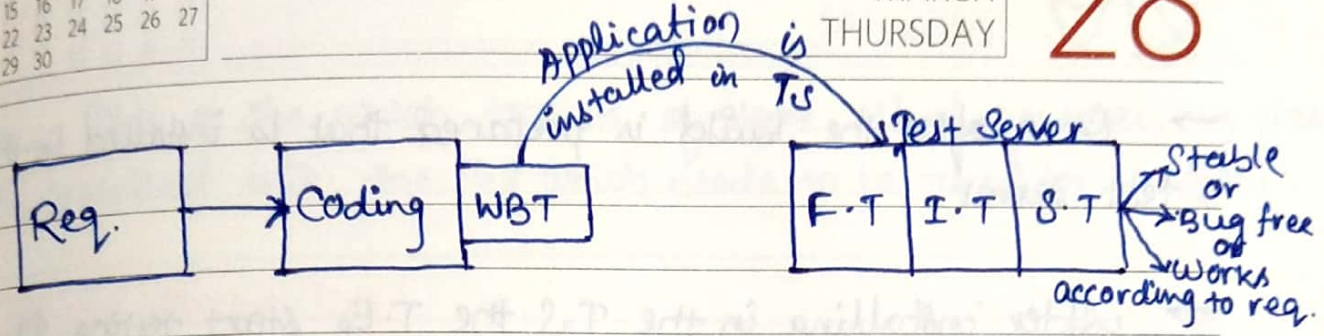
Pre-requirements of a TEs.

2013 ① Req. ② Test URL ③ Assigned modules

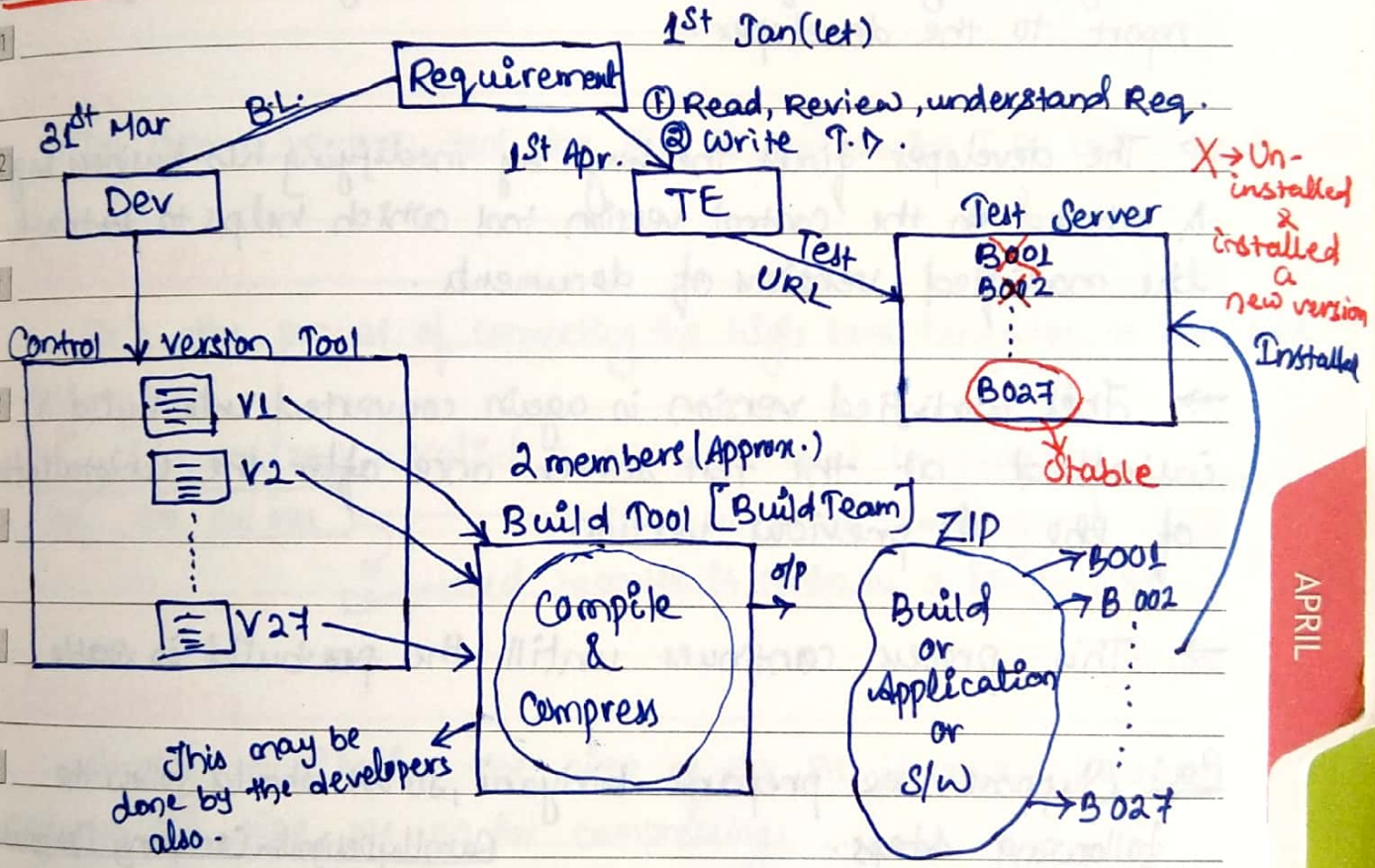
④ DB Schema : Table structure.

Scanned with CamScanner

14.5.14



Build Process:



→ Once the requirements are collected from the customer, the developers ~~are~~ start writing the business logic where as the test engineers read, review & understand the requirement & preparing the test documents.

→ Once the developers complete their business logic they store that in the control version tool.

→ The build team collects the ~~files~~ files & ~~make~~ prepare the build.

→ Once after the build is prepared that is installed in the test server.

→ After installing in the T.S the T.Es start testing. If during testing they encounter some bug then they immidiately report to the developer.

→ The developer fixes the bug by modifying his business logic & stored in the control version tool which helps to ~~to~~ track the modified versions of documents.

→ This modified version is again converted into build & installed at the test server once after the unistallation of the 1st previous version.

→ This process continues untill the ~~pre~~ build is stable.

Ex: Suppose we prepare biriyani, so we should follow the following steps.

	Family Designation	Company Designation
→ Collection of requirements →		BA.
→ Collecting the ingradients →	PAA	Dev
→ Cooking →	MAA	Build
→ Testing →	Jobless & spider students	TEs
→ Eating →	Family	client.

Build:

It is a s/w which is converted ~~for~~ from coding to the application format.

or, It's a S/w which consists of some set of features, bug fixes & installed into the T.S which needs to be tested for stability.

Control Version Tool:

These are the S/ws which are used to keep track of the changes.

NOTE:

In Control version tool the BL as well as the T.Ds are stored.

Compilation:

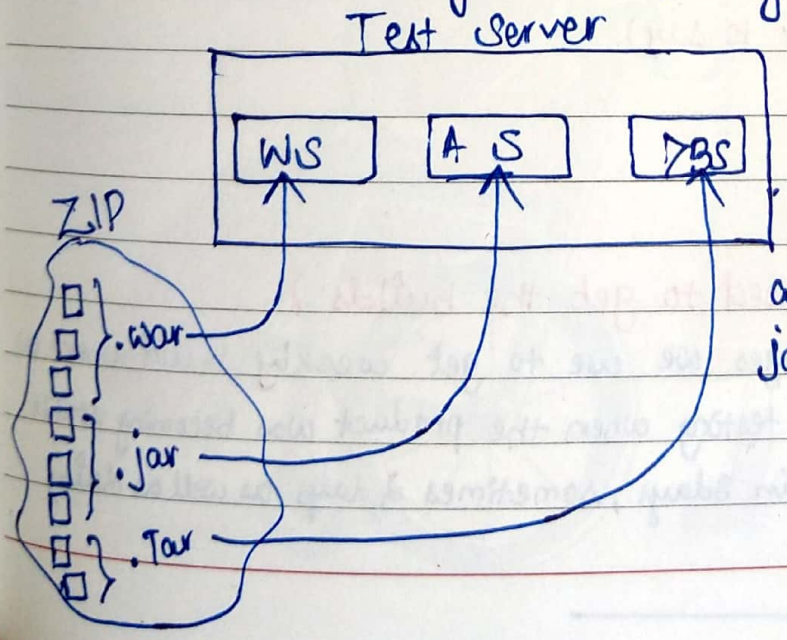
It's the process of converting the high level language to low level language.

Ex: If UN/PSW = valid ; → High Level Language] compilation
1001 1100 1101 0001 → Low Level Language ←

compressed because it contains a large size.

Compressing:

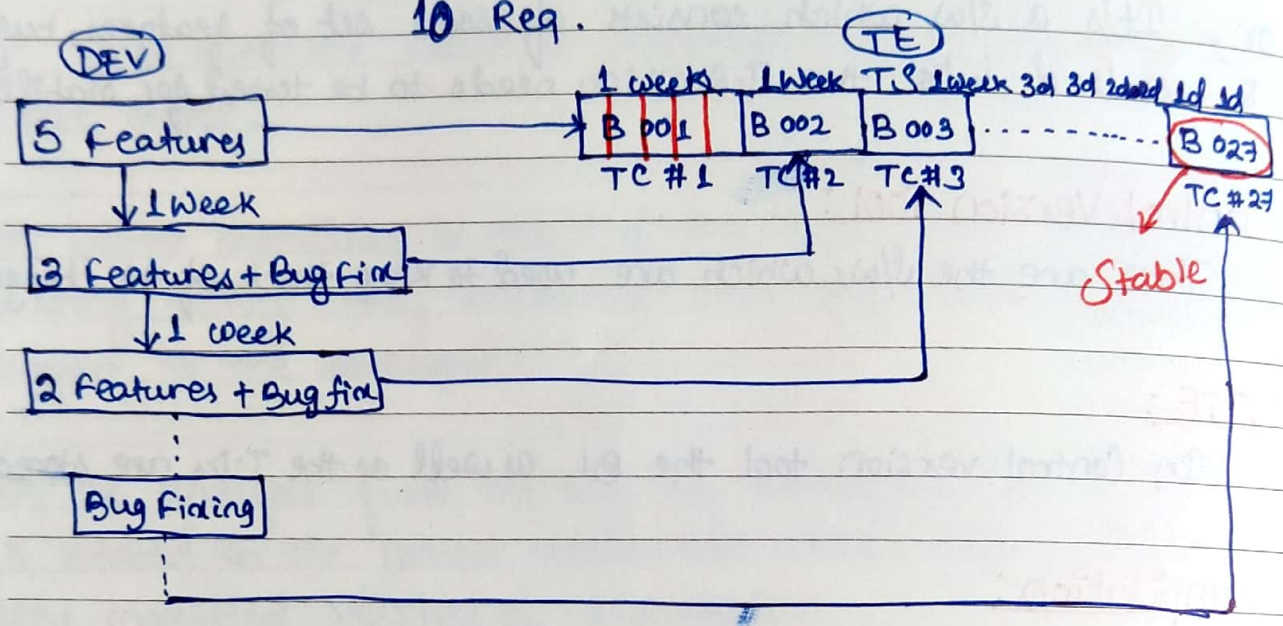
During compilation the size of the BL increases, so to reduce the size we go for compressing.



war → web archieve
jar → Java archieve

Scanned with CamScanner

Ex: Suppose 10 requirements are there.



Test Cycle (TC):

→ It's the time duration to test each build. ~~is k~~

→ Each new build is the modified version of the old build.

NOTE:

Normally when a build is prepared that is installed by the following members,

1. Build Team (Prefer to say)
2. Dev Lead
3. Test Lead

Q. How regularly we used to get the builds?

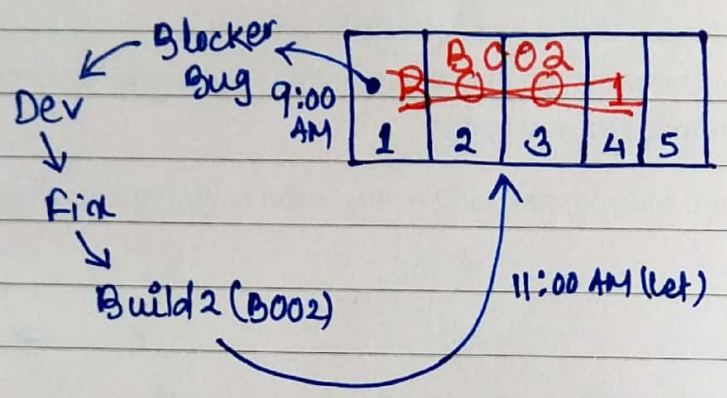
Ans: In the initial stages we use to get weekly build where as in the later stages of testing when the product was becoming stable we use to get once in 3days, sometimes 2 days as well as daily builds.

Scanned with CamScanner

Re-Spin:

→ The process of getting more than one build in the same test cycle is known as respin.

→ It is the permanant sol.ⁿ but time consuming.



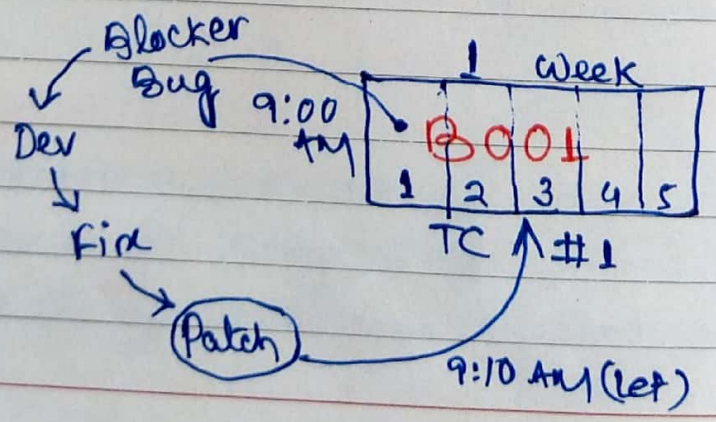
NOTE:

Blocker Bug: It is the bug for which the TE can't move forward with his testing.

Patch:

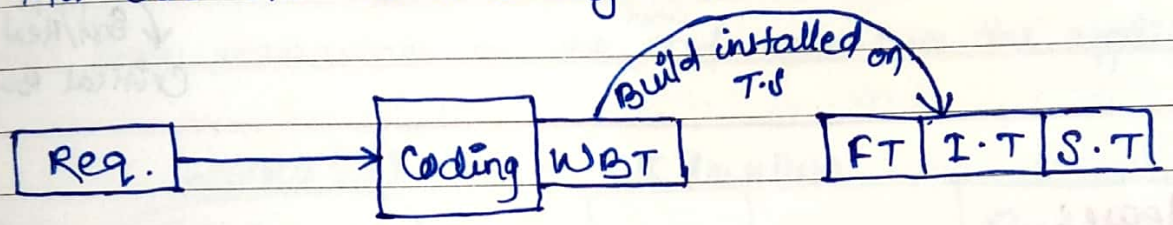
→ It is a s/w which acts as an immediate fix for blocker issues.

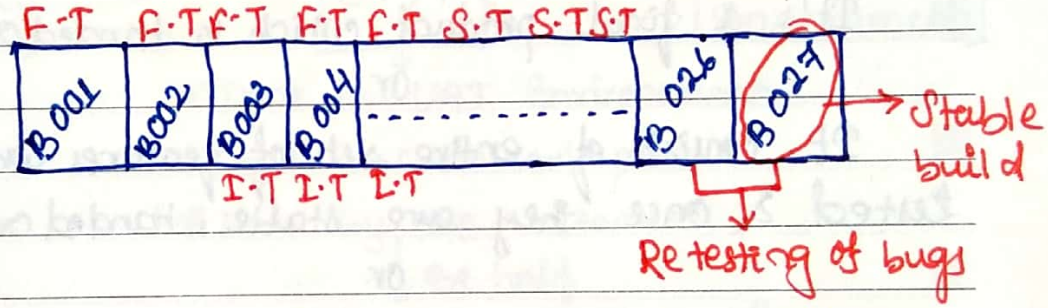
→ It is the temporary sol.ⁿ but time saving.



Scanned with CamScanner

Ex: Suppose a tube is having a small hole, so if we'll go to the puncture shop then the workboy will patch it so that the tube can work as before. If the tube has a cut portion in a large area which can't be patched. So in this case he has to change the entire tube.

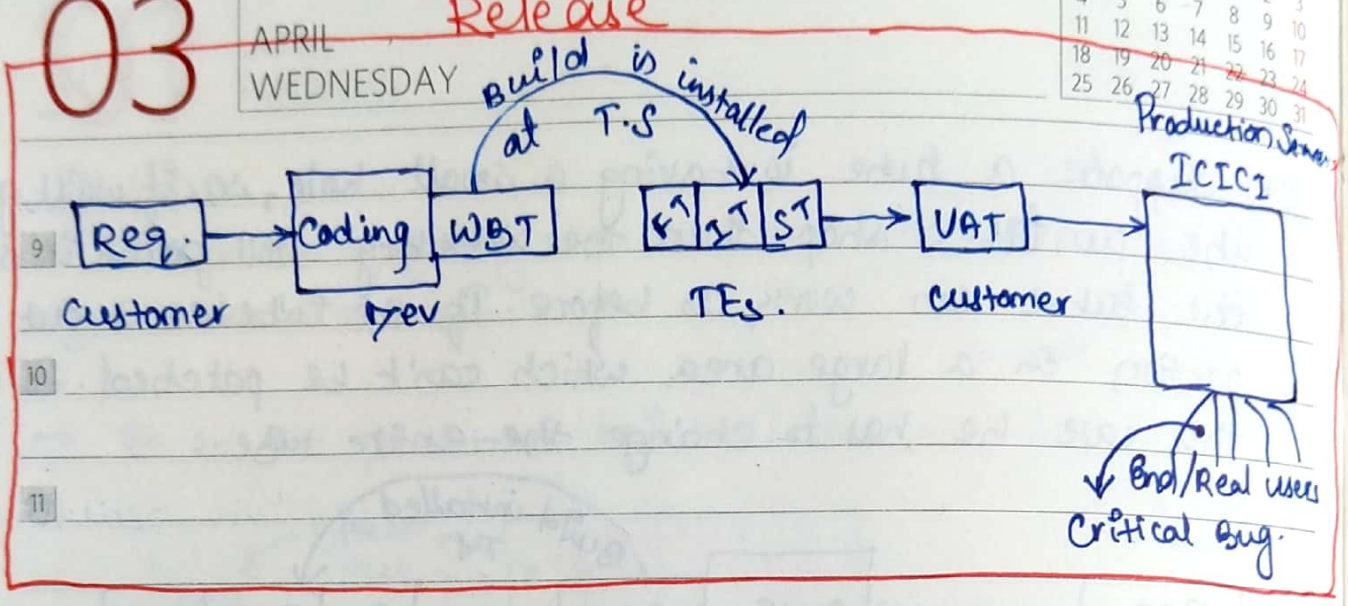




Here in the above diagram in some few builds the F.T is done. When some few features are developed so at that time the T.E do F.T as well as the I.T. When all the requirements are developed, there as soon as F.T & I.T is over the T.Es are busy in the S.T. In last some few builds all kind of testing shall be completed & at that time the T.E check one round of the retesting of the bugs. In the last build the build is stable & it can be handed over to the customer.

But if after handing over to the customer & the real users find some critical bug then it may create some issue for the company as well as for the T.E. So we've to go for one round of UAT.

Scanned with CamScanner



Release:

It's a final product which is handed over to the customer.

or

It consists of entire set of features which are developed, tested & once they are stable, handed over to the customer.

or

It consists of all the activities right from requirements, coding, testing & handing over the product to the customer.

User Acceptance Test (UAT):

→ It is a type of testing done by the customers before accepting the product.

→ Generally UAT is done for customer satisfaction.

Ex: Let we go for buying a Chappal. While we choose a Chappal at that time we check for all activities like is there any damage or not, it's slipping or not, by pulling the stiches we check ~~edit~~ whether the stiches are loosing or not. The above all the things done

Scanned with CamScanner

because of the satisfaction. In this way while buying a product the customer always check the product for their satisfaction.

→ Here customer checks the business scenarioes or the real time scenarioes or ~~the~~ how end user uses the application.

WIPRO (TS)

Req. (FT, IT, ST) ← According to TEs (6 month)

15 days (let)

UAT → Production Environment

* Done in UAT Environement
* Done in customer's place
1. Done by the customer
   ↓ Be half
   TEs [Domain Expert]
2. Done for customer Satisfaction.
3. Done as per a. business scenarioes
   (or) b. Real time scenarioes
   (or) c. how end user uses the application.

NOTE:

Domain Experts:

These are the members who is having the continuous experiance in a single domain.

Ex: Suppose 2 test engineers are there. One is having 3 yrs. of experiance in banking application & another one has same 3 yrs. of experiance in ERP (Enterprise Resource Planning), insurance. The new project will come on banking. So the person having the experiance in banking he'll be treated as domain expert.

Scanned with CamScanner

Cases of UAT:

Case-I:

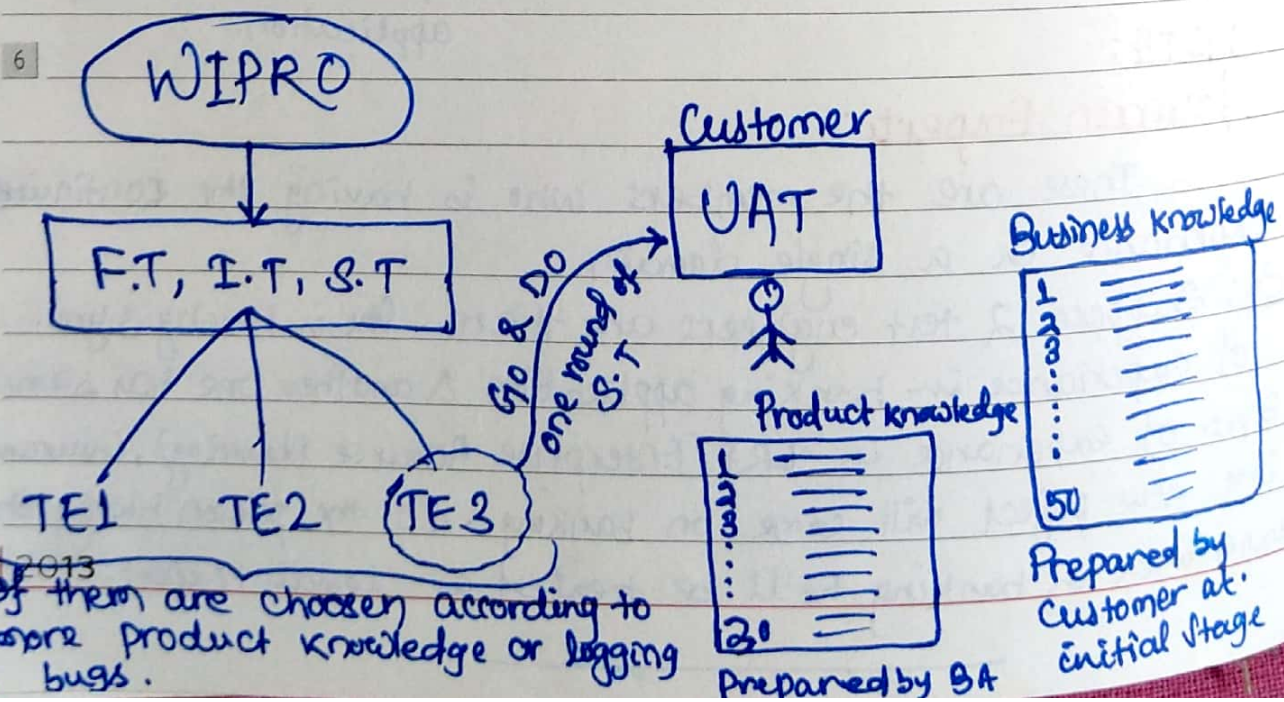
→ Here the TEs those are domain experts test the application on behalf of the customer.

Case-II:

→ Some customers can't afford the domain experts because they are highly paid.

→ So, they hire/pick one TE who is having a good product knowledge about the product.

→ The TE will move to the customer's place & start testing according to the business knowledge (documented format of requirements from the customer side) & product knowledge (documented format prepared by the TEs).



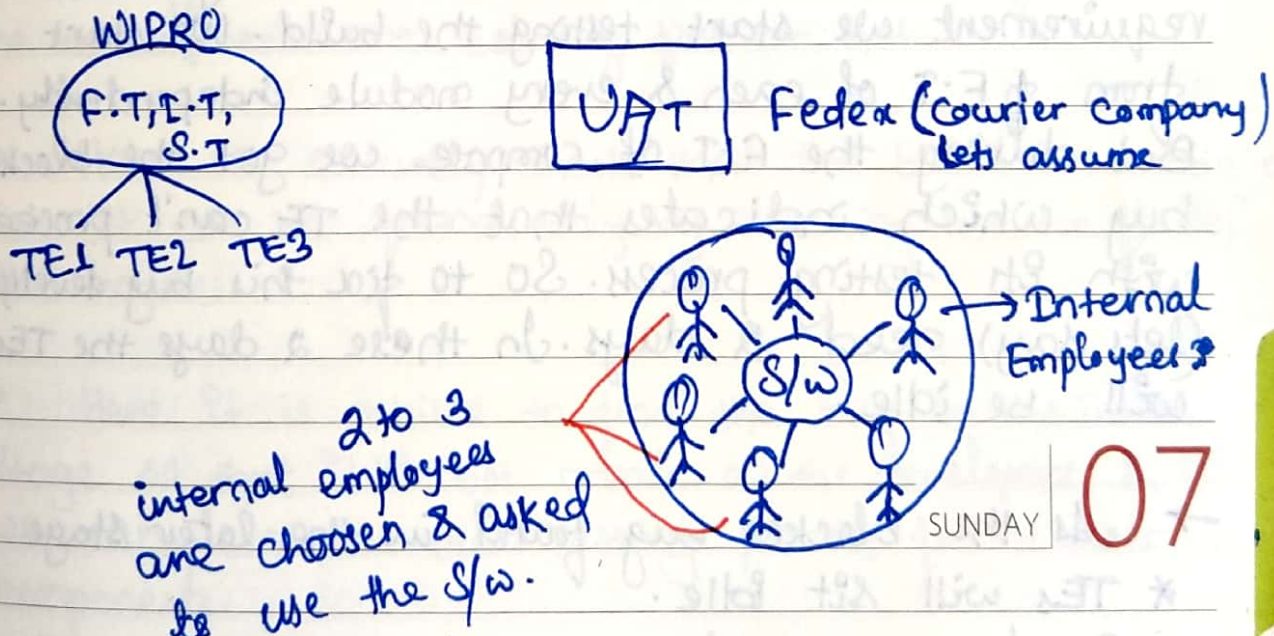
One of them are choosen according to the more product knowledge or logging more bugs.

Case-III:

→ In this case the UAT is done by the internal employees.

→ As the internal employees are very much familiar with the product so some persons are choosen according to the more knowledge about the product.

→ This is only possible/applicable only if the product is enhanced/S/w which is used internally in the organization & already the project is exist.



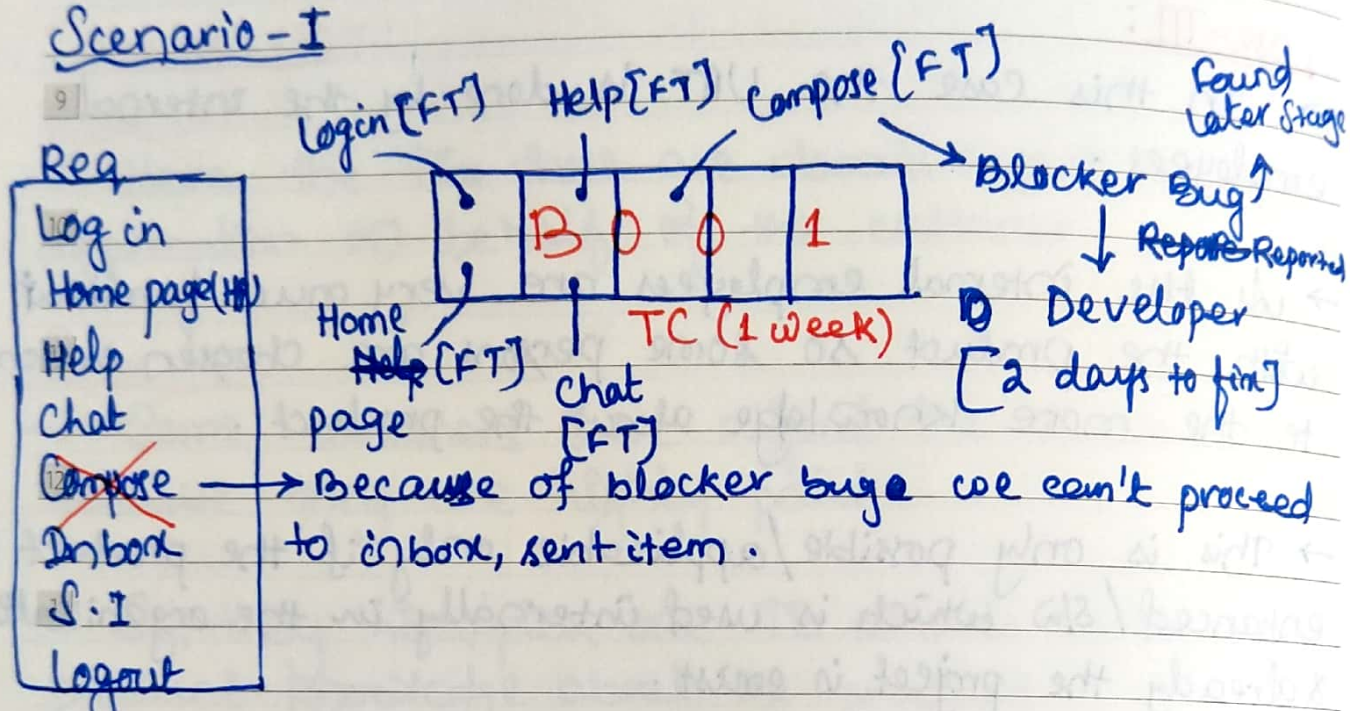
NOTE:

Don't speak about the cases of UAT until & unless asked.

Scanned with CamScanner

20.5.14

Scenario - I



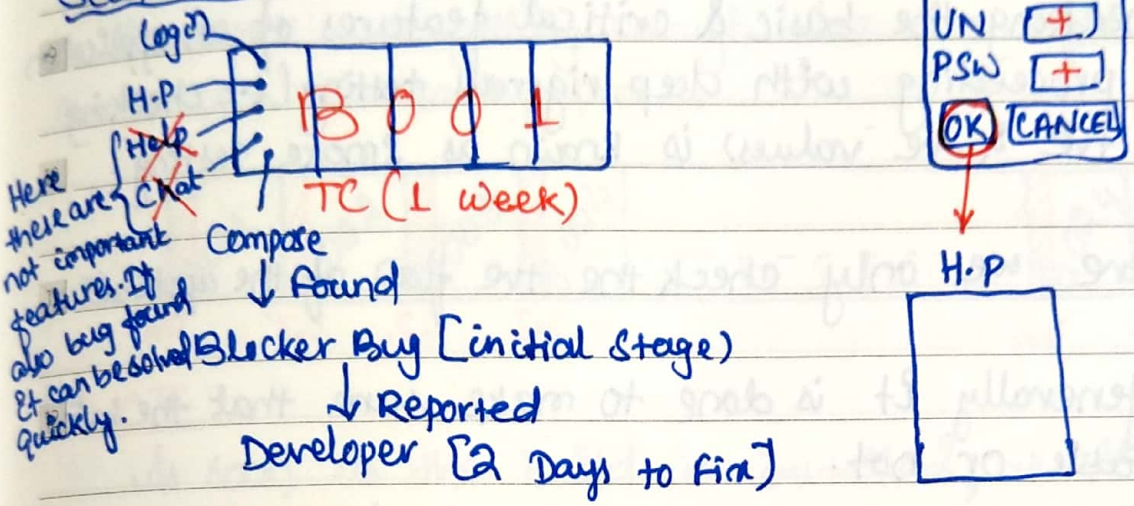
→ Here in this case as soon as we got the requirement we start testing the build. We start from F.T of each & every module independently. But during the F.T of compose we got the blocker bug which indicates that the TE can't proceed with its testing process. So to fix this bug developer (lets say) need 2 days. In these 2 days the TEs will be idle.

→ As the blocker bug found in the later stage

* TEs will sit idle.
* Developers are also under pressure because whenever the blocker bug reported to them they need some time to fix that at the same time they are busy with other features of the project. Because of more pressure they may create more bugs.
* It is a time consuming process.

Scanned with CamScanner

Scenario - II :



→ Here in this case we check the components & modules with single +ve value so that we can be confirmed that the components are working fine or not.

→ Here we actually check that the build is testable or not. It can be found only by checking the +ve flow of the modules.

→ Here it is possible to find the bug in the intial stage so that it'll not create on the developers & we can continue our testing process for the other components.

→ So the above type of testing is called as Smoke testing.

→ Here we should identify those components. for which the blocker bug may issued or for which the TE can't proceed. These kind of features are called as basic & critical features.

Scanned with CamScanner

Smoke Testing/Sanity Testing/Dry Run/Skim Testing/B.V.T:

→ Checking the basic & critical features of an application before proceeding with deep rigorous testing (i.e checking for all +ve & -ve values) is known as smoke testing.

→ Here we only check the +ve flow of the application.

→ Generally it is done to make sure that the build is testable or not.

Why Called Smoke Testing?:

→ In ancient stage people said that smoke is the previous stage of fire i.e whenever smoke will be there fire will start.

Ex: Suppose a hero & heroine lost their path. So to beg help they create smoke by blowing the sticks with an ignization which will create the fire & after that large amount of smoke will be in the air which can be identified when an aeroplane or helicaptor will pass in that way.

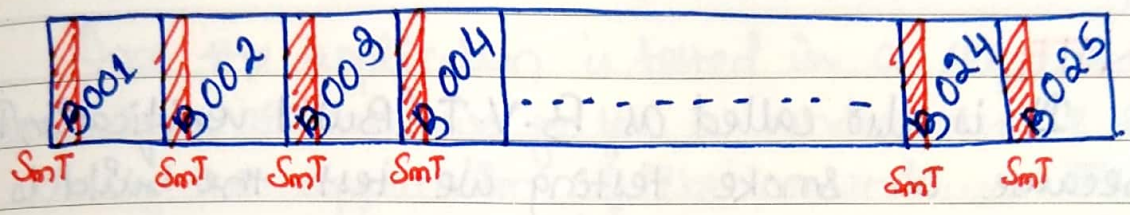
→ So before doing F.T, I.T, S.T we need a smoke test for the build.

Why Smoke testing?:

Generally it is done to catch the blocker bug at early stage so that.

Scanned with CamScanner

a. TEs are not sitting idle & can proceed with other features.
b. Developers are getting enough time to analyze & fix the bug.
c. It is a time saving process.



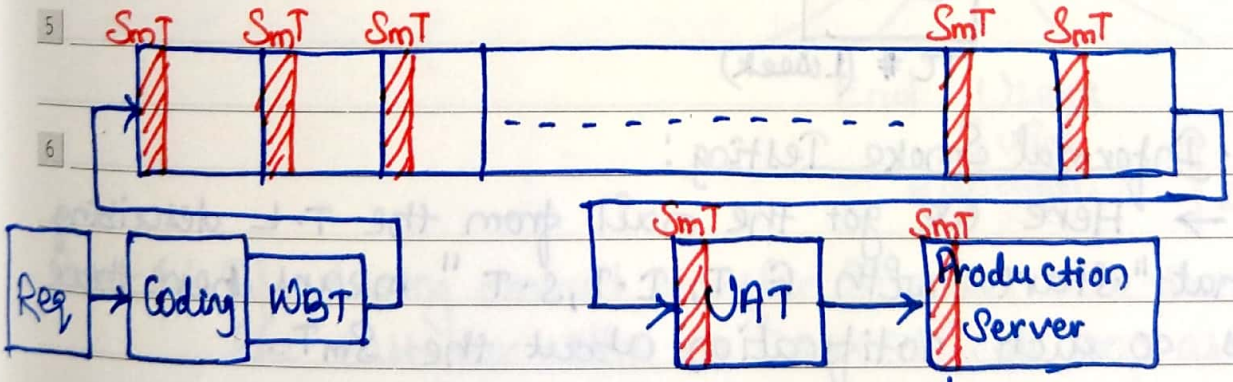
NOTE:

As soon as the build is installed, at 1st we've to do one round of smoke test for each & every build because ~~the b~~

(I) there may some change in the B.L of the new build
(II) Installation problem.

Summary:

We go for smoke testing as soon as ~~possible~~ the build is installed on the test server.



Here the smoke testing is done ~~by~~ in presence of BA, Development ~~Manag~~ Manager, Test Manager & the customer ~~by~~ using the customer login profile.

Summary:

Whenever the build is installed in a new place, we should go for one round of smoke testing.

NOTE:

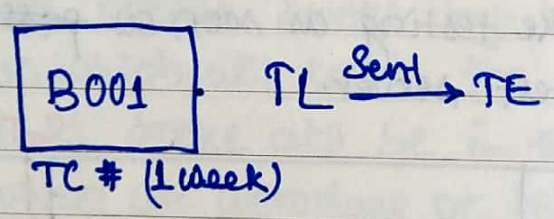
It is also called as B.V.T (Build verification Testing) because in smoke testing we test the build is either testable or not.

Types of Smoke Testing:

Generally, there are two types of smoke testing.

1. Formal Smoke testing:

Here we got a mail from the T.L describing that "Start with Smoke test, F.T, I.T, S.T" with the build as well as T.C.



2. Informal Smoke Testing:

→ Here we got the mail from the T.L describing that "Start with F.T, I.T, S.T" means here there is no such notification about the SmT.

→ Here the T.L won't inform us to start with smoke testing.

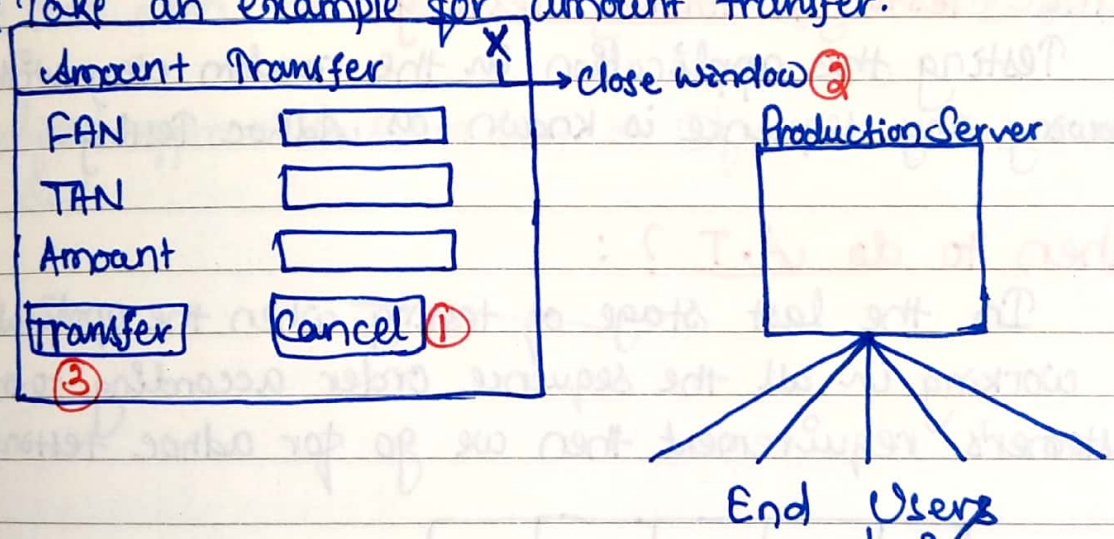
Scanned with CamScanner

Is Smoke Testing Mandatory? :

NO, but it is a very good approach to find the blocker bug in the initial stage itself.

Once the application is tested in a sequential manner then at last we go for a random test i.e here we check the application as like an end user. This kind of test is generally called as "adhoc testing" or "monkey testing".

Ex: Take an example for amount transfer.



Case-1: If during amount transfer it takes more time then the customer click on "CANCEL". As soon as he cancels he gets the confirmation message that "Amount Transfer Successfully". To cross check it he asks his friend/receiver. So in this kind of situation it'll be better if we disable the click button.

Scanned with CamScanner

Case-2: Some persons feel bore & they close the window. As soon as they close the window they got a confirmation msg. So the sol.^n is we should prompt a msg while closing the window.

Case-3: Some persons bored, so repeatedly they click on "TRANSFER". So that they got the confirmation msg many times. So the sol.^n here is just disable the transfer button after 1st click.

Adhoc Testing / Monkey Testing:

Testing the application in the random way without following any sequence is known as Adhoc Testing.

When to do A.T ? :

In the last stage of testing when the application is working in all the sequence order according to the Customer's requirement then we go for adhoc testing.

S.T	F.T	I.T	S.T	A.T

Why A.T required?

Since the user doesn't know how to use the application because he never follow the user manual & may use it in a adhoc way / random ways & find some issues to cover these problems we should do one round of adhoc testing.

Scanned with CamScanner

NOTE:

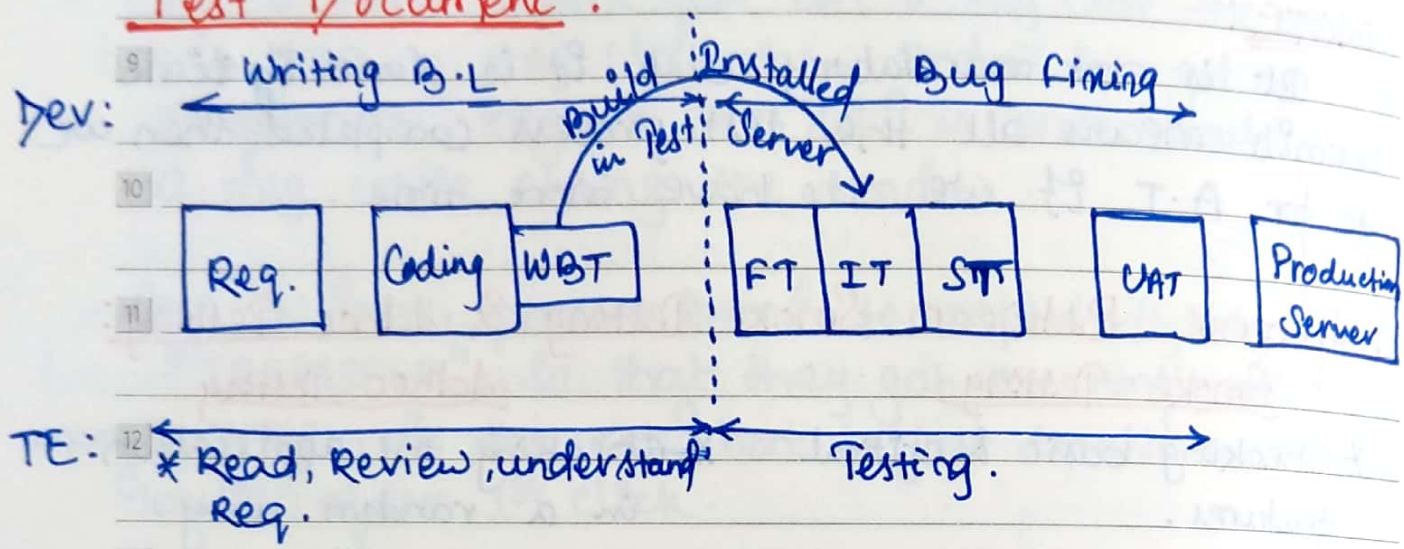
It is not mandatory but it is done if time permits means all the test process completed then we go for A.T if we do have more time.

Difference Between Smoke Testing & Adhoc Testing:

Smoke Testing	Adhoc Testing
* Checking basic & critical features.	* Checking the application in a random way.
* As soon as build is installed	* Last stage of testing.
* Done with +ve flow	* Most of A.T are of -ve scenarioes.
* Good approach to start with S.T.	* If time permits only.

21.5.14

Test Document :



If we won't prepare T.D,
* We may forget the values.
* Time consuming for New TE when joins (if the TE left the company)
* Test Consistency is not there.

So we prepare the T.D, so that
* We won't forget the values.
* Time saving & ~~we~~ New TE can start ~~with~~ testing according to the T.D.
* Test consistency is maintained.

Definition:

→ These are the documents prepared by TEs before starting the test execution process.

→ The T.Ds are prepared simultaneously when the developers are busy in developing the B.L of the application.

Scanned with CamScanner

→ T.D should be completed with in the time the build is ready for testing.

Process:

→ As soon as the SRS is prepared, the requirement is being kept in a centralized stored place which is known as "Requirement Repository".

→ Then the T.L assigns some modules/features of the requirement to some TEs.

→ The TEs may not get the flow of requirement as well as the terminologies used in the requirement document while they go through the document during a assigned certain time period.

→ Then a meeting is conducted by BA which is called as "Requirement Acceptance Meeting / RA Meeting" where developers, TEs & leads are attending & clarifying regarding the flows & terminologies.

→ Then all the developers & TEs ~~walk~~ walk through the corresponding modules of requirements.

→ Then the developers are busy with writing the B.L & TEs prepare the T.D.

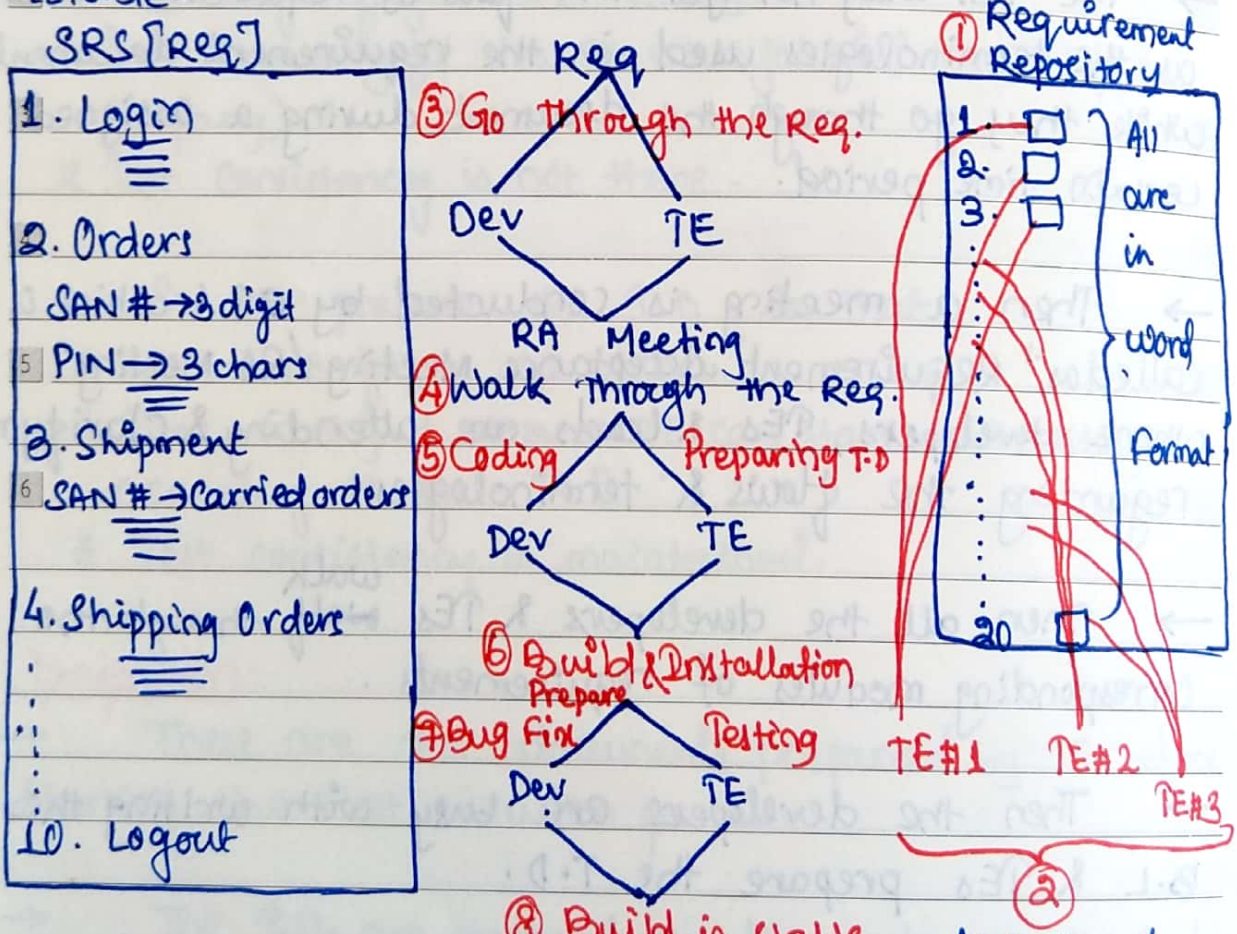
Scanned with CamScanner

→ Once the B.L is over it's prepared as build & installed on T.S.

→ TEs start testing their assigned modules. (or) TEs busy in test execution process.

→ During this period, if TE found any bug it is reported to the developers & developers busy with bug fixing.

→ This process continues untill & unless the build is stable.



22.5.14

Test Scenarios:

→ It is a high level document prepared by the TEs that consists of all the possible combinations or ways of testing the application.

→ Generally, it is written to understand the flow of the application.

→ We never concentrates for F.T, I.T & S.T while preparing the test scenario document.

→ It is done while the developers are busy in writing the S.L & before the build is prepared.

Req	Test Scenarios (DO / Check)
1. Login	LOGIN: 1. Enter valid UN & PSW, Check H.P is displayed
2. Compose	2. Enter invalid UN & PSW, Check E.M is displayed
3. Inbox	3. Leave UN & PSW blanck, Check EM is displayed
4. S.I	4. Enter UN & PSW, Cancel it, Check fields are reset
⋮	5. Enter UN & PSW, Check UN on H.P
	6. Compose mail, Send it & check for con msg.
	7. Compose mail, Send it & check in inbox of Rx & S.I of sender.
	8. Compose mail, SAD, & check for con msg. (DO / Check.)
10. Logout	⋮

NOTE:

Points to be remembered while writing Test Scenarios:

→ Always start writing the test scenarios module by module so that more no. of test scenarios can be covered & we'll not miss out any scenario.

→ Every scenario should be in one sentence or maximum of two so that its easy to understand.

→ Every sentence should have do's & check's.

NOTE:

Normally production based companies prepare this but service based companies directly prepares the test cases.

Test Cases:

→ It is a in detailed document which defines the step by step procedure to test an application.

→ It consists of i/ps, navigation steps & all possible scenarios to test an application.

→ During preparation of test cases we should have
(I) Requirement
(II) Test case Template which is the format for preparing the test cases. It is provided by the T.L to maintain the uniformity & normally use the excel sheet.

Scanned with CamScanner

Test Case Template:

HEADER

Test Case Name/Id :
Test Case Type : F.T.Case/I.T.C/S.T.C
Requirement # :
Module Name :
Status :
Severity :
Pre-Condition :
Test Data :
Release :
Version :
Brief Description :

BODY

Step #	Description	I/Ps	Expected Result (X)	Actual Result (Y)	Status	Comments
			X = Y		Pass	
			X ≠ Y		Fail.	

FOOTER

Author : Who prepares the test case.
Date :
Reviewed By :
Approved By :

NOTE:
The above format is a generic format.

NOTE:
It is also prepared when the developers are busy with their B.L.

Severity:

→ It indicates the importance of the test case.

Ex: Suppose we've 200 test cases then we've to test all the 200 test cases. But if we've don't have more time / enough time & we are having 150 test cases then we've to check as per the severity.

→ It may be "Critical" or "major" or "minor".

Release:

It is the final stable product handed over to the customer.

Release Vs Version:

Release: It is the major modification done to the existing application.

Version: It is the minor modification done to the existing release.

Ex:

Cheeta	1.0
Cheeta	2.0
Cheeta	3.0
Tiger	1.0
Tiger	2.0
Lion	1.0
Lion	2.0
Lion	3.0
Lion	4.0

Major, Major, Minor, Minor

Release, Version.

Scanned with CamScanner

NOTE:

① Pre-Condition:

It is the minimum data setup or configuration that needs to be created before executing the testcases.

② Test Data:

It is the data pre defined to do the test execution.

③ Generally the pre-condition changes from test case to test case.

Functional Test Case Template of Login Module:

1. Login

UN	[ ]	→ 3 digit
PSW	[ ]	→ 3 alphabets
OK	} Enabled	
Cancel		

Format of Test case name

Release_version_project_module.

Test Case Name : Cheeta_2.0_ICICI_Login
Test case Type : Functional Test Case
Requirement # : 1
Module : Login
Status : Filled up after test execution.
Severity : Critical
Release : Cheeta
Version : 2.0
Pre-Cond.ⁿ : 1 valid Login < UN, PSW
Test Data : UN:-123 & PSW :- abc
Brief Description : This case is to check functionality of Login.

Scanned with CamScanner

Filled up after test execution.

Step #	Description	I/Ps	Expected Result (ER)	Actual Result	Status	Comments
1.	Open "BROWSER" & Enter "URL"	http://---- --------	"LOGIN" page MBD			
2.	Enter Following value of "UN"					
	(I) Valid data	(I) 123	(I) Accept.			
	(II) Invalid data	(II) 456	(II) E:M "Invalid UN/PSW" MBD			
	(III)	(III) N/A	(III) E.M "UN Can't be blank" MBD			
3.	Enter following values of "PSW"					
	(I) Valid Data	(I) abc	(I) Accept			
4.	Enter valid "UN" & "PSW", click on "OK" Button	123 abc	"Home" page MBD			
5.	Enter "UN" & "PSW" & Click on "CANCEL"	123 abc	All the fields are reset			

Author :
Date :
Reviewed By :
Approved By :

Scanned with CamScanner

NOTE:

The most challenging thing while writing the test case is to have maximum test coverage i.e to derive all the possible values both +ve as well as -ve values.

Scanned with CamScanner

Test Case Design Technique:

→ These are the methods or techniques or procedures which are used while ~~teste~~ writing the test cases to achieve the max. test coverage.

→ Different Test case design techniques are,
(I) Error Guessing (EG)
(II) Equivalence Partitioning (EP)
(III) Boundary value Analysis (BVA)

(I) Error Guessing:

In this method, the T.E starts guessing the values based on his understanding of the requirement.

Ex: Amt Transfer:
Amt : 5K - 50K
Values:
(I) 6K → Accept
(II) 51K → E.M
(III) 2K → E.M
(IV) ☐ → E.M
(V) Decimal → E.M
(VI) Alphanumeric → EM
(VII) Copy & Paste → EM

Drawbacks:

→ It depends on person to person completely
→ Min. Test coverage may not gurantee i.e one +ve & one -ve value.

Scanned with CamScanner

→ Boundary values are not covered.

(II) Equivalence Partitioning:

(a) **Range:** If the req. is range of values then derive the test case for one +ve & two -ve values.

Ex:

EP		EG
(I) 6K → Accept (II) 4K → EM (III) 61K → EM	+	(I) Decimal → E.M (II) [ ] → E.M (III) Copy & paste → E.M

(b) **Set:** If the req. is a set of values then derive the test case for one valid & two invalid i/ps.

Ex: Product ID [1, 2, 3, 4, 5]

EP		EG
*1 → Accept *6 → E.M *0 → E.M	+	*2 → Accept *3 → Accept *4 → Accept *5 → Accept *0L → E.M *1.0 → E.M *Glass → E.M *1,2 → E.M

1 → T-Shirt
2 → Shoes
3 → Bag
4 → Glass
5 → Watch
↓
Test for each & every value.

(c) **Boolean:** If the req. is a boolean i.e True/False then derive the test case for both true as well as false cond.?

Ex:

Scanned with CamScanner

Ex: (i) Obama is CM of Karnataka
- ○ True
- ○ False

(ii) Osami is maa of
- ○ Dinga
- ○ Dingi
- ○ Champa
- ○ Chameli

EP		EG
* valid → +1 * Invalid → -1	+	* Select all * Leave all

(III) Boundary Value Analysis:

If the req. is a range of values i.e [a, b], derive the test case as follows.

ⓐ a, a+1, a-1

ⓑ b, b+1, b-1

Ex:

BVA		EG
* 50K → Accept * 5001 → Accept * 4999 → EM	* 50K → Accept * 50001 → E.M * 49999 → Accept	+

↓

4 +ve & 2 -ve values.

NOTE:

(I) For max. test coverage following any technique with error guessing.

(II) For range of values, go for BVA with EG for max. test coverage

(III) For set of values & boolean, go for EP with EG for max. test coverage.

Scanned with CamScanner

30.05.14

Ex: ① UN → 3-6 alphabets → Range

According to B.V.A.

* abc → Accept
* abcd → Accept
* ab → E.M
* abcdef → Accept
* abcdefg → E.M
* abcde → Accept

② PSW → 3 digit,
Straight Forward Req. → Prefer E.G.

NOTE:

By default always prefer E.G or if the req. is straight forward prefer E.G.

Ex: ① Amt Transfer → [1 Lakh - 7 Lakh] → Range

1 Lakh → Accept
8 Lakh → E.M
50000 → E.M

→ Pressman Method

② Loan [1 lakh - 7 lakh] → Range

1 Lakh → Accept
8 lakh → E.M
50K → E.M

Criteria
(I) 1-2 Lakh → 5.68%
(II) >2-5 lakh → 3.29%
(III) >5-7 lakh → 1.99%

Here in the above case one +ve & two -ve values are tested which can't cover all the min. test coverage.
So for min. test coverage,

[1L, 2L, 5L, 7L]

BVA: +1 -1 +1 -1 +1 -1 +1 -1

-ve value

2013 EG : Decimal → E.M
[ ] → E.M

General Practice Method

Scanned with CamScanner

→ Whenever the req. is a range of values then we simply prefer the pressman method with one +ve & two -ve values. Whenever the req. is range of values with the criteria is being attached at that moment pressman method never covers the min. test coverage. So we go for the general practice method.

→ Here we normally choose the min. values from the criteria along with the max. value. Or simply we can say it the boundary values of each criteria value.

→ After choosing the boundary values we apply B.V.A so that the min. test coverage can be achieved but still we can't achieve max. test coverage.

→ To achieve max. test coverage we go for E.G along with B.V.A.

How to write Functional Test cases:

Req. ⇒ 10. Amt. Transfer

FAN	[ ]	→ 10 digit
TAN	[ ]	→ 10 Digit
Amt	[ ]	→ 5-50K
Transfer, Cancel	} Enabled	

Scanned with CamScanner

Test Case Name : Cheeta_2.0_ICICI_AT_FT
Test Case Type : F.T.Case
Req. # : 10
Module : Amt Transfer
Status :
Severity : Critical
Release : Cheeta
Version : 2.0
Pre-cond.ⁿ : 1 valid Login, 2 valid Acc.No., min. bal must exist
Test - Data : UN→Dinga, FAN→1234567890 [5-50K]
PSW→Dingi, TAN→1234567891
Brief Description : Check Functionality of A.T

Step #	Description	i/ps	E.R	A.R	Status	Comment
1.	Open "BROWSER", Enter "URL"	http://.....	"LOGIN" page MSD	As expected	P	X
2.	Enter valid "UN" & "PSW" & click "OK" button	Dinga Dingi	"HOME" page MSD	As expected	P	X
3.	Click "Amt-Transfer"	N/A	"A.T" page MSD	As expected	P	X
4.	Enter values of "FAN"					
	(I) valid FAN	(I) 1234567890	(I) Accept			
	(II) Invalid A/c	(II) 1234567899	(II) E.M "FAN A/C # Doesn't Exist" MSD			
	(III) Blank	(III) N/A	(III) E.M "FAN A/C # can't be blank" MSD			

Scanned with CamScanner

Step					
5.	Enter value of "TAN"				
	(i) valid TAN	(i) 1234567891	(i) Accept		
	(ii) Invalid TAN	(ii) 1034789980	(ii) E.M "TAN A/C # Doesn't Exist" MBD		
	(iii) Blank	(iii) N/A	(iii) E.M "TAN A/C # Can't be blank" MBD		
6.	Enter value "Amt"				
	(i) Valid Amt	(a) 5000 (b) 5001 (c) 50000 (d) 49999	(a) (b) (c) (d) } Accept	As expected	Pass Fail → Log a bug.
	(ii) Invalid	(a) 4999 (b) 500001	(a) (b) } EM "Amt b/w 5-50K" MBD		Not accepted
	(iii) Blank	N/A	(iii) "Amt can't be blank" MBD		
	(iv) Amt > Bal	As per Testing	(iv) "Insufficient Bal" MBD		
7	Enter valid "FAN", "TAN", "Amt" click on "TRANSFER"	1234567890 1234567891 5000	Conf. msg. "Amt Transfer Successfully".		

Scanned with CamScanner

8. Enter "PAN", "TAN", "AMT", click on "CANCEL"	1234567890 1234567891 5000	All fields must be reset			

Author :
Date :
Reviewed by :
Approved by :

Test Execution:

→ The process of testing the application by executing the the test cases is known as test execution.

→ It starts only after the ~~the~~ test document is prepared, reviewed, approved & the build installed on T.S.

Status of Test Cases:

(I) Pass: When all the steps of the test cases are working as expected the status is pass.

(II) Fail: If some of the ~~steps~~ steps of the test cases are not working as expected then the status is fail.

NOTE:

① Partial: When all the major steps of the test cases are working as expected where as any of minor steps are not working as expected then status may be partial.

Scanned with CamScanner

(II) After all the Status in body part is pass then, we'll update the status as pass in header.

(III) When there's a bug, it's being logged & reported to developer & developer fix it & send it to T.E for retest the app.

Scanned with CamScanner

Integration Test case for Amount Transfer:

Test Case Name : Cheeta_2.0_ICICI_AT_Int
Test case Type : Integration Test Case
Req. # : 10
Module : Amount Transfer
Status :
Severity : Critical
Release : Cheeta
version : 2.0
Pre-Cond^n : 2 valid logons, 2 valid A/C #, Min. Bal Exist
Test Data : User 1 — Dinga, Dingi ; User-2 — Manga, Mangi ; AC # — 1234567890, 1234567891 ; (5K-50K)

Brief Description : This case is to check data flow of A.T

Step#	Description	I/ps	E.R	A.R	St	C
1.	Open "BROWSER" & Enter "URL"	http://:...	"LOGIN" page MBD			
2.	Enter "FAN" User valid "UN", "PSW" & click on "OK" button	Dinga Dingi	"HOME" page MBD			
3.	Click on "AT"	N/A	"A-T" page MBD			
4.	Enter valid "FAN", "TAN", "AMT" & Click on "TRANSFER"	1234567890 1234567891 5000	Conf. Msg.: "Amount Transfer Successfully" MBD			
5.	Click on "A/C Bal"	N/A	Current Bal = Prev. Bal − Trans. Amt			
6.	Click on "LOGOUT"	N/A	"LOGIN" page MBD			

Scanned with CamScanner

7.	Enter valid "TAN" user "UN", "PSW", click on "OK"	Manga Mangi	"Home" page MBD			
8.	click on "A/c Bal"	N/A	Current Bal = Prev. Bal + RCV Bal			
9.	click on "LOGOUT"	N/A	"LOGIN" page MBD			

Author :
Date :
Reviewed by :
Approved by :

System Test Case for OD Feature :

Test Case Name : Bhalu_3.0_ICICI_OD_ST
Test case Type : System test case
Req # : 11
Module : OD
Status :
Severity : Major
Release : Bhalu
version : 3.0
Pre cond.ⁿ : 2 valid Login, 1 Valid EmpId, Sal, Access DB
Test data : User :- Dinga, Dingi
Manager :- Manga, Mangi
1 valid EmpId :- 1122
Sal :- 10K
Brief Description : This case is to check end-to-end flow of OD Feature.

Scanned with CamScanner

Step#	Description	I/Ps	ER
1.	open "BROWSER" & Enter "URL"	http://.....	"LOGIN" page MBD
2.	Enter valid "UN", "PSW" for user & click on "OK"	Dinga Dingi	"Home" page MBD
3.	Click on "OD"	N/A	"OD" page MBD
4.	click on "APPLY FOR OD"	N/A	"APPLY OD" page MBD
5.	Enter valid "EMPID", "SAL", "OD Amt", click on "APPLY".	1122 10 K 20K	(i) Conf. Msg: "APPLY OD SUCCESSFULLY" (ii) OD AMT accept only twice of sal.
6.	Click on "LOGOUT"	N/A	"LOGIN" page MBD
7.	Enter valid "UN", "PSW" for Mangger, click on "OK"	Manga Mangi	"HOME" page MBD
8.	Click on "OD"	N/A	"OD" page MBD
9.	Click on "APPROVE OD"	N/A	"CHECKBOX FOR VARIOUS User" MBD
10.	Select check box for Dinga & click on "APPROVE"	N/A	(i) Conf. Msg: "OD AMT APPROVED SUCCESSFULLY" (ii) OD Amt should be credited to user only after 30 days of approval
11.	Click on "LOGOUT"	N/A	"LOGIN" page MBD
12.	Access DB & Do necessary change	update OD set CD = "App. Date + 30" where empid = 1122;	(i) CD should be changed/updated successfully. (ii) OD Amt credited to Dinga.
13.	Enter valid User "UN", "PSW" & click on "OK"	Dinga Dingi	"Home" page MBD

2013

Scanned with CamScanner

14.	Click on "OD"	N/A	"OD" page MBD
15.	Click on "A/C OD BAL"	N/A	Cnt Amt = Prev Bal + Received OD Bal
16.	Click on "LOGOUT"	N/A	"LOGIN" page MBD
17.	Enter valid user "UN", "PSW", click on "OK"	Dinga Dingi	"HOME" page MBD
18.	Click on "OD"	N/A	"OD" page MBD
19.	Click on "REPAY OD"	N/A	(i) Repay OD 1st time: OD Amt 20000, Int 500, P.F 500, 21,000 — (ii) Repeat 1-18: OD Amt 20000, Int 500, 20 500, PF N/A for 2nd time
20.	Click on "REPAY"	N/A	Conf. Msg: "OD Amt Repaid Successfully"
21.	Click on "A/C Bal"	N/A	Current Bal = Prev Bal - Repaid OD Amt
22.	Click on "LOGOUT"	N/A	"LOGIN" page MBD

Author :
Date :
Reviewed By :
Approved By :

NOTE:

1. Pre cond.? is a min. set up which
   (a) Can't create
   (b) Not using in testing.
2. Requirement should be covered in E.R so that new T.E can get the requirement.

Scanned with CamScanner

Case-1:

Whenever the application is unknown like ERP, SAP, SCM, Insurance, Banking to us ~~in the interview~~ & the interviewer asked to write the test cases then we should ~~counter~~ ask him for the requirement.

Case-2:

When the interviewer asked us to write the test case for the known application like Gmail, Yahoo, Google, Facebook but he has not given with module, then write the test scenarios.

Case-3:

Whenever a module is given & asked to write the test cases, at that time we should write the test scenario in the test case template without i/ps & the test case type because SRS document is not with us.

Ex: For Compose

Header without Status

Step#	Description	i/p	ER	AR	S
1.	~~Click Enter valid~~ Compose mail, send it chk for conf. msg.	✓	✓	✓	✓
2.	Compose mail, send it, Check inbox of Rx & Sent item of sender.	✓	✓	✓	✓
⋮	⋮				

Scanned with CamScanner

Author :
Date :
Reviewed By :
Approved By :

Case-4:

If it is asked to write the test case template then, write the whole test case template.

Case-5:

If it is asked about the test cases on actitime or from our corresponding project then always get prepare with 20 to 25 test scenarios before going to interview & if it is asked for a particular module then get prepare with 5 test cases on end-to-end flow minimum.

Case-6:

If it is asked to write test cases on objects like

* pen
* Paper weight
* A/C
* ATM
* Water Bottle
* Mobile
* Chair
* Etc....

then write the test scenarios step by step starting from F.T, I.T, S.T, C.T, P.T, U.T, A.T.

Ex: For Pen,

* Check all the components are there or not.
* Fit all the components, check they are well fitted or not.
* Open cap & start writing
* write in different different places like wall, table, etc.

Scanned with CamScanner

* Throw the pen to the floor check it is broken or not.
etc.....
* check for colour & feel is good or not.
- - - - - - .
- - - - - - - .

Test Case Review:

Date = Date	: Read, Review, understand Req → TE
Date - Date	: write test cases → TE
Date - Date	: Review test cases → Other TE.
Date	: Approve test case → TL
Date	: Test Execution → TE

↓ Time slots.

NOTE:

Once after the test case writing is completed by all T.Es then only we go for test case review. We can't go for the review when one T.E is busy in writing the test cases of his corresponding module.

Here every activity is scheduled or having a time slot.

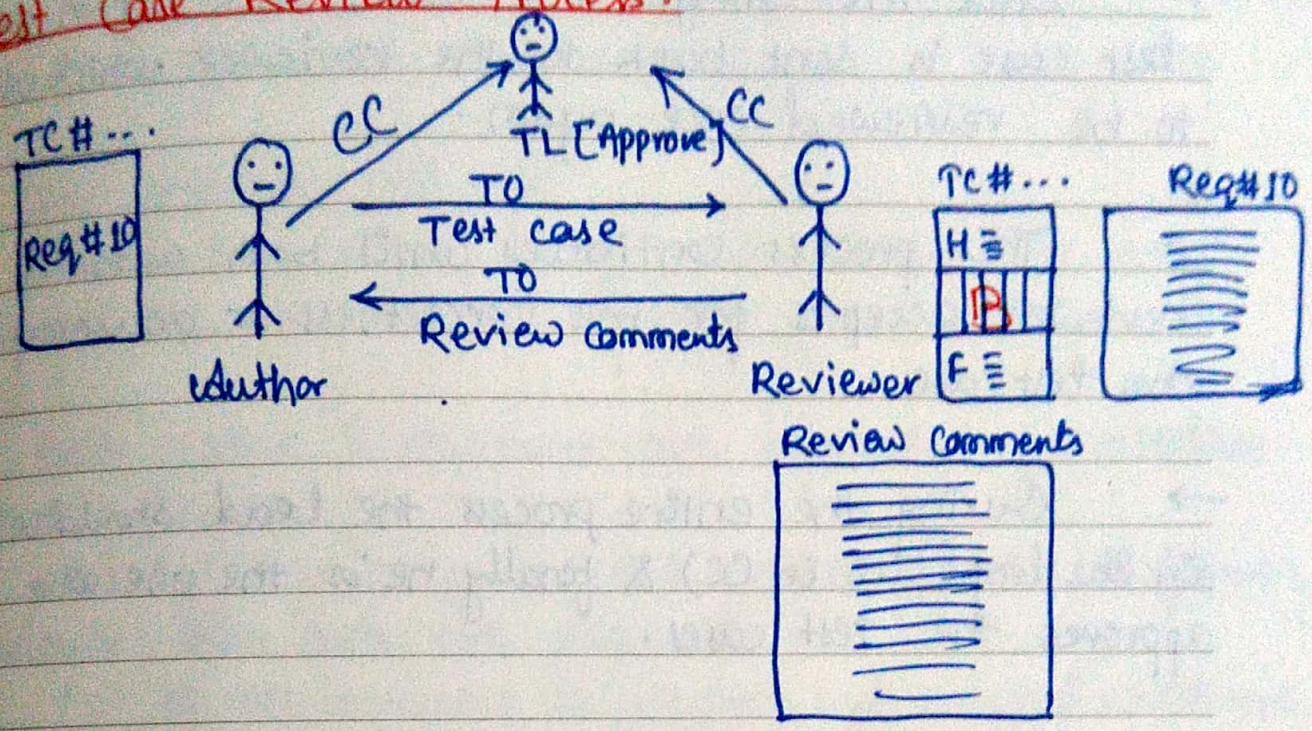
NOTE:

Q. Why test case review is necessary?

Ans: Once the test case is written, it should be reviewed by the other test engineer to make sure that the test case is accurate i.e it has max. test coverage & it is written in a proper flow so that any one can reuse the test case.

Scanned with CamScanner

Test Case Review Process:



→ Once the test case is written, it should be sent for the review process to check the accuracy (i.e max. test coverage & proper flow) of the test case.

→ The reviewer opens the test case & corresponding requirement & start reviewing.

→ During review process, he may encounter some mistakes which needs to be documented in a separate sheet.

→ The reviewer sends back ~~to~~ the review comments to the author.

→ The author goes through the review comments & do the necessary changes where ever required.

→ Once the changes are made, once again the test case is sent back to the reviewer which needs to be reviewed once again.

→ This process continues until both author & reviewer accepts for the correctness or accuracy of the test case.

→ During the entire process the lead should be kept in the loop (i.e in CC) & finally he is the one who approves the test cases.

→ This written, reviewed & approved test cases are stored in a centralized location known as "test case repository"

Test Case Review Template:

Test Case Name	Step#	Severity	Review Comment	Author Comment.
Cheeta_2.0_ICICI_AT	N/A	Critical	Pre cond.ⁿ missing	Not Fixed since no pre cond.ⁿ with reason
	5	Critical	+ve values missing for FAN	Fixed
	9	Major	Need more +ve values	Fixed
	14	Minor	Need more -ve values	Fixed
	22	Minor	Sentence not clear	Fixed
	=	=	=	=
	=	=	=	=
	=	=	=	=

Scanned with CamScanner

Review Ethics [Followed by Reviewer]:

1. Look for max. test coverage.

2. Find mistakes not sol.?
   Here the author may not be serious & he may depend on reviewer if reviewer gives sol.?

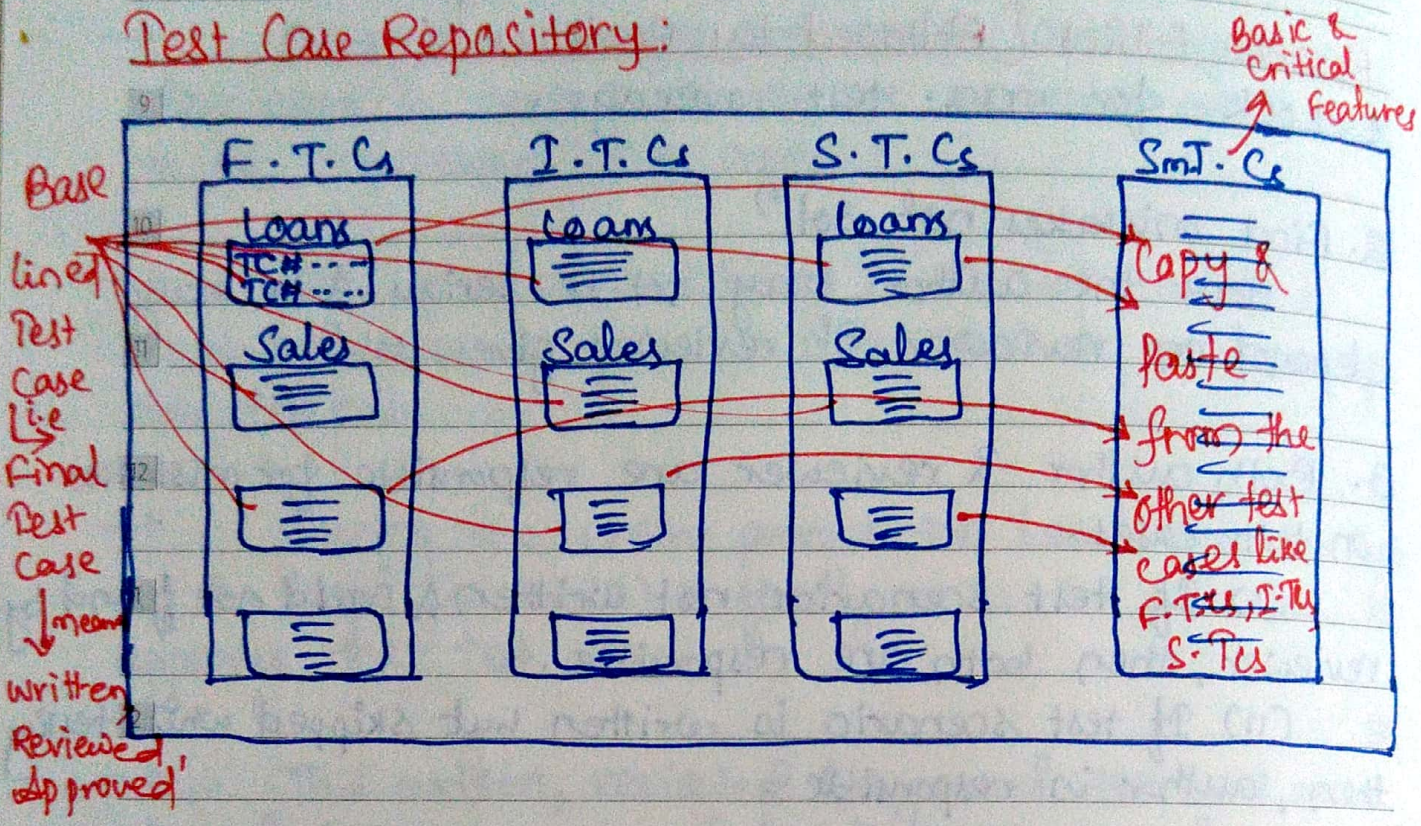
3. Both author & reviewer are responsible for mistakes on test cases.
   (I) If test scenarios not written & could not found by reviewer, then both are responsible.
   (II) If test scenario is written but skipped while testing then, author is responsible.

4. Look content not author.
   Here we should not create useless things against the author for the personal issues.

Ex: Two best friends named Sita & Gita worked in a same company. During appraisal Sita got less hike as compared to Gita & they are working on different projects. So Sita got angried. Again they both put Sita & Gita in the same project & Sita is the reviewer of Gita.

Scanned with CamScanner

Test Case Repository:



→ Normally Test Case Repository creates by the TL.

→ It is a centralized location where all the base lined test cases are stored.

→ The test cases can be stored in the repository only after they are written, reviewed & approved.

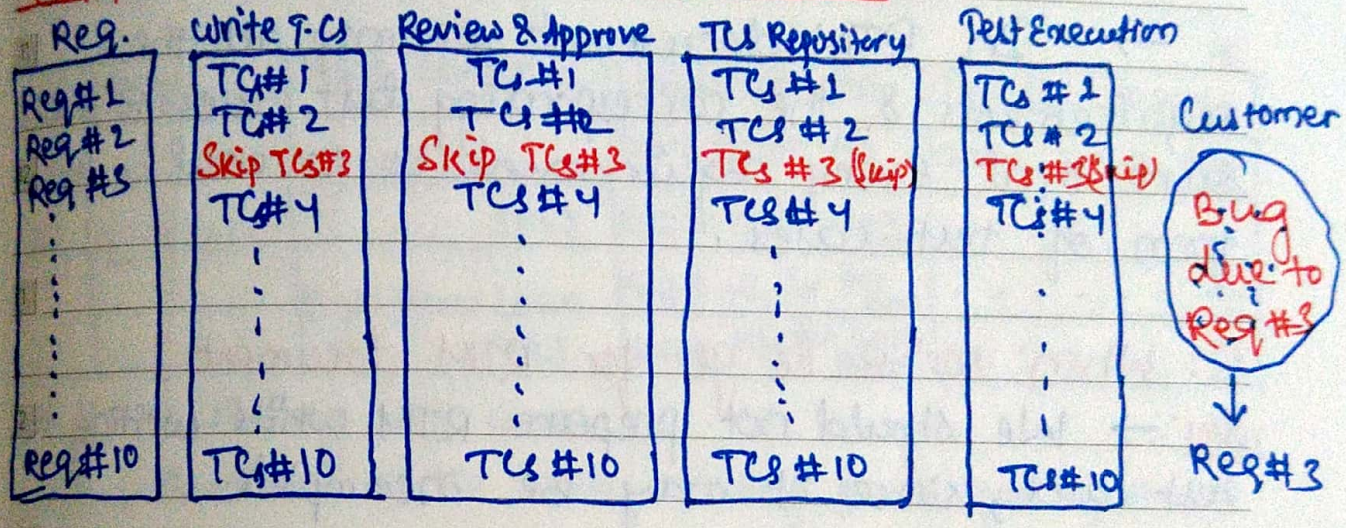
→ The base lined test cases can't be made any changes after storing in test case repository.

NOTE:

We don't write a separate set of smoke test cases where as we pick them from the existing

test case repository based on the basic & critical features of the application.

RTM (Requirement Traceability Matrix) or CRM (Cross Reference Matrix) :



Sometimes due to some problem like if the TE can't understand the requirement, TE skips to write the test case on that particular requirement, which can't be caught during review & approve process, stored at repository & also during the test execution because the TE might forgot about that requirement. But whenever it (product) is handed over to the customer, he encountered with bugs in requirement. So to avoid such kind of problems we have a matrix called requirement traceability matrix which traces the skipped requirements

RTM

Req.	TCs Name
R1	TCs#1
R2	TCs#2
R3	X
⋮	⋮
R10	TCs#10

Missing Test case.

Q. What is ~~RM~~ RTM document & why?

Ans: → It is a document which is prepared before test execution to make sure that we are not missing any requirements for testing.

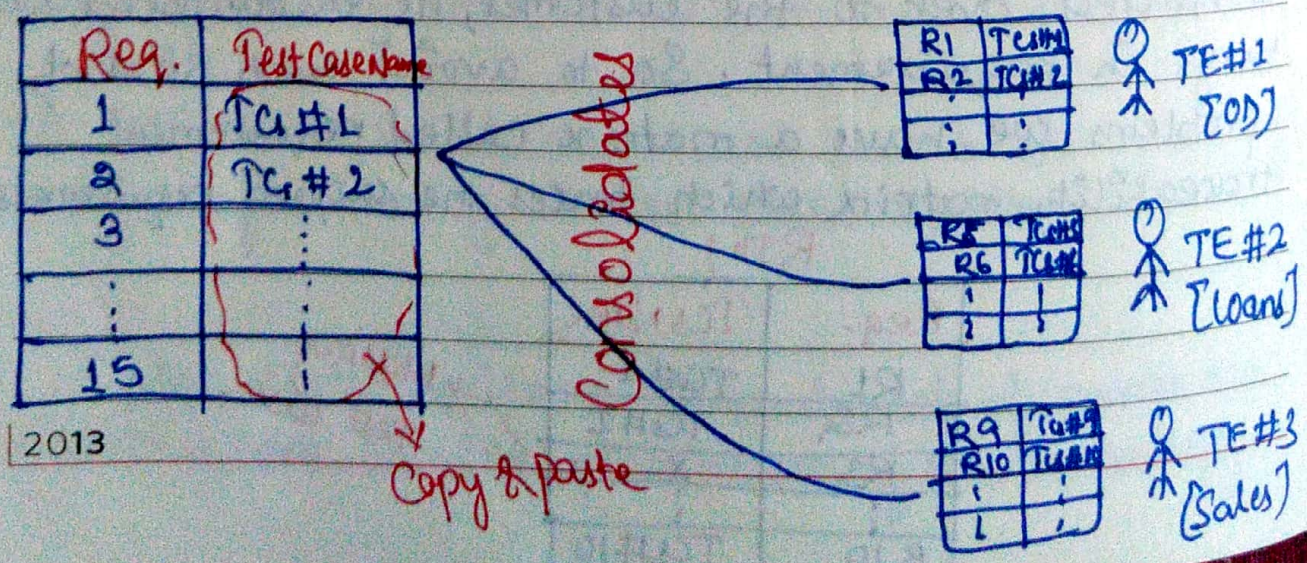
→ In RTM document we map all the requirements & the corresponding test cases to make sure that every requirement is covered in the form of test cases.

Q. When we've to go for RTM document?

Ans: → We should not prepare RTM while writing the test cases, since it may be incomplete.

→ Also not at the time of review & approval since it may get rejected.

→ RTM should always be prepared once the test cases are stored in the test case repository & before we start the test execution process.



Scanned with CamScanner

Q. Who prepares the RTM document?

Ans: Every T.E prepares RTM w:r:t their modules & finally lead consolidates & prepare a RTM w:r:t release.

RTM Template:

Req#	Module Name	High Level req.	Low Level req.	Test Case Name
5	LOANS	5.1. Personal Loan (PL)	5.1.1. PL for pvt Emp.	Cheeta_2.0_PL_pvt_Emp
			5.1.2. PL for Govt. Emp.	X
		5.2. Home Loan (HL)	5.2.1. HL for pvt Emp	- - - - - - - -
			5.2.2. HL for Govt. Emp.	- - - - - - - -

X → Missing test case

↳ Given to write with in very short span of time & reviewed by T.L.

Test Document:

1. Test Plan
2. Test Scenario
3. Test Cases
4. RTM.

Test Execution Report (TER):

→ It is a report prepared by the lead which defines the stability of the application.

→ It consists the information like the no. of test case written, executed, pass, failed & their percentage.

Scanned with CamScanner

→ We handover this report to the customer to inform about the stability of the product.

Module Name	Total # of TCs written	Total # of TCs executed	Total # of TCs pass	Total # of TCs fail	Pass %	Fail %
Loans	480	480	476	4	98	2
Sales	490	490	490	0	100	0
Market	370	370	368	2	99	1
Total	1340	1340	1334	6	98.7	1.3

TER | LOANS | SALES | MARKET

↓

TER [MARKET]

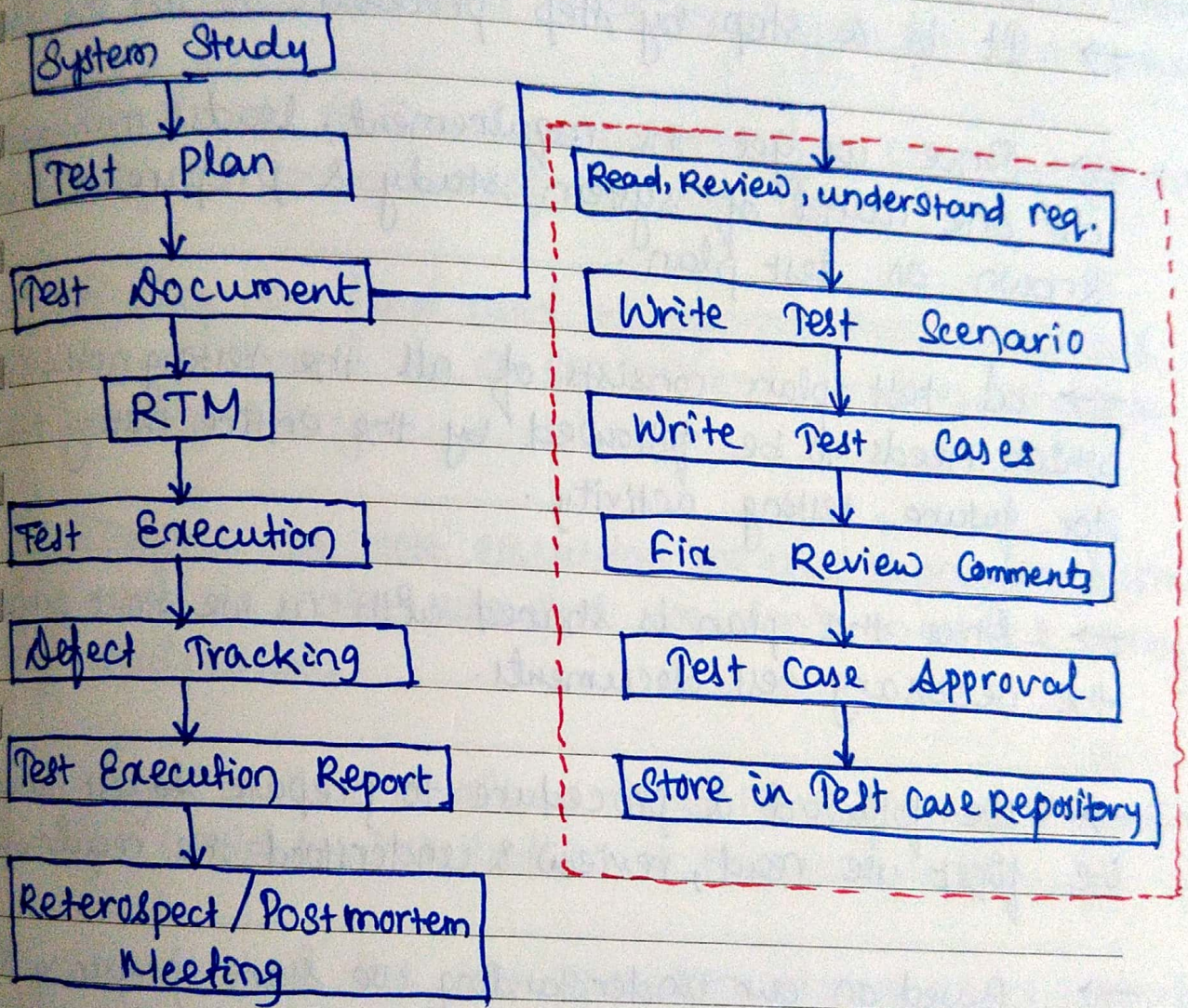
SL#	TCs Name	Status	Comment
1	—	Pass	⋮
2	—	Pass	⋮
3	—	Pass	⋮
4	—	Fail	Reason
⋮	—	⋮	⋮
376	—	⋮	⋮

Exit Test Plan → 97%

T.E.R → 98.7% ↓ Increased

Scanned with CamScanner

STLC (S/w Test Life Cycle):



Scanned with CamScanner

STLC:

→ It is a step by step procedure to test the s/w.

→ Once we get the requirement, leads & managers do one round of system study & prepare a document known as test plan.

→ A test plan consists of all the testing activities which needs to be followed by the entire testing team for future testing activity.

→ Once the plan is shared with us we start preparing the necessary test documents.

→ We followed a procedure to prepare test document i.e first we read, review & understood the requirement

→ Based on our understanding we start preparing a high level document known as test scenarios.

→ Once the flow is cleared, we start writing the test cases which consists of in detail step by step procedures to test the application.

→ The written test cases were sent for the peer review where they were reviewed & send us back with the necessary review comments.

Scanned with CamScanner

→ We did the necessary changes, fixed the review comments which is required & send back once again for review process.

→ This process continued until both of us accepted for the accuracy or correctness of the test cases.

→ Finally, the lead used to approve these test cases & they were stored in a centralized location known as test case repository.

→ Before the test execution we used to prepare an RTM document to make sure that we don't miss out testing any requirements.

→ By this time developer would have finished the coding & build would be installed on to the test server.

→ We started the test execution process where we perform all the types of functional & non-functional testing.

→ During the execution process we encountered many bugs which were logged & tracked by using some defect tracking tools.

→ The defect tracking continued until all the bugs were fixed & the application was stable & working according to the req.

→ Finally we prepare a test execution report which defines the stability of the product & the product is handed over to customer.

→ In the last we had a retero spect meeting where we discuss pros & cons (+ve & -ve or adv. & disadv.) of the current process & any improvement that can be done in the future releases.

Scanned with CamScanner

Defect Tracking:

Defect : Deviation from requirement
Bug : Informal name of defect
} Used by T.E

Error : Mistake in business logic/code → Used by dev

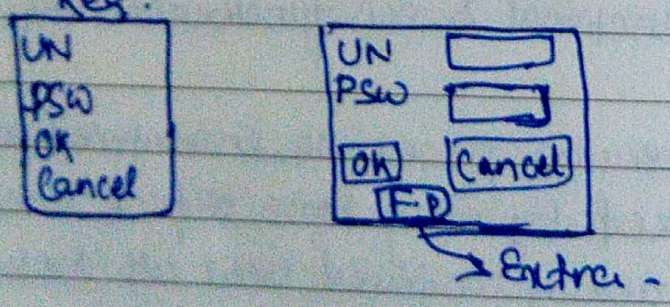
Issue : Problem faced by customers → Used by customer

Failure : Defects leads to failure of application

Why defects?

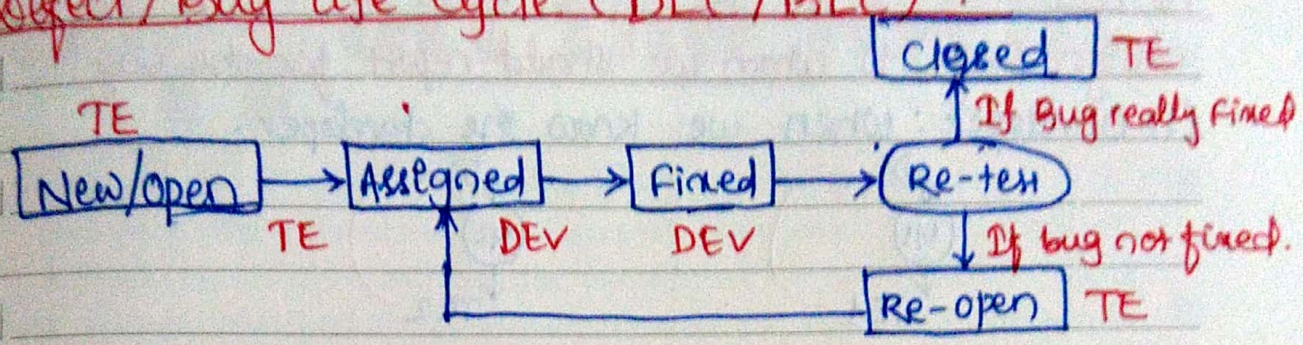
* Wrong Implementation.
* Missing Implementation
* Extra Implementation.

Ex: Req.



Scanned with CamScanner

Defect/Bug life Cycle (DLC/BLC):



→ It indicates the entire life cycle of a bug right from a stage it was found, fixed, retested & closed.

→ The status of the bugs are new or open, assigned, fixed, closed or re-opened.

* As soon as we get a bug, the status is "new or open".

* The bug is reported to the concerned person by changing the status to "assigned".

* The developer does the necessary changes in business logic & makes the status as "Fixed".

* Once the bug is fixed the test engineer re-test the bug & changes the status accordingly i.e
  (I) Closed : If changes done properly.
  (II) Re-open : If the bug is still exist.

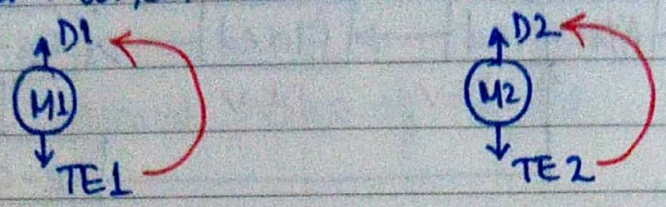
→ This process continues untill the bug is fixed & the status is closed.

Scanned with CamScanner

NOTE:

Assign Bug [to whom we should report for the bug]

(I) Developer : when we know the developer.



(II) Development Lead : If we don't know which developer developed the module.

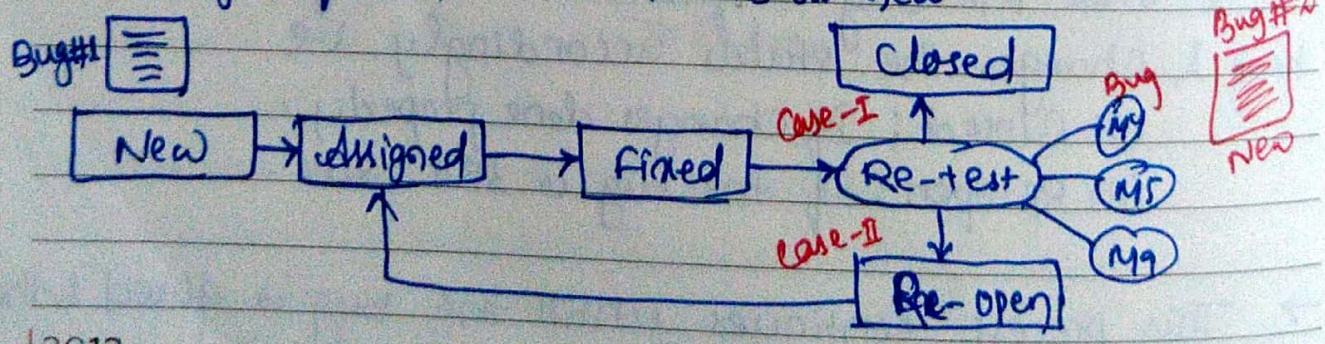
(III) Test lead : (a) when no interaction with developer team (i.e s/w developed in one company & tested at other)
(b) Bug review (i.e normally for freshers)
It is time consuming.

Case - I :

If there's a change in BL & Bug#1 is fixed & after retesting, the status is closed. If we get another bug then the status is new for the new bug.

Case - II :

If the bug is not fixed means the bug is still exist at that time the logged bug will be re-open as status & log a new bug (if found) as the status as "new".

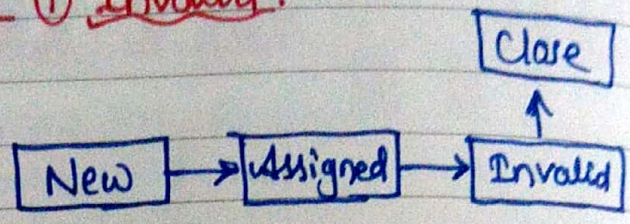


Scanned with CamScanner

Other Status of Bugs: ① Invalid:

(I) Bug Report

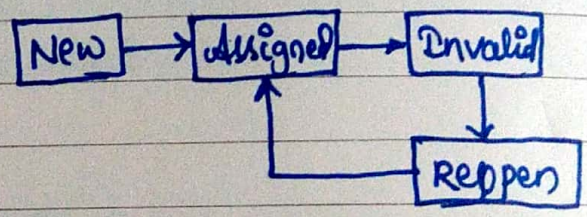
I. Login
II. Compose, send it
Obs: Mail not displayed in "Thrash"
Comment:



Here developer changes the status to invalid because the T.E may misunderstood the requirement.

(II) Bug Report

1. Login
II. Compose, send it
Obs: mail not displayed in "S.I".
Comment



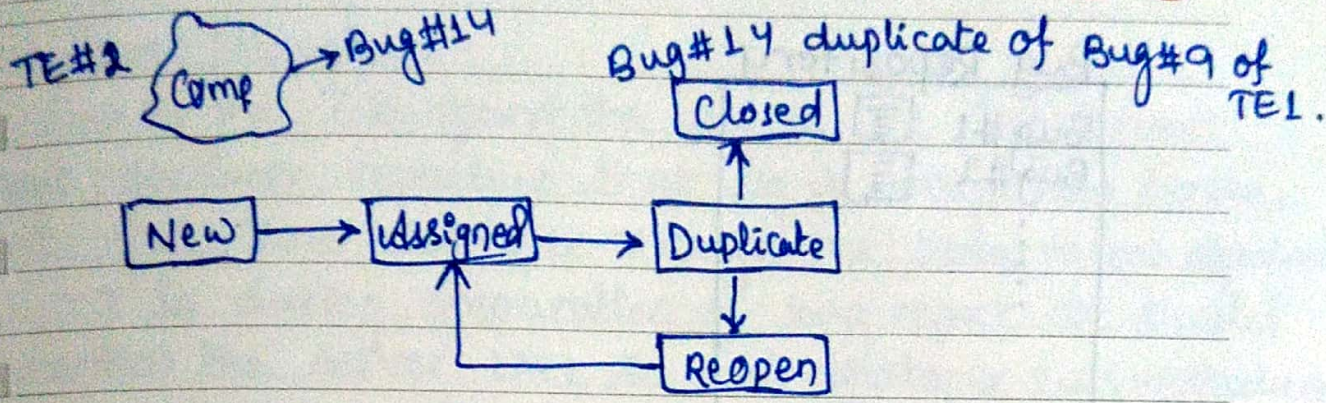
Here the developer may misunderstood the requirement, so T.E change the status to reopen & gives a comment.

② Duplicate:

→ When a bug is logged by another T.E while testing & other TE want to log the same bug then at that time developer change the bug status to duplicate.

→ This generally happens when the modules are dependent or the common modules.

Scanned with CamScanner



Here TE#2 logged a bug (which is bug#14) which is similar to bug#9 by TE#1. So at this time the developer change the status to duplicate for TE#2.

If TE#2 logging the bug#14 which is not similar to bug#9 at that time he reopens & assigned again to the developer.

How to avoid the duplicate:

As soon as we get a bug search in bug report

Case-I: If bug exist, don't log a bug.

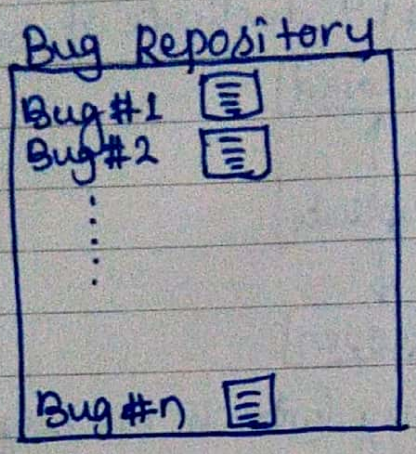
Case-II: If bug doesn't exist, log a bug, report to the developer & make CC to all the testing team.

NOTE:

Bug Repository:

~~If the~~ It is the centralized place where all the bugs found by the T.Es are kept or stored.

Scanned with CamScanner



NOTE:

When the bug status is

(I) Open → Don't log a bug

(II) Assign → Don't log a bug.

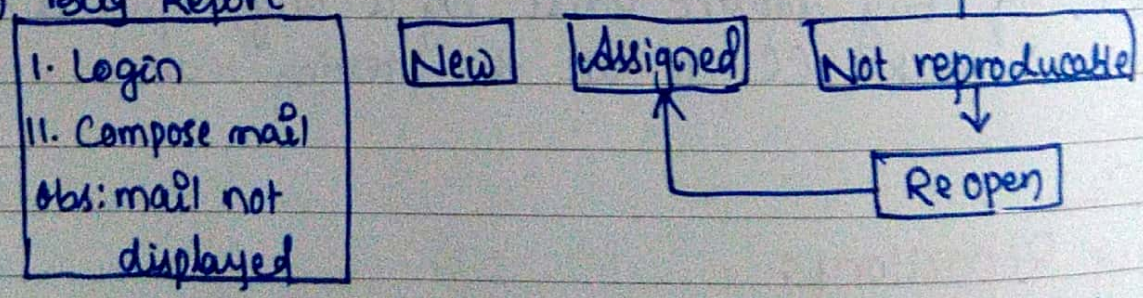
(III) Fixed → Don't log a bug.

(IV) Reopen → Don't log a bug.

(V) Close → Log a bug.

③ Not Reproducable:

(I) Bug Report



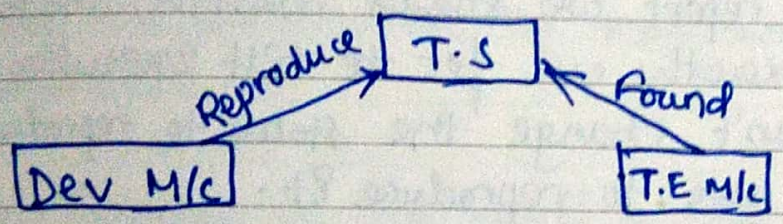
Here the T.E forgot to write S.A.D & forgot to write display in draft. But when developer reproduced the bug he didn't find the bug. Developer changes the status to ~~invalid~~ "not reproducable" because of the incomplete steps given by TE.

Scanned with CamScanner

(II) Server Mismatch:

→ If the T.E found the bug in one Test Server but developer reproduce that bug in another test server, at that time developer changes the status to "not reproducable"

→ So during preparation of bug report we should mention the server URL so that developer can reproduce the bug & fix it.



(III) Platform mismatch:

→ If T.E found the bug one combination of platform i.e XP with I.E but developer tries to reproduce that bug in another platform like win7 with Mozilla, at that time he may not get the bug or unable to reproduce the bug. So at that time developer changes the status to "not reproducable".

→ So for proper convinient we should mention the OS as well as the browser.

(IV) Data mismatch:

TE
as user < UN, PSW
Try to delete
Not deleting

Dev
as admin < UN, PSW
Try to delete
deleted

Scanned with CamScanner

→ In this case the T.E try to delete something by loging as a normal user & ~~was~~ unable to delete but developer able to delete that one by logged as an administrator. i.e in this case the T.E uses the application as an other user & developer uses/reproduces that as an another user.

→ So in bug report we should mention the test data what we exactly use for the test execution, so that developer can't change the status to reproducable because he is able to reproduce it.

(V) Build Mismatch:

Suppose as soon as T.E find a bug he reported to the developer. Developer analyses some bugs & fix them in that current build. But ~~some~~ still some more bugs he unable to reproduce in the same build.

I [B001] ← 5bugs
↓
Fix only 3 bugs → Reproduced in B001

II [B002] TE → Retests 3 bugs
↓
Fixes 2 more bugs → Reproduce B002
1 bug fixed automatically

(VI) Inconsistency Bug:

→ If T.E found one bug. While reproducing again for that bug he didn't find. Means some time he find the bug & some time he doesn't find. That kind of bug is called as inconsistency bug.

→ So as soon as we find a bug we should take the screenshot of that bug so we'll be in saferside.

Ex: Suppose a T.E found a bug. As soon as he found a bug he rechecked that for the confirmation. 2nd time the bug was not found. 3rd time he checked again still he didn't find. 5th time again he found. Again he checked that feature but he didn't find any bug. He passed that feature & the product was released. After release the customer found the bug & he reported at the company. So company will blem the T.E who tested that particular feature.

Scanned with CamScanner

④ Can't Fix:

→ The bugs won't be fixed normally are the minor bugs.

New → Assigned → Can't fix

→ This kind of situation happens generally for the following causes.

(I) Bugs present in the core of code i.e as soon as the project is feasible at the timing design the members prepares the framework for the project.

Ex: If we want to re-build a house, we should change the walls not the pillars. If we break/modify the pillars then the entire house will be lost.

(II) Cost of fixing the bug more than keeping it.

09 SUNDAY Ex: Suppose the bug is minor, it needs more source & more technology which costs more according to the entire project. This kind of bug won't effect on the functionality. So we don't expend more for such kind of bugs.

Scanned with CamScanner

(III) No technology support i.e the programming language which we use that may not support such kind of functionality.

The above three causes are the root cause for the can't fix status.

⑤ Deferred:

New → Assigned → Deferred → Minor (Should be fixed in the next release)

Ex: B001 | B002 | B003 - - - - - - - - - B025 | B026 | B027

B002 → Deferred
It can't be deferred here

B025 → ~~Bug #1 (or) Bug #16~~
Here it can be deferred because 15 bugs are there in developer's plate.

→ Deferred bugs are the minor bugs i.e whenever the the bug is minor & he doesn't have time to fix that because of time at that moment developer makes the status deferred.

→ It can't be fixed in the same release because of the time constraint.

→ It should be accepted by both developer & T.Es.

→ The bug status is deferred, won't accept in the initial stage but it can be accepted in later stage.

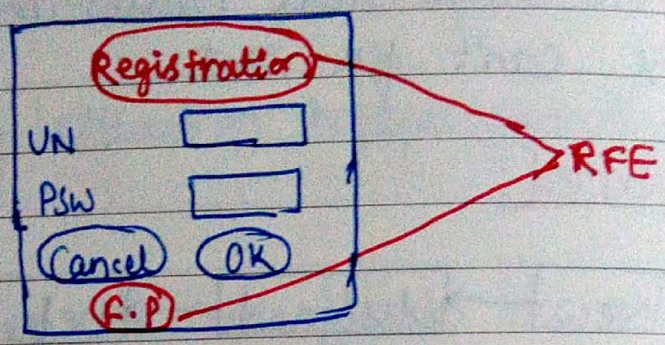
Scanned with CamScanner

Request for Enhancement (RFE)

New → Assigned → RFE

Ex: Req

(1) Login
1.1 UN
1.2 PSW
1.3 Cancel
1.4 OK



Application is working as per the requirement but it doesn't have the user friendlyness.

NOTE:

1. **Invalid:**
   Developers doesn't accept it as a bug.
2. **Duplicate:**
   The same bug has been reported by different T.E multiple times.
3. **Not Reproduciable:**
   Developer accept it as a bug but not able to reproduce the bug due to some reasons.
4. **Can't Fix:**
   Developer accept it as a bug, able to reproduce as well but can't do the changes due to some constraints or limitations or drawbacks or problems.
5. **Deferred:**
   Developer postpone the bug to the future releases

Scanned with CamScanner

due to time constraints.

6. R.F.E:

These are the suggestions given by the T.Es towards the enhancement of the application.

Severity:

→ The impact of the bug in the application is known as severity.

→ It may be the blocker, critical, major, minor.

Priority:

→ This indicates which bug is to be fixed 1st.

→ It may be the Urgent (P1), high (P2), Medium (P3) Low (P4).

Ex:

① Amt Transfer → Click → [ ] → Blank page — Blocker P1

we can't proceed in our testing.

② Login as User A → Click on Amt Transfer → Click on Transfer → Confirmation Message is displayed Checking Balance → Not showing correctly — Critical P1

i.e Here functionality is not working properly

Scanned with CamScanner

③ Login as User A
Click on Amt Transfer
Click on transfer → Confirmation message is not displayed
Checking Bal → Showing correctly.
Here functionality is working but gives wrong or no information.

Major P3

④ All UI issues → Minor P4

⑤ Message not displayed in S.I
displayed in inbox

Major P2 or P3

⑥ #1 in S/w testing.
www.wipo.com → Page

Minor P2.

NOTE:

1. Blocker: When we are not able to proceed with out testing activity.
2. Critical: The main functionality is not working according to the requirement but we are able to proceed with testing.
3. Major: Able to proceed with testing, main functionality is also working but the supporting functionalities are not as expected.
4. Minor: All UI issues are minor bugs.

Scanned with CamScanner

Bug Report Template:

Manual bug report generally prepared in Excel sheet.

Bug # Id :
Test Case Name :
Module Name :
Feature # :
Status :
Reporter :
Assigned to :
CC :
Date :
Release :
version :
***Severity :
***Priority :
Test Data :
Server / URL :
OS & Browser :
Build # :
Attachment of screenshots :
Brief Description : Mail not displayed in S.I of Sender.
Steps to Reproduce : (I) Login as user A
(II) Compose mail, send to user B
(III) Confirmation msg. is displayed
(IV) Also mail displayed in inbox of User B
(V) Click on S.I of user A.
Observation : Mail not displayed
Expected Result : Mail should be displayed in S.I of user A.

Scanned with CamScanner

17.6.14

Drawbacks of Manual Bug Report:

1. Manual Error
2. Searching is Tough
3. Time Consuming.
4. Logging & Tracking is tough
5. Not Secured.

Solution for the drawbacks:

Bug Tracking tool like,

1. BugZilla
   BugZini
2. BugZero
   Jeera
3. Mantis
   Quality Center → Test Management Tool.

Logging a Bug:

→ At 1st we've to enter URL, then Home page will be displayed

→ Login as own user name & password.

→ Click on new bug link. Bug report template will be displayed.

→ Fill the necessary fields & make it commit.

→ After clicking commit a mail send to the necessary persons.

Scanned with CamScanner

NOTE:

→ UN & PSW created by manager or leads.

→ Tool is being customized.

→ During logging a bug, bug id won't be created. But after committing bug id will be auto created.

→ After the bug report is viewed by the developer, he changed the status as fixed with an additional comments.

Bug Template:

Bug Id : Auto Generated
Test Case Name : Disabled from 2nd time
Module Name : [ ]
Feature # :
Status : [ ]
Reporter : Auto Generated
Assigned To : Mail id
CC : Group id
Date : Auto Generated
Release : [ ]
Version : [ ]
Severity : [ ]
Priority : [ ]
Test Data : Disabled from 2nd time

Scanned with CamScanner

Server/URL : Disabled from 2nd time
OS & Browser : Disabled from 2nd time
Build : CurrentBuild → Click on this link
Attachment : Browse → Always in jpeg.
Brief Description :
Description :
Steps to reproduce :
Observation :
Expected Result :

Eg: Additional Comments :

Dev #4 5-July-2014 5:21

Bug Fixed in B007

TE# Date Time

Bug has been retested according to requirement hence changing status to close.

NOTE:

The additional comments will be changed by one phase to other phase like if developer fixed the bug, TE changes the status to close, TE changes the severity & priority, etc.

Scanned with CamScanner

Query a Bug:

To find the bug from the repository we search with certain parameters like,

Status :
Severity:
Priority :
Reporter :
Assigned to :
Date :
Module :
Version :
Description :

NOTE:

We have to ~~log~~ search a bug only when it's in fixed or reopen.

Eg:

Bug	Status	Description
Bug # 26	Close	
Bug # 74	Fixed	
	Close	
	Close	
	Fixed	
	Reopen	

Scanned with CamScanner

18.6.14

Advantages of the Test Cases:

1. To document knowledge of the TE :
Though TE may forget the scenarios or values.

2. To achieve maximum test Coverage i.e both +ve & -ve values.

3. To have test consistency i.e
   * proper flow
   * Same set of scenarios can be tested in multiple releases.

4. Time saving i.e Re used by same T.E in later stage or by new or other TE if author leave the company.

5. It becomes process oriented rather than person oriented because testing depends on person to person But once we prepare document It becomes process oriented.

6. Test Cases used by automation team to create test scripts.

Characteristics of the Test Case:

1. It should be in detail & proper flow.
2. It should be easy to understand i.e it should have simple language.

Scanned with CamScanner

3. It should have minimum steps & maximum Coverage.
4. It should catch maximum bugs due to the maximum coverage.

1.Q. How many test cases written in a day?
Ans: (i) Depends on the complexity of requirement.
(ii) Approximately 3 to 4.

2.Q. How many test cases you have executed in a day?
Ans: (i) Depends on the complexity of the test case.
(ii) Approximately 2 to 3.

3.Q. How many test cases you have written in your Project?
Ans: Approximately 90 to 110
2-3 months → 4 to 6 weeks → 20 to 30 ~~weeks~~ days
→ 60 to 80 or 90 to 120 testcases.

Scanned with CamScanner

22 JUNE SATURDAY

19.6.14

Regression Testing:

Rel-1 [5 Features]	Rel-2 [5+3]	Rel-3 [5+3+4]
300 T.Cs 4 TE 4 Months	250 New TCs 4 TE 4 Months 150 old TCs from Rel-1	320 New TCs 4 TEs 4 months 140 old TCs Rel-2 160 old TCs Rel-1
300 TCs.	(250+150 = 400 TCs)	140+160+320 = 620 TCs.

---- Rel-N

Regression Test cases ← Executing — Regression Testing

Changes
* Addition of New Features
* Deletion of old Features
* Modification

Needs → Code change.

NOTE:

Regression Testing takes place only from the 2nd release onwards.

23 SUNDAY

Re-executing the old test cases across multiple releases to make sure that changes i.e addition, deletion, modification has not affected the existing functionality is known as regression testing.

2013

Scanned with CamScanner

Regression Test Cases:

The old test cases which may get affected due to the changes & which needs to re-execute is known as regression test cases.

NOTE:

We don't write a separate regression test cases, rather we pick them from the existing test cases repository.

Challenges of Regression Test Cases:

→ Identify the impact area.

→ Time Consuming.
* Test Cases increases release by release.
* Less Resources.

→ No accuracy.: * Monotonous Task
* Repeatative Task.

Types Of Regression Testing:

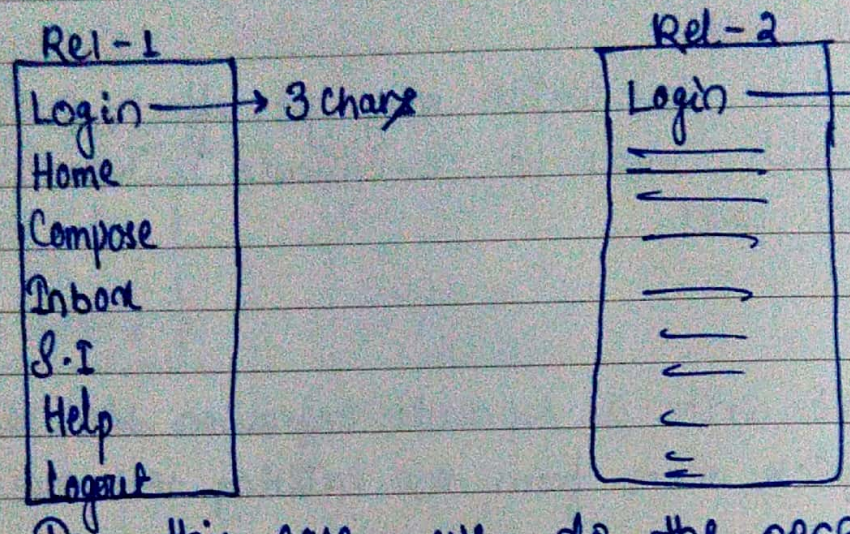
1. Unit Regression Testing:

→ Here we are checking only the changes but not the impact area.

→ Here we cut & paste the test cases after doing the modification (where required) from release-1 to release-2.

Scanned with CamScanner

Eg:



In this case we do the necessary change in the testcase of release-1 & cut it & paste in the testcase repository for release-2.

② Regional Regression Testing:

→ Here we are checking the changes and we also check the impact areas.

→ The impact area ~~than~~ can be found by doing Impact analysis.

→ Eg: If there is code change happened than we've to check for inbox, sentitems because these are the impact areas.

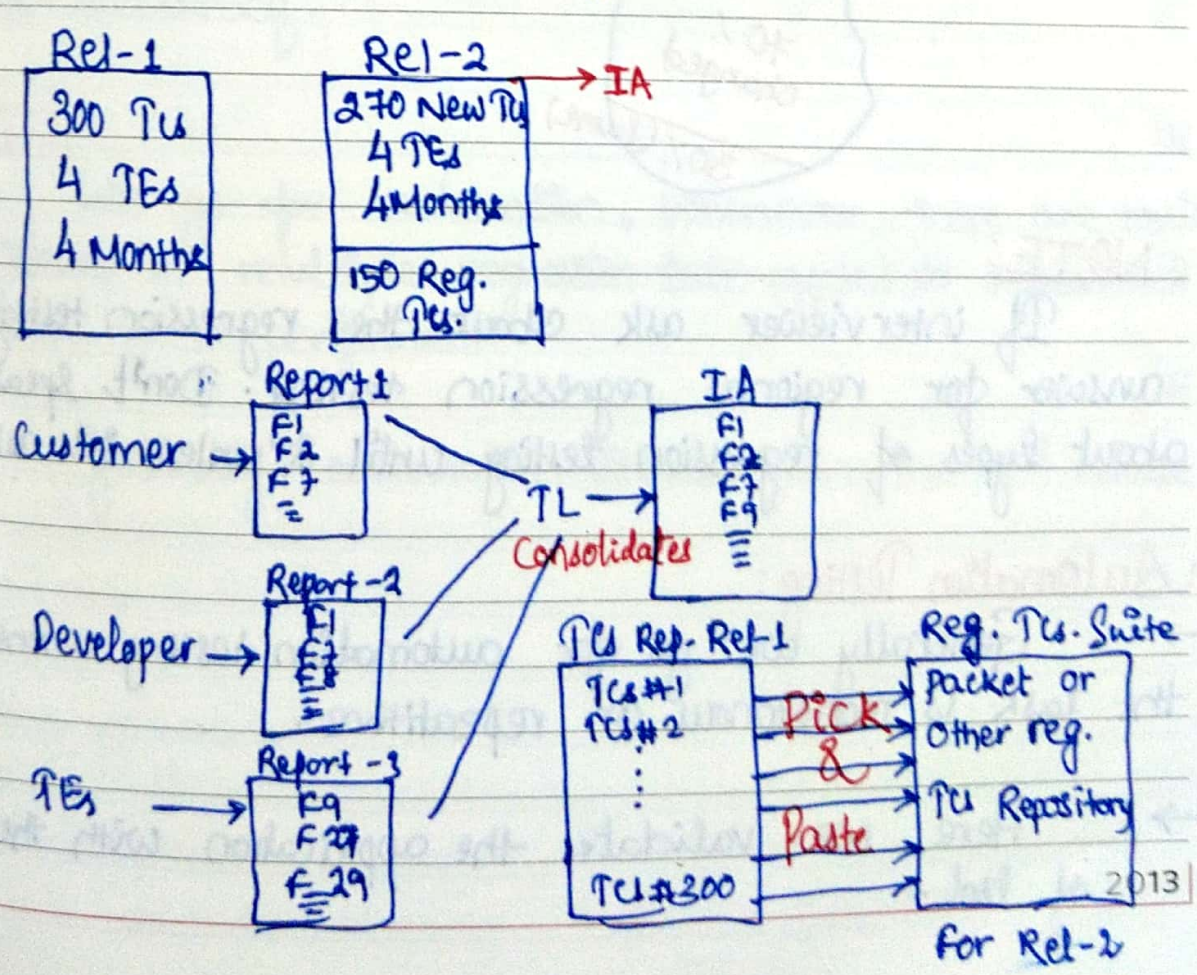
→ Generally the impact area can be identified by the impact analysis.

Scanned with CamScanner

→ Normally the impact analysis can be done by the customer according to his "business knowledge", the developer according to his "Coding knowledge", and most importantly by the test engineer according to his "product knowledge".

→ The different different personalities do ~~their~~ impact analysis & prepare the corresponding reports & send to the lead.

→ Finally lead consolidates all the reports & pick the test cases & store those in a centralized place known as "Regression Test Suite".



Scanned with CamScanner

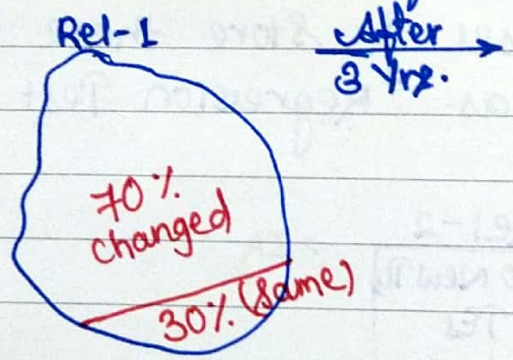
3. Full Regression Testing:

→ Here we are testing the changes with all the old features from the previous release.

→ We are testing in this way because the impact area is huge.

→ Here the most of features are being changed.

→ Basically this type of testing done for the critical application i.e. life or machine critical.



NOTE:

If interviewer ask about the regression testing then answer for regional regression testing. Don't speak about types of regression testing until & unless it's asked.

Automation Testing:

→ Generally we go for automation testing whenever the task is monotonous or repeatitive.

→ Here we validate the application with the help of tool.

Scanned with CamScanner

→ It is of 2 types.

(I) Function Testing: QTP, Selenium, WinRunner, SilkTest, etc.

(II) Performance Testing: Load Runner, JMeter, NeoLoad, eLoad, WebLoad etc.

→ Except Selenium & JMeter (i.e. open source or free) all others are licensed.

→ Regression testing means we check the changes & the impact area with the help of tool & we use functional testing tool.

→ We use tool to save the time & to maintain the accuracy.

NOTE:

(I) We go for automation, whenever there are multiple releases or multiple regression test cycles or repeatitive task to be performed.

(II) Generally automation starts from the 2nd release.

Scanned with CamScanner

Automation Life Cycle:

Rel-1
350 TCs
4 T.Es
4 Months

Rel-2
300 New TCs
4 T.Es
4 Months
+
150 Reg. TCs.
↓
15 days

→ IA
Manual TE [Writing New TCs]
Automation TE [Creating Scripts]

To create Script:

Things Needed to create Scripts

TCs. → Stable Application → Tool

Why Automation:

Login

- Manually Testing (1hr) let
- Automation Testing
  - Creating Script (2hr) → Can be reused.
  - Executing Script (1min)

How Execution Takes Place:

TCs Name	Test Script Name	Status
TCs#1	TS #1	Pass
TCs#2	TS #2	Pass
⋮	⋮	⋮
TCs#150	T.S #150	Fail

→ Retested Manually by Manual T.E. If bug is there report it to developer, after fixing it, retest it.

Scanned with CamScanner

27.6.14

RECORD THE LIST OF THINGS YOU MIGHT WANT TO DO.

NOTES

Advantages of Automation:

1. Accuracy is maintained since it is not a monotonous task for the tool.

2. Time Saving

3. Re used

Process of Automation Life Cycle:

→ Generally we go for automation in the case of repeatitive task or multiple releases.

→ From the 2nd release, the manual team starts working on the new features & the automation team starts creating the scripts for regression test cases.

→ To create this script we need testcases, tool & the stable application.

→ Once all the scripts are created according to the test cases, the execution is done on the new application.

→ During the execution, the encountered bugs are reported to the developers which needs to be fixed & sent back for retesting.

→ This process continues until all the ~~bugs~~ regression test cases which are automated are passed.

Scanned with CamScanner

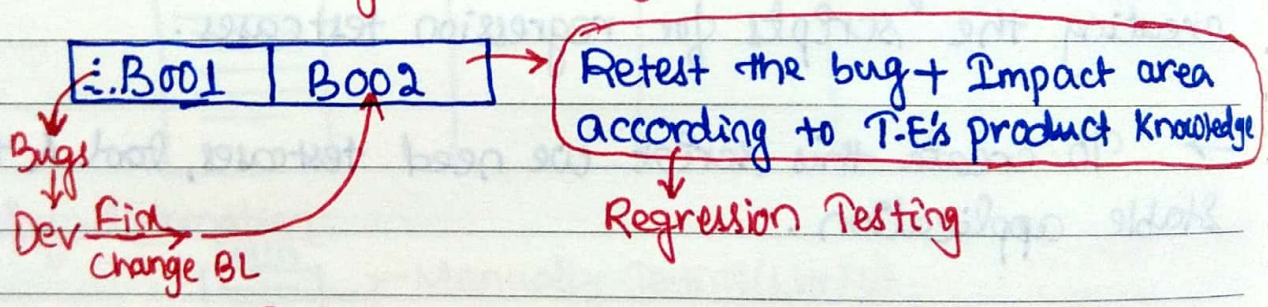
Test Cases Can't be Automated:

1. Unstable Feature
   * New Features
   * Changing Features
     Eg: Change in compose & inbox will be tested even if inbox having the changes.

2. No technology Support:
   Eg: Captcha, virtual keyboard.

3. Test cases which is having complex code, time consuming or lots of human involvement is required.

Build Level Regression Testing:



Regression Testing:

Re-executing the old test cases across multiple releases or builds to make sure that enhancements or bugfixes doesn't affect the existing functionalities is known as regression testing.

Why or When:

Whenever developer does some code changes to make sure that these changes don't affect the existing functionalities, we go for regression testing.

Scanned with CamScanner

M	T	W	T	F	S	S
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

Retesting & Regression Testing:

Retesting:

Bug → Fix → Retest only changes
Similar to unit regression testing

Regression Testing:

Bug → Fix → Retest bug + impact area

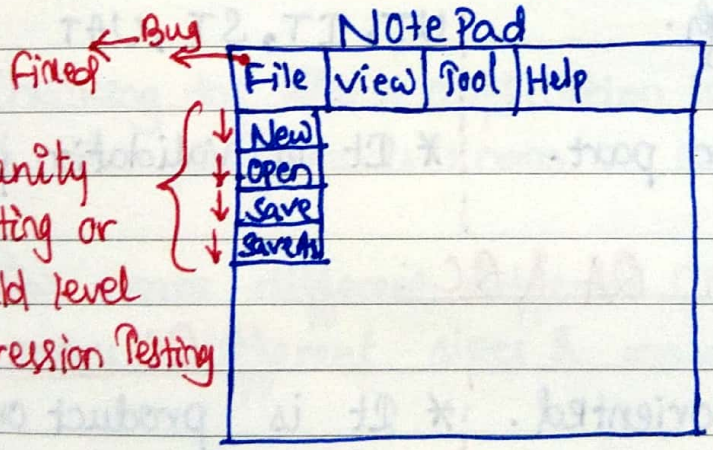
Scanned with CamScanner

28.6.14

Difference between Smoke Testing & Sanity Testing:

Smoke Testing	Sanity Testing
* As soon as the build is installed on the server.	* When bug fixes are done.
* It is shallow & wide i.e. High level.	* It is narrow & deep i.e. in detail.
* It can be documented or scripted.	* It can't be documented or scripted.

Smoke Testing



Difference between functional & non-functional testing:

Functional Testing	Non Functional Testing
* Here we are checking the functionality of the application according to the requirement i.e. Unit Testing, Integration Testing, System Testing, UAT.	* Here we are checking the application w.r.t. the non-functional factors like, performance testing, usability testing, compatibility testing
* Can be document	* ~~Can't~~ Documented after result comes out.

Levels of Testing or Testing Methodology:

1. Unit Testing : Component Level
2. Integration Testing : Data Flow
3. System Testing : End-to-end
4. UAT : Business Scenario.

Difference Between Static & Dynamic Testing:

Static Testing	Dynamic Testing
* Here we check the code/s/w without executing/validating.	* Here we check the code/s/w by executing/validating.
* It includes the activities like review, walk through.	* It includes the activities like UT, IT, ST, UAT
* It is verification part.	* It is validation part.

Difference between QA & QC:

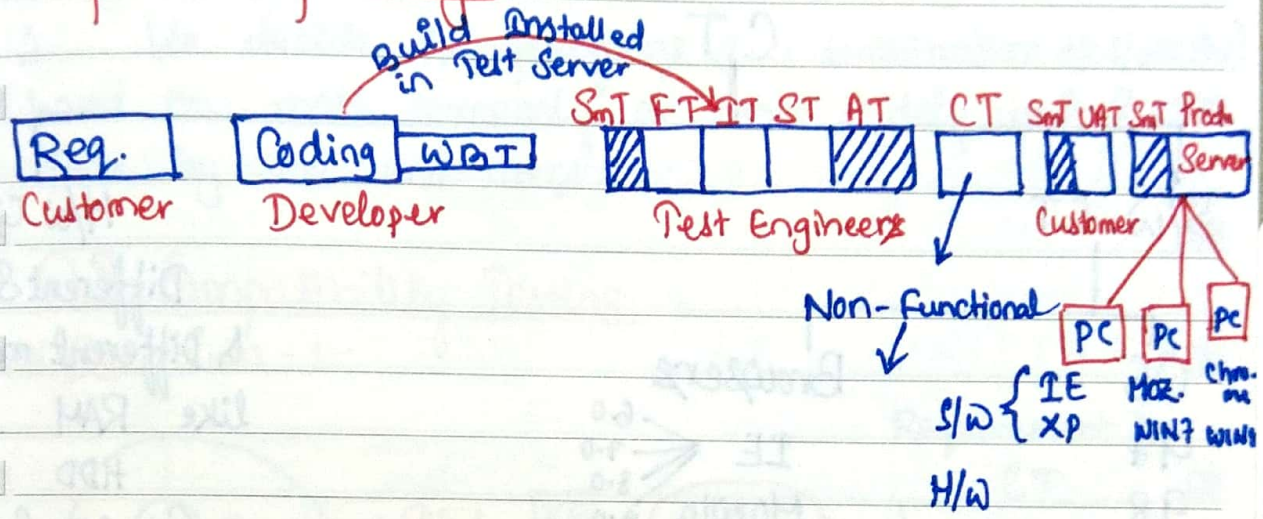
QA	QC
* It is process oriented.	* It is product oriented.
* We check if product is developing right.	* We check if the product is developed right.
* It includes activities like review, walk through.	* It includes activities like UT, IT, ST, UAT.
* It is verification part.	* It is validation part.
* It is preventive method. Finding the bugs before product is developed.	* It is detective method Finding the bugs after the product is developed.

Scanned with CamScanner

21.5.14

NON-FUNCTIONAL TESTING

Compatibility Testing:



→ Checking the ~~S/w~~ * application in different different S/w & H/w platforms is known as compatibility testing.

→ S/w means different different OS & browsers and H/w means different sizes & makes.

Why:

Once the S/w is released into the market, it can be accessed by the real users or end users in different S/w & H/w configurations & during this process, they may encounter some compatibility issues.

To cover these issues, we do one round of Compatibility testing.

Scanned with CamScanner

When

We do Compatibility testing once after the application is functionally stable.

CT

- S/W CT
  - OS
    - 97
    - 98
    - 2000
    - XP
    - VISTA
    - WIN7
    - WIN8
    - WIN 8.1
    - MAC
    - UNIX
    - UBUNTU
    - LINUX
    - Red Hat
    - Suse
    - Sun Solaries
  - Browsers
    - IE (6.0, 7.0, 8.0, 9.0)
    - Mozilla
    - Chrome
    - Safari
    - Opera
    - EPIC → By Indians
    - Netscape Navigator
    - UC
- H/W CT
  - Different Sizes & Different makes like RAM, HDD, Graphics Cards, Processor, etc..

Most Commonly Used i.e. 98% Use

Rest Use % = 2%

Scanned with CamScanner

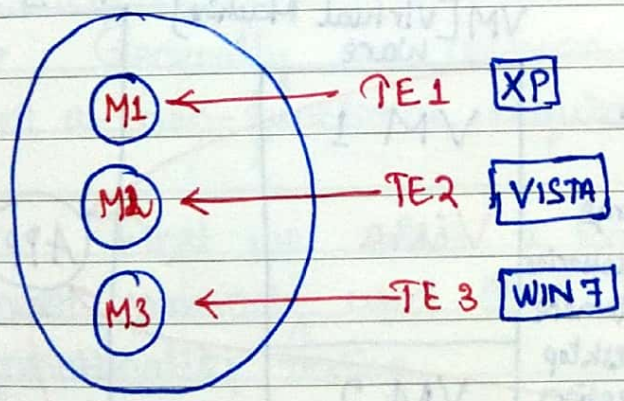
NOTE:

1. The most challenging thing in C.T is to test for all the combinations of S/W as well as H/Ws.

2. We decide the platforms (i.e. combinations of S/W & H/W) based on most commonly or most widely used S/Ws & H/Ws by the end users.

OS Compatibility Testing

* Scenario-1:

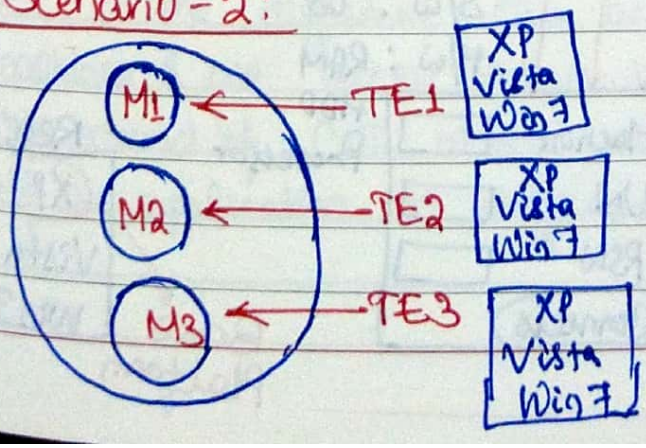


Requirement for CT

XP
Vista
Win 7

Here in this case three different systems are with different different OSs. So here we can't do CT because M1 may be compatible for XP but may not for Vista & Win 7 & so on.

* Scenario-2:



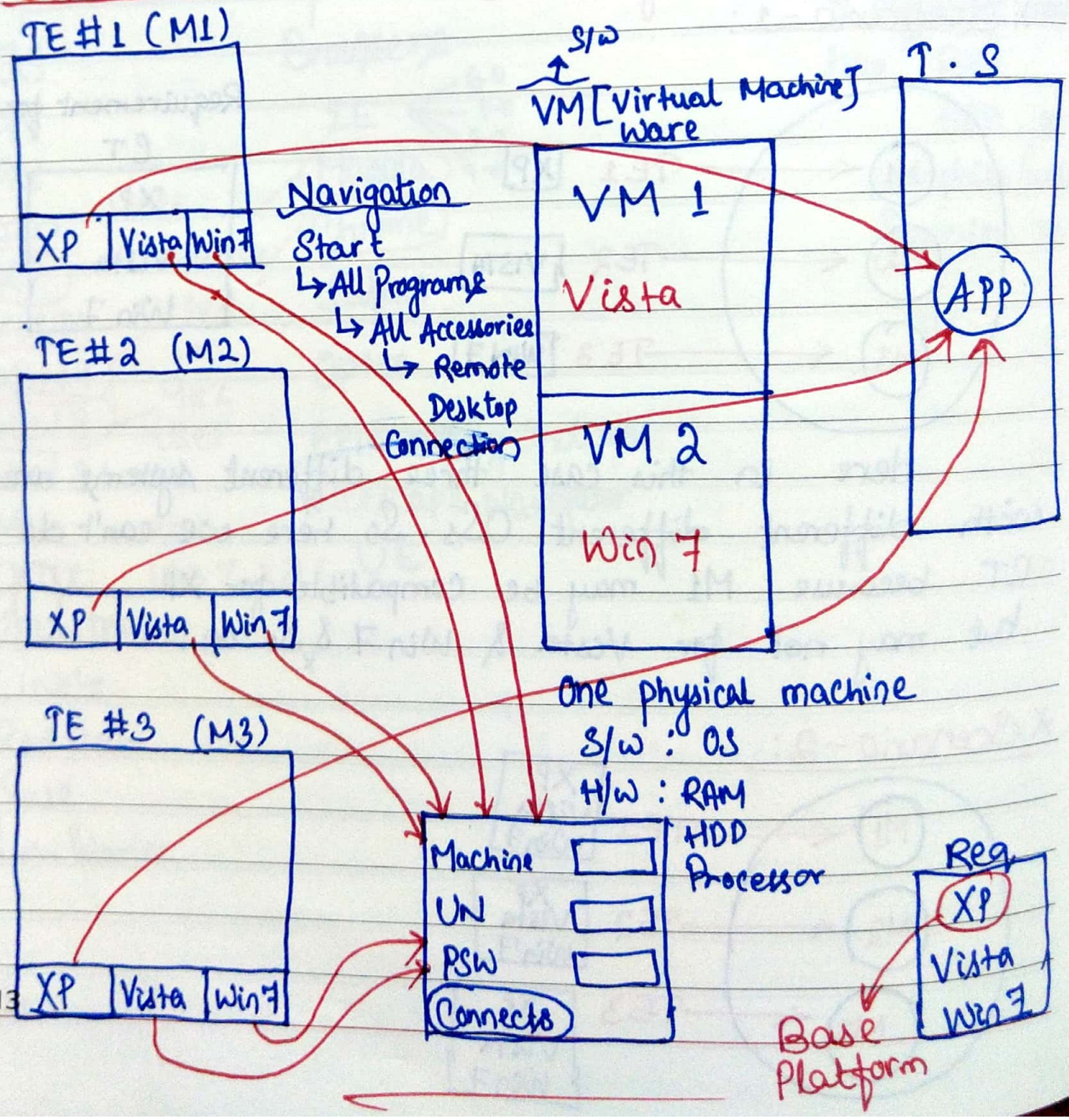
Scanned with CamScanner

* In the above scenario, we've to restart again & again to the system to reproduce the bug in different OS.

* For installing many OSs, a highly configured system is required which is very expensive.

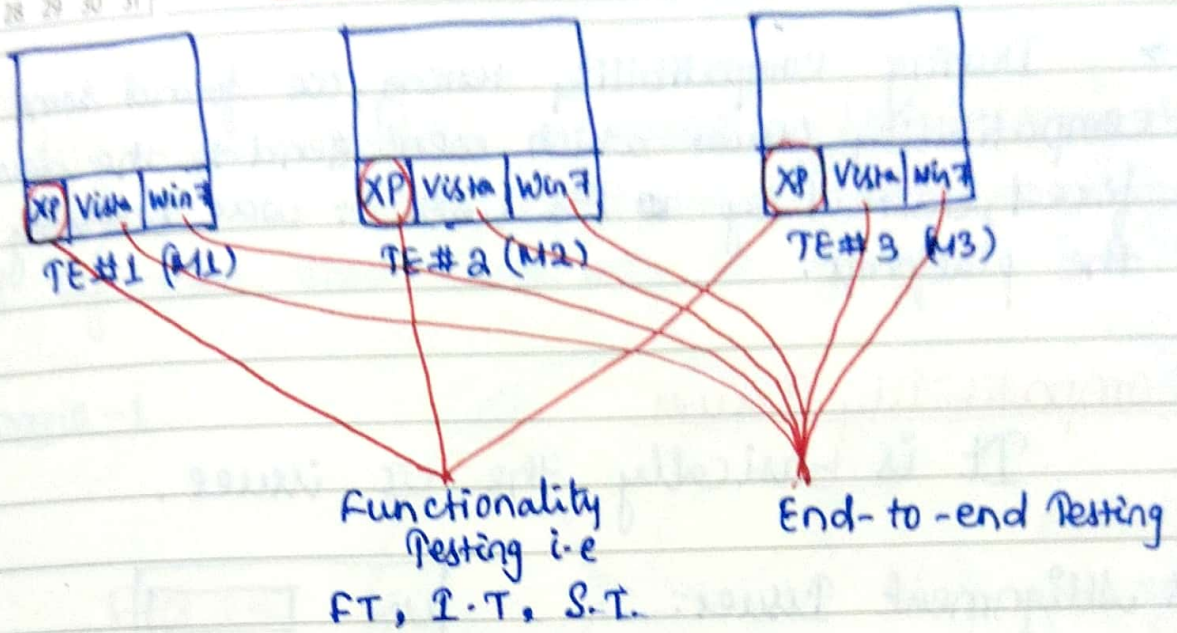
* It is a time consuming process to test the application.

* Scenario-3:



Scanned with CamScanner

22.5.14



Process:

→ Generally the customer gives both the functional as well as non-functional requirements.

→ First we select a base platform based on the most commonly used OS & started one round of functionality testing.

→ Once the application is functionally stable, we start with compatibility testing where we test the application on different different platforms.

→ The different OSs are installed on to the VM machines & we access these machines remotely & perform one round of C.T where we checked end-to-end flow of an application on different S/w & H/w platforms.

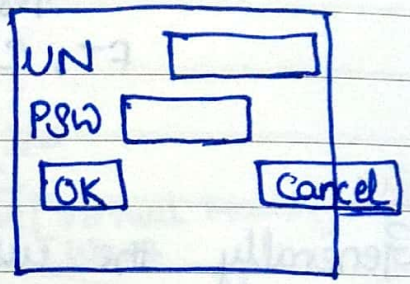
Scanned with CamScanner

→ During compatibility testing, we found some compatibility issues which were send to the developers, fixed, retested by TEs until it was working on all the platforms.

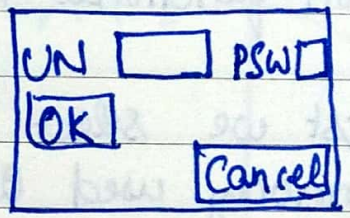
Compatibility Issues:

It is basically the UI issues.

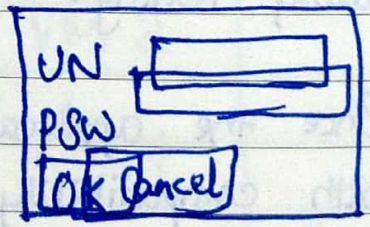
* Allignment Issues.



* Scattered Object



* Overlapped Object



* Color & Font Size

* Broken links.

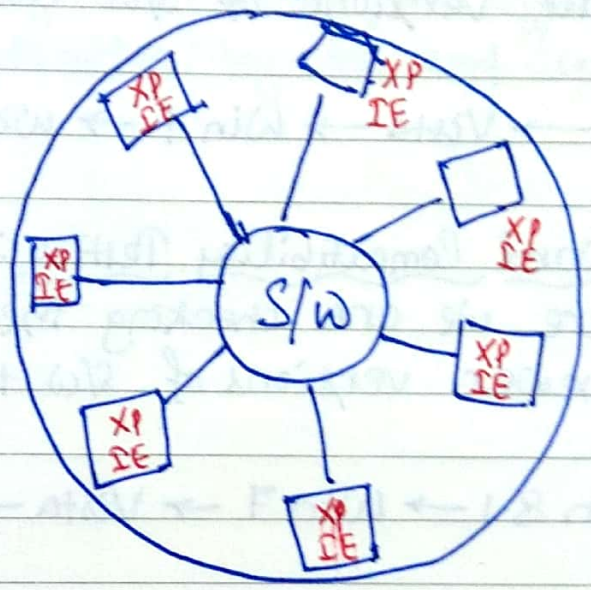
Scanned with CamScanner

Which:

Any S/ws which is used by/accessed by the end users & we don't have control over the platforms used by them. These cases only we go for C.T.

Scenario-1

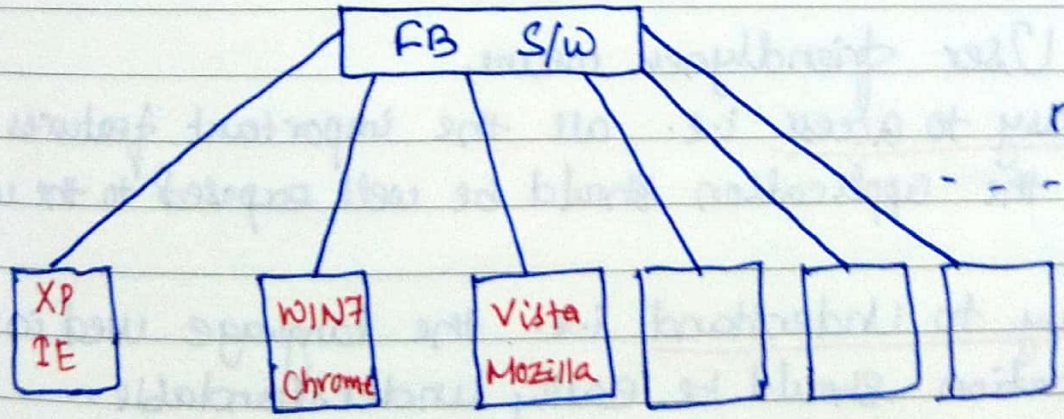
Fed ex



1000 Users with XP & IE

In this case C.T is not required.

Scenario-2



n-users

In this case C.T is required.

Scanned with CamScanner

Types of Compatibility Testing [According to S/w]:

① **Forward Compatibility Testing:**

Here we are checking the compatibility issues from older versions of S/w towards new versions of S/w.

Eg: XP → Vista → Win 7 → Win 8.1

② **Backward Compatibility Testing:**

Here we are checking the compatibility issues from newer versions of S/w towards old versions of S/w.

Eg: Win 8.1 → Win 7 → Vista → Win XP.

Usability Testing:

→ Checking the user friendlyness of an application is known as usability testing.

→ User friendlyness means,

* Easy to access i.e. all the important features of the application should be well exposed to the user.

* Easy to Understand i.e. the language used in the application should be easily understandable.

* Look & feel should be good i.e. color & font sizes should be attractive.

Scanned with CamScanner

* Navigation steps to perform a task should be easy.

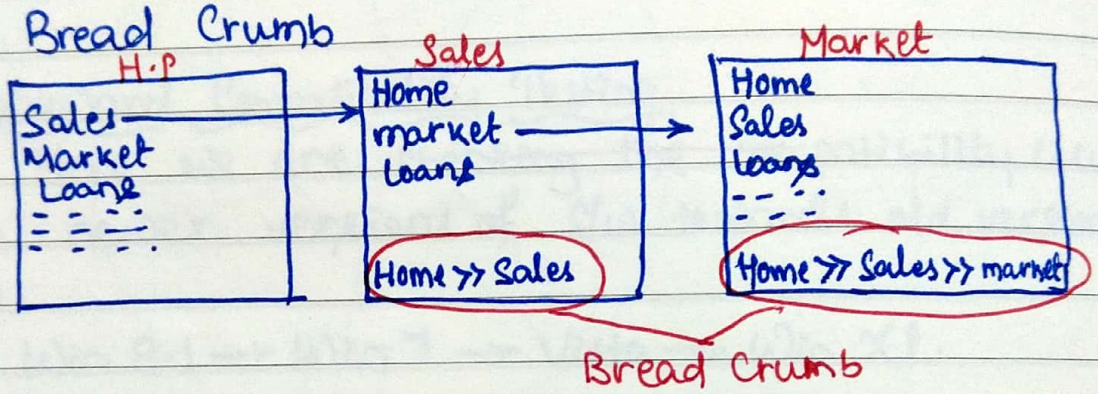
* It should be also be faster in access.

→ Overall user friendly means, the efforts needed to learn the application by the end users should be very less.

Scanned with CamScanner

Usability Checklist:

1. Every page should have "Home" link.
2. Every page should have "Logout" link
3. Every page should have "Help" Link.
4. Bread Crumb



5. Alt tags / Mouse Over : Whenever we keep the mouse in a particular this, it shows the description of the respective part.

408 Checklists.

NOTE:

Generally usability testing is always done by using some usability checklist, so that it becomes process oriented rather than person oriented.

Scanned with CamScanner

Process:

① Identify the needs of usability testing.

② ~~Deri~~ Derive the usability checklist.

③ Customer review & approval.

④ Execute the usability checklist on the application.

⑤ Prepare usability checklist testing report.

Usability Checklist Testing Report : (By BA)

Page Name	1	2	3	4	5
Login	N/A	N/A	P	N/A	P
Home	P	P	F	N/A	P
Sales					

→ For comment writing.

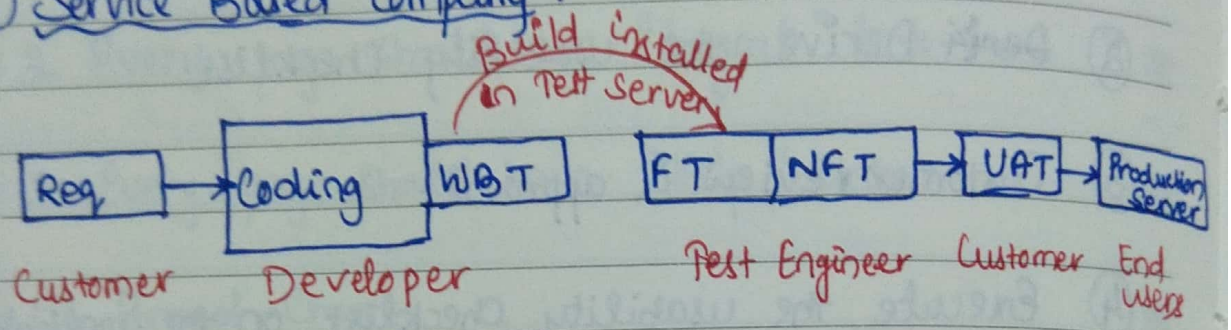
NOTE:

No sequence is maintained for the non functional testing, but it should be done after function testing.

Scanned with CamScanner

NOTE:

① Service Based Company:

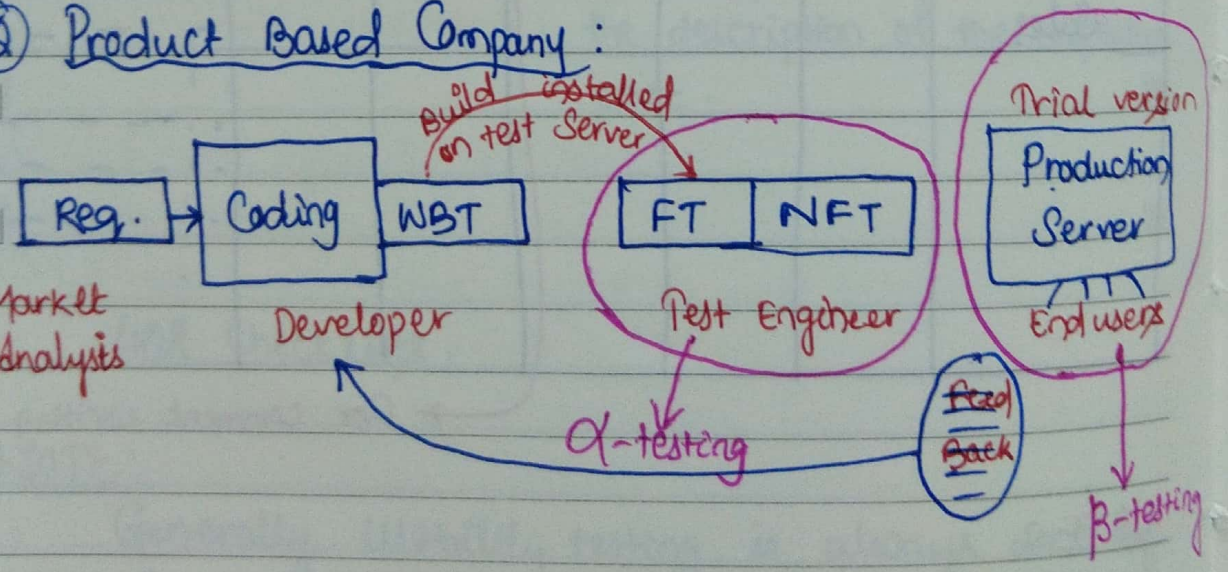


α-Testing:

→ It is a type of testing done by the test engineers at developers site (where the application was developed i.e. inside the company.

→ All functional & non-functional testing are consisted here.

② Product Based Company:



Scanned with CamScanner

β-testing:

It is a type of testing done by the end users at customer's site i.e. where the user uses the application.

NOTE:

Generally β-testing is similar to UAT done in service based companies.

Exploratory Testing:

→ Generally we go for exploratory testing whenever the requirement is not exist or it exist but not clear.

→ These cases we follow an approach i.e.

① Explore the application where using application in all the possible ways.

② Understand the flow of application based on exploration.

③ Prepare the necessary test document so that we don't forget the scenarios.

④ Start testing the application.

The above approach is known as exploratory testing.

Scanned with CamScanner

Disadvantages:

1. Time Consuming

2. Bug may misunderstood the requirement i.e. Eg: 1+2=4 which can't be logged as a bug.

3. Requirement may misunderstood the bug. i.e. 1+2=3 if not found as same then log a bug.

4. No maximum test Coverage.

Is exploratory testing a good approach?

As a professional test engineer we never suggest for exploratory testing but in the case if requirements are not clear or requirements are not there, these cases we may go for exploratory testing.

Who does exploratory testing or Who is a better person for exploratory testing?

Generally it should always be done by domain experts, since they've the knowledge on similar kind of applications.

Eg: Suppose a company dingdong pvt. ltd. has assigned with an application to test. There is one test engineer in that company who is newly joined.

Scanned with CamScanner

they've written roughly the requirements which is not clear. Developers develop the application. But while testing that application, the requirement is not clear. He should do one round of exploratory testing to explore the scenarios of that application.

Test Plan:

It is a document prepared by the leads & managers that consists of all the testing activities which needs to be followed by the test engineers for their future testing process.

NOTE:

Test plan is a dynamic document i.e. it is flexible for changes as per the change in requirement.

When it is Prepared?

This document is prepared even before the test engineers start their testing activity.

or,

As soon as the requirements are handed over to the leads & managers, they prepare this document before testing activities start.

Scanned with CamScanner

Attributes / Contents of Test Plan:

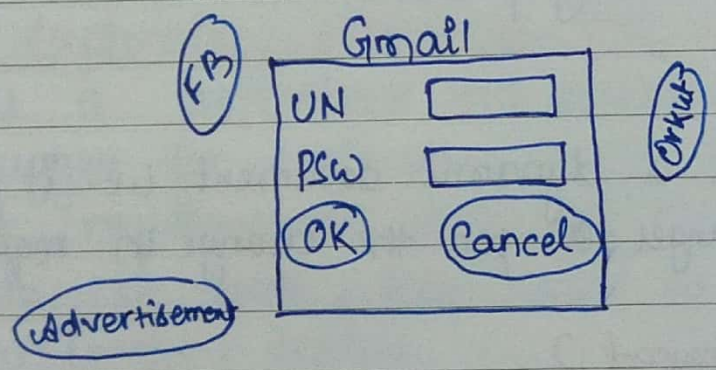
1. Objective

It defines the aim or goal of writing the test plan.

Eg: This plan is for Cheeta – 2.0 – icici
Application version customer

2. Scope:

Here we define the features which needs to be tested & the features that need not be tested.



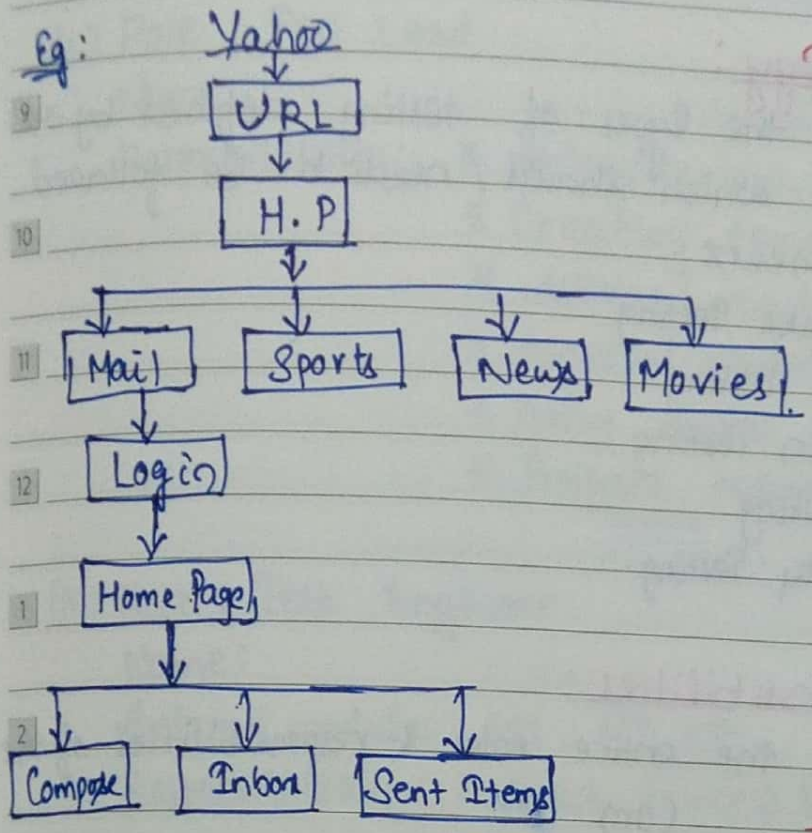
In Scope	Out of Scope
Login	FB
Compose	orkut
Inbox	Help → Technical writer
S.I	—
—	—
—	—
Need to be tested	Need Not to be tested

3. Approach

It defines the flow of the application defined in form of a flow chart.

Scanned with CamScanner

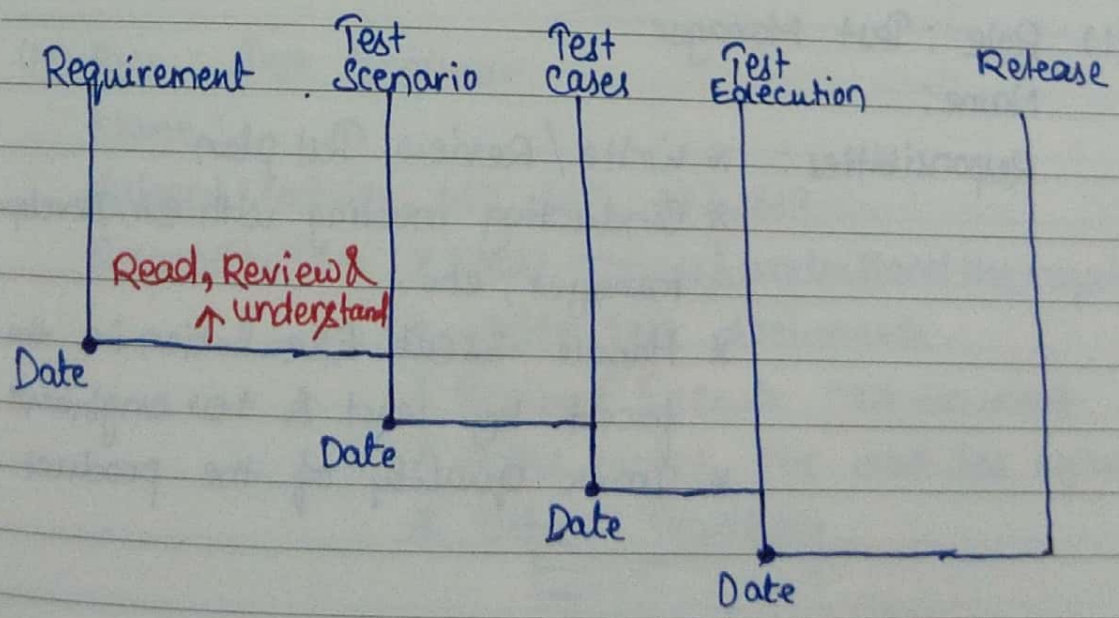
Eg:



Flow of the application in form of Flow Chart.

4. Schedule :

It defines the time duration for each & every testing activities.



Scanned with CamScanner

5. Test Methodology:

These are the types of testing defined by the lead or manager which should / needs to be followed by the test engineers.

i.e.
* Start Smoke Testing
* Functional Testing
* Integration Testing
* System Testing
* Compatibility Testing.

6. Roles & Responsibilities:

It defines the entire roles & responsibilities of the testing team. (or)

It defines the different designations & task that needs to be perform by different set of peoples in the testing team.

Eg:

(1) Role : Test Manager

Name:

Responsibities:
* Write / Review Test plan.
* Conducting meeting with BA, Development manager, etc.
* Handle ~~Escati~~ Escalation i.e. the issues faced by lead & test engineers.
* Track quality of the product.

Scanned with CamScanner

(II) Role : Test Lead

Name :

Responsibilities: * Write/Review Test plan.
* Creating templates
* Assigned modules
* Approval modules
* Daily Status meeting
* Prepare reports.

(III) Role : Test Engineer

Name:

Assigned modules : M1, M3, M4

Responsibilities : * Read, review & understand the requirement
* Write test document.
* Review / Execute test document.
* R.T.M w:r:t their modules
* Defect Tracking

(IV) Role : Test Engineer

Name :

Assigned Modules : M2, M5, M7, M9

Responsibilities : * Read, review & understand the requirement
* write test documents.
* Review / Execute test document
* RTM w:r:t the modules ~~as~~ assigned
* Defect Tracking

Scanned with CamScanner

7. Test Environment:

In this section they define the entire S/W & H/W configurations of the servers as well as the test engineers machine.

Eg:

(I) Main Server : QA Main1
Test URL : http://- - - - - - - - - -
S/W : OS, Web Server, App. Server, DBS
H/W : RAM, HDD, Processor.

(II) Back up Server: QA Main 2
Test URL :
S/W :
H/W :

(III) Staging Server:
Test URL :
S/W :
H/W :

(IV) Test Engineers machine
Base Platform : XP
S/W : OS, Browsers
H/W :

Scanned with CamScanner

8. Test Deliverables :

→ These are the list of documents that needs to be handed over to the customer along with the application.

→ It is generally the test o/ps which is prepared by the lead.

→ Here we give,

(1) Test Cases : If the customer gives his requirement to different companies in different releases
Eg: Release-1 (Wipro) & Release-2 (Infosys)

(2) Test Scripts

(3) R.T.M

(4) Test Execution Report i.e. pass, % pass, fail, % fail status

(5) List of bugs existing in current release.

(6) List of fixed bugs from the previous release

(7) User Manual, Installation process.

(8) List of platforms on which application is tested.

Scanned with CamScanner

28.05.14

9. Assumption:

These are the possible problems that may occur during testing process. i.e.

(I) If TE #1 left

(II) If main server is down

10. Risks:

Here we define all the risk encountered due to the assumptions. i.e. from step-9

(I) Quality is affected.

(II) Test Engineers sit idle.

11. Mitigation / Contigency / Back up plans:

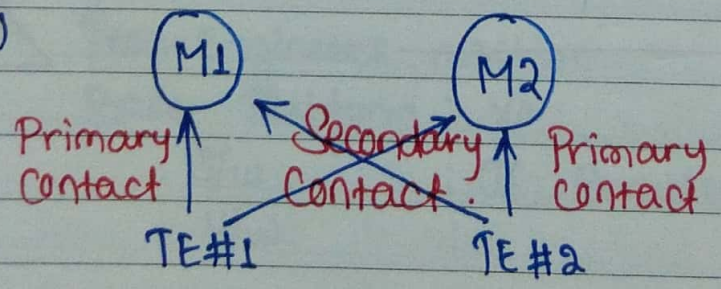
It defines the back up plans for the assumed risk during the testing process. i.e. from step-9

(I) modules should have secondary contact

(II) back up server should be there.

In brief for 9, 10 & 11.

(I)



* TE #1 left (9)
* Quality of M1 affected (10)
* Modules (II)
  - Primary Contact
  - Secondary Contact.

Scanned with CamScanner

(II) * Main server down (9)
* TE sit idle (10)
* Back up server. (11)

12. Tools:

The tools used for testing process are defined under this section.
(or), Simply the S/ws used to test an application.

(I) Functional Testing Tool
Name : QTP (Quick Test Professionals) (By HP)
Version : 11.0
Licenses : 4 (i.e. used in 4 Systems)

(II) Performance Testing Tool
Name : Load Runner (By HP)
Version : 11.0
Licenses : V users (Let 1000 Users)
It is used to check the response time of the application for the end user's request.

(III) Test Management Tool :
Name : Quality Center (QC) (By HP)
Version : 11.0
Used for : * Requirement
* Write/Execute Test Cases.
* Defect Tracking
* Report Generation.

Scanned with CamScanner

13. Entry & Exit Criteria:

→ It defines the conditions that should be satisfied before starting the testing & stopping the testing.

→ The entry criteria would be,

* Coding should be completed.
* Build should be installed on to the test server.
* Necessary test documents should be prepared.
* Finally, resources should be ready.

→ Similarly the exit criteria for testing would be,

** All the written test cases should be executed.
** Max % of the test cases should be pass.
** It also depends on severity of the bugs like there should be no blocker, critical or major bugs where as some minor bugs may exist in the application.

	F.T	I.T	S.T
** % of T.Cs Executed	90%	95	100
** % of TCs pass	85	90	96
** Based on Severity			
Blocker	0	0	0
critical	3-4	1-2	0
Major	6-8	3-4	0
Minor	8-10	5-6	3-4
	EXIT F.T / Entry I.T	Exit I.T / Entry S.T	Exit S.T & Entry for Release

Entry Criteria

* Coding should be done
* B.I.T.S
* T.D
* Resources

According to Test Plan 96%
According to Test Execution Report 98%
↓
Can be released

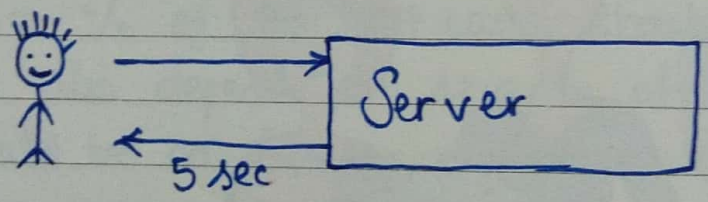
Performance Testing:

→ Checking the behaviour of an application by applying the load ~~to~~ is known as performance testing.

→ In performance testing we check for certain factors like
(I) Response time
(II) Load
(III) Stability.

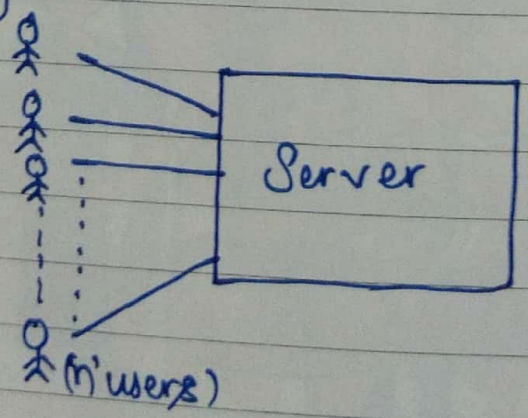
(I) Response Time:

It is the time taken by the server to respond to the client's request.



(II) Load:

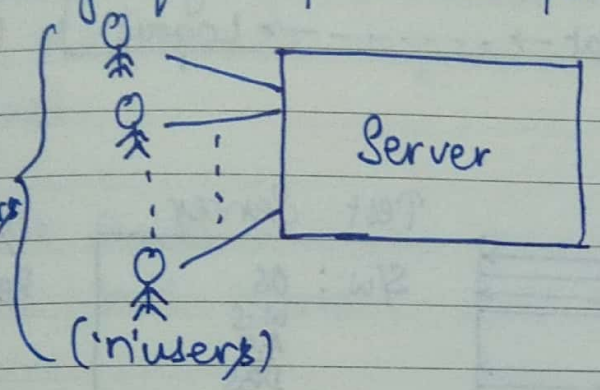
No. of peoples or users accessing the application simultaneously.



Scanned with CamScanner

(III) Stability:

It defines the no. of users using the application continuously for a particular period of time.



When?

We go for performance testing once the application is functionally stable.

Why?

Once the application is released into the market it may be accessed by 'n' no. of users simultaneously & during this process, they may find some performance issues. So to avoid this, we do performance testing.

NOTE:

Performance testing should not be done manually because it's costlier approach, time consuming as well as accuracy can't be maintained. So it's done by some tools.

Eg: for Tools:

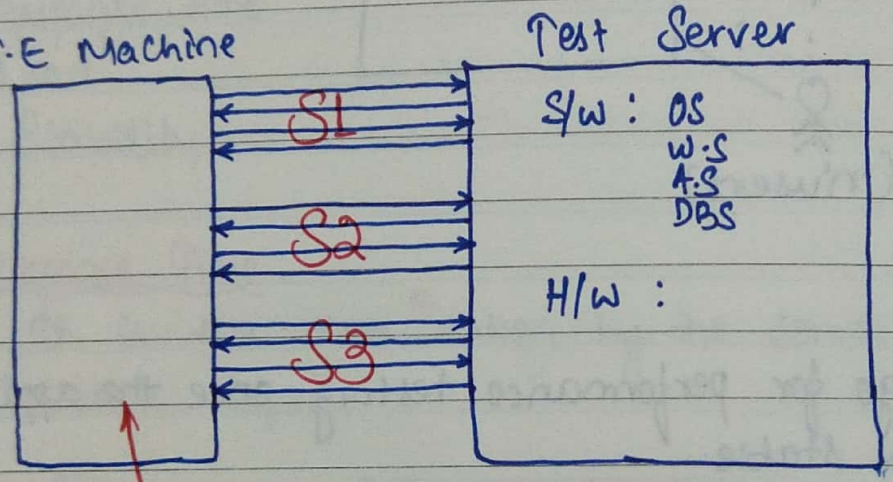
Load Runner, Jmeter, Neo-load, web-load, E-tool, web stress tool, ...

Scanned with CamScanner

Process:

S1 : Login → Compose → Logout
S2 : Login → IB → Download → Logout
S3 : Login → Chat → ..... → Logout.

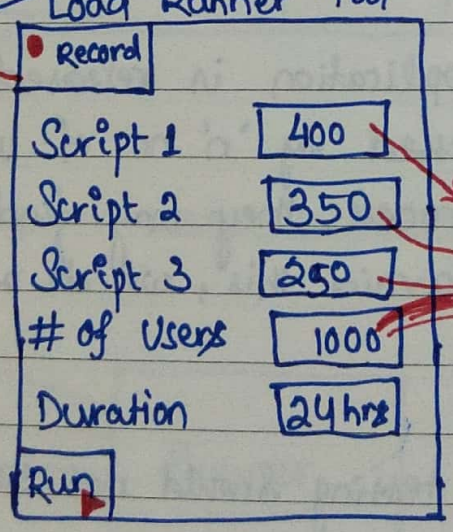
→ Should be in MS Word Format.



Requirement
Users : 1000
Res. Time : 5sec.

Install

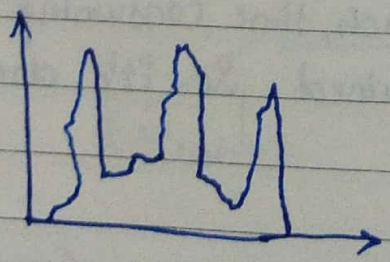
Virtual client



400 + 350 + 250 = 1000

Vusers i.e. Virtual Users

04 SUNDAY



Avg. Response Time = 4.7 Sec
Max. Response Time = 7 Sec.
Min. Response time = 3.8 Sec.

Scanned with CamScanner

NOTE:

In the previous process, the response time calculated by the test engineers meets (i.e. equal or less than) with the client's requirement. So we can release the product.

We should have the knowledge on 'C' language for the load runner.

Process:

1. Identify the performance testing scenarios according to,
   a. Most frequently used
   b. Most Critical Features
   c. Huge Data transactions.

2. Create Scripts → Scenarios with the help of tools.

3. Distribute the loads according to the usage pattern
   Eg: Gmail, Compose, Inbox, chat

4. Execute the scripts.

5. Get the result in the form of a graph.

Scanned with CamScanner

NOTE:

In performance testing we can't reproduce the bug.

Bottle Neck:

S/w: OS
WS | AS | DBS

H/w: RAM
HDD
Processor

Avg. R.T = 6.3 sec.
Needed = 5 sec.
Result Not meets to requirement.

Steps:

1. Analyze results : It's done by a set of persons like Architect, BA, Dev. Manager, Testing Manager, DBAdmin, N/w Admin

2. Identify the "bottle neck" (performance issue).

3. May present in
   * Code
   * DataBase
   * S/w i.e. web Server, App Server, database Server
   * H/w
   * N/w

4. Tunning : Adjusting the above things from step-3.

Scanned with CamScanner

5. Re-execute the script (because already script is there)

6. The above process continues till the result meets the requirement.

NOTE:

Generally, we do performance testing on the applications which are accessed by multiple users simultaneously.

Types:

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

There are two more types, ~~are~~ :

* Volume Testing
* Soak Testing / Endurance Testing.

Load Testing:

Checking the behaviour of an application by applying the load less than or equal to the desired load is known as load testing.

Eg: 1000 Users with 5 sec

Desired Load Goal

Scanned with CamScanner

Stress Testing:

Here we are checking the application by putting the load greater than the desired load.

Scalability Testing:

The process of checking the performance of an application by increasing or decreasing the load in a particular interval is known as scalability testing.

No. of Users	Avg. R.T in sec
1000	4.5 sec
1100	6
1200	6.5
1300	8
1400	8.5
1500	Crash

No. of users increased

Upward Scalability Testing

No. of Users	Avg. R.T in sec.
1000	7 sec
900	6.8
800	5.6
700	4.9

No. of users reduced

Downward Scalability Testing

Stability Testing:

Here we start checking the performance by applying the desired load for a particular duration of time.

Scanned with CamScanner

Volume Testing:

Here we check the application behaviour by applying huge volume of data transactions.

Soak / Endurance Testing:

Checking the behaviour of the application by applying the load for a long duration of time in an environment which is un-supportive is known as soak or endurance testing.

Min. Set up [Application]	Unsupportive
S/w	
H/w	
10 lakhs	Execute Test script [24 Hrs]
1000 users	3-4 Years
5 sec.	5 sec.

Scanned with CamScanner

Miscellaneous:

1. Draw Back of RTM:

* All the test case name can't be covered for the requirement.
* Only one testcase can be covered.

(or)

* RTM only ensures that atleast one testcase is covered for every requirement. It doesn't ~~gutrantee~~ guarantee that all the test cases are covered for single requirement.

2. What is defect clustering?

Defect ~~clusting~~ clustering states are the small number of modules which contains most of the defects detected.

3. What is hot fix?

It is the very immediate fix of the high priority bug (say within 7-8 hrs) if the release is on the next day or if it is found by client.

4. What is parallel testing?

It is a testing technique in which the same ~~typ~~ i/ps are entered in two different versions of the application & reporting the anomalies.

Scanned with CamScanner

Characteristics,

* Ensures that the new version of the application performs correctly.
* Ensures consistencies & inconsistencies are the same between old & the new version
* Ensures the integrity of the new application.
* Verifies the data format between the two versions have changed or not.

5. What is latent & golden bug?

The bug that is not identified in the past versions of the software application is known as Latent Bug.

The bug that is occured in every instance of the application with high severity & high priority.

Scanned with CamScanner