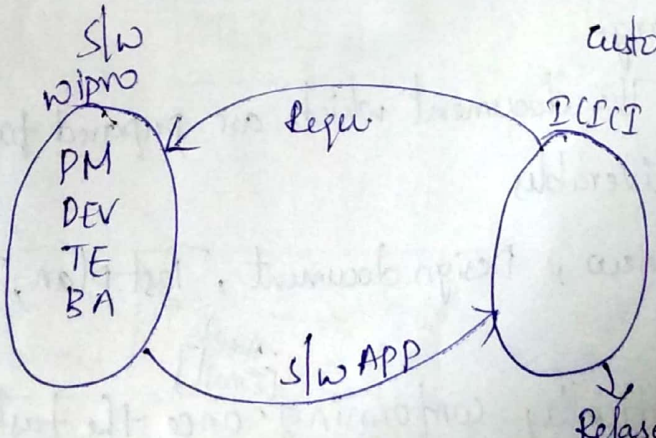
Terminologies:

- **Domain** - It is catagorisation of application into diff fields

  Eg:- BFSI - Banking financial service Insurance
  - Ecommerce - Filpcard, Snapdeal etc.
  - Retail - wealth fargo (vegtable, etc.)
  - Entertainment - EA (Electronic Art)
    ↓
    TV, Games, Adds Related things
  - Telecom - Telephone Networking.
    Eg - TCS, Subex.
  - Education
  - Travelling
  - Healthcare ≡ Ge, Smids

- **REQ** - Requirments

Stakeholders - People who involved in a project directly or indirectly
(or)
People who are directly or indirectly involved in a software project



→ customess and project team directly involved.

→ End users are indirectly involved

- RTM - Requirements Traceability Matrix
- KT - Knowledge Transfer
- Onsite - Customer place.
- Offshore - Developer place.
- QC - Quality control - process oriented
- QA - Quality Assurance - product oriented

⇒ If we follow QA [i.e if we follow process, we get high quality product.

- DL - Distribution List

⇒ It is an email ID which consist of multiple email ID's

Eg:-

Dev+DevL → dev.abc@xyz.com — (11)

T-E+TL → Test.abc@xyz.com — (21) +

Dev+TE → abc@xyz.com — (33)

- Review - It is a process of checking or identifying mistakes in the document which is prepared for project purpose

eg:- Code Review.
Test case

- Giant :- Big/large
- Deliverables :- The document which are prepared for project purpose are deliverables

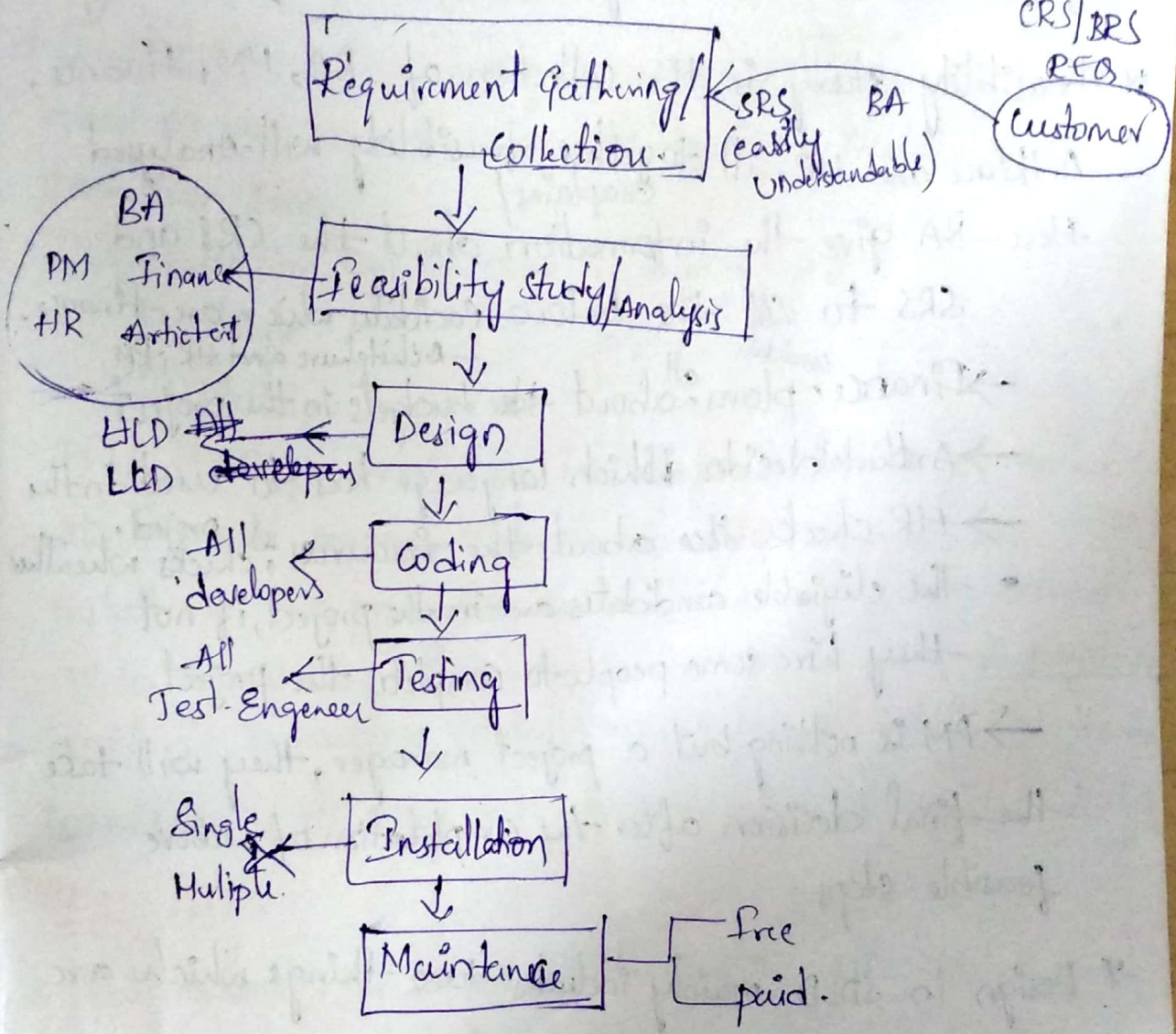
eg:- Code Review, Design document, Test plan, Test case RTM.

- Signoff :- Officially conforming [through email] once the task has been completed from the respective team.

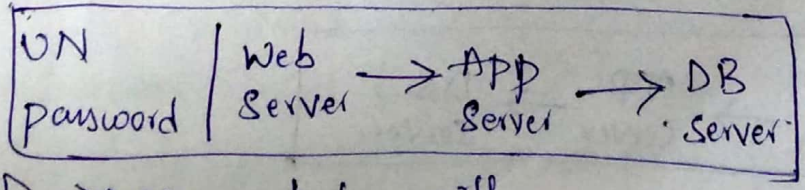
SDLC - Software Development Life cycle.

SDLC - It is a procedure to develop the software application.

SDLC



HLD



LLD → programs to be written

login → Create session

↓

Record login time

↓

Send Response

* The conversation between CRS and SRS can be easily converted by B.A. It can transfer the information through BA to SRS easily understandable.

* This is the work done by the Requirement Gathering.

* Feasibility study is the collection of BA, PM, finance, Architecture and HR. all together feasibility will Analysed.

Here - BA give the information (explains) about the CRS and SRS to diff the below candidate which are finance, Architecture and HR, PM.

→ Finance - candidate plans all about budgets in the project

→ Architect decides which language has to used in the project.

→ HR checks about the Resources is project; checks whether

- the eligible candidates are in the project, if not they hire some people to complete the project.

→ PM is nothing but a project manager, they will take the final decision after the completation of above feasible steps.

* Design in SDLD mainly includes two things which are HLD and LLD.

Example for HLD : designs the blue print.

* Coding will done by the developers. They develop the code according to Requirements.

* Testing is done by Test Engineers. Here the code which is developed by the developers are tested by test Engineers.

* Installation :- Once testing is completed (application) it has to be release to end users. So, we need to do a installation to production server.

⇒ Installation can be done in two ways
  i) Single
  ii) Multiple.

i) If it single installation, installation team is not needed. either (or) project manager (or) develope (or) Test Engineer can install to the production server (This may take around 15 to 30 minutes Eg:- Application which are from Playstore

ii) If it is multiple installation, a separate intallation team is needed. (It may take more duration & repeated activity)

Eg:- ATM.

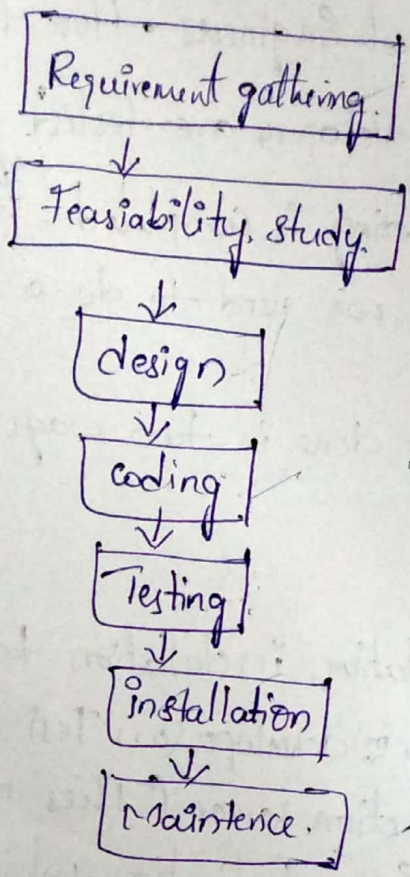
* Maintenance :- The support which is given after product is installed to the production server.

It can be a) Free
  b) Paid.

In maintenance: defect fixes will happen changes will be handled (add, modify, Remove, of a feature.

Models of SDLC

i) Waterfall Model



4/5/18

→ It is one of the oldest model in SDLC.

→ Developer used to do testing, currently Test Engineers are doing testing.

→ When do yo go for waterfall model?

i) When Requirements are ~~freezed~~ freased.

ii) for shoiterm projects

iii) To develop a small application

Advantages of waterfall model :-

i) We get stable product if requirement does not change

ii) Initial investment is less (Because only developer are hire & Test Engineers are hire at later stage)

Disadvantages:—

i) Req, Design are not tested.
ii) Testing is happening only after coding.
iii) If Req changes there will be lot of Rework.
iv) Since we are not testing REQ, Design; We may find more defects while testing & that will lead to lot of Rework again

Eg:—
→ 3G, 4G
→ Nokia
→ Kingfisher

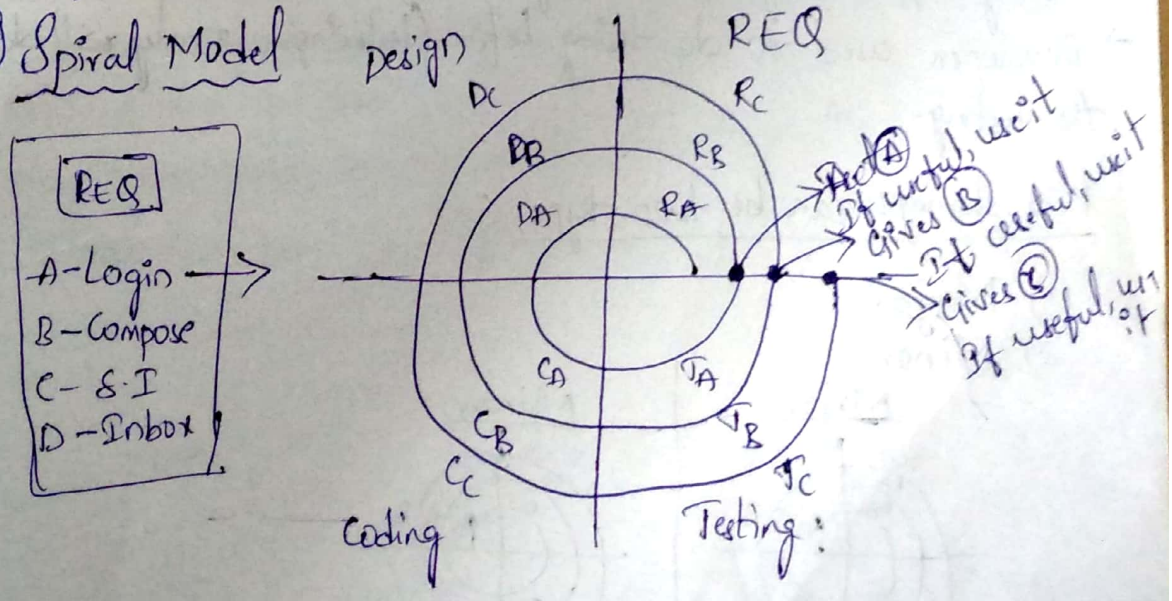
⇒ Why Requirement changes?

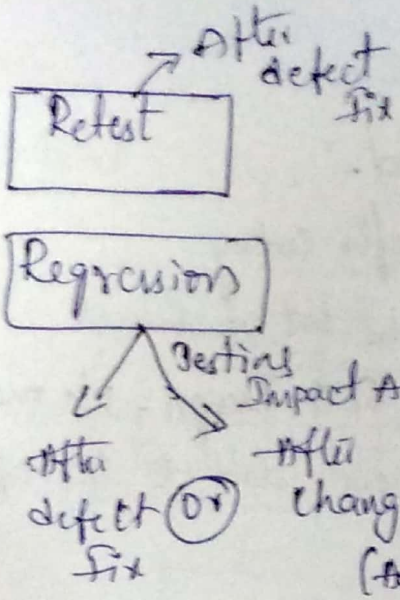
Because of the competition, because of change in technology, Because of change in business

⇒ Why developers should not be involved in testing?

i) They are over confidence
ii) They think what ever they developed are perfect
iii) They always want to see the product building not product breaking
iv) Even though they know defects are there they will hide it
v) They will not accept there mistakes

② Spiral Model





find defect - TE
Raise/log/Report - TE
Fix defect - Dev

* When do we go for spiral model?

⇒ When ever REQ are given stage by stage or part by part when ever there is lot of dependency, between the modules

Advantages of Spiral model

⇒ REQ changes can be are allowed

⇒ Customer can Receive part of application & which can release in to market if it is feasible to use by end users.

Disadvantages:

⇒ REQ and design are not tested.

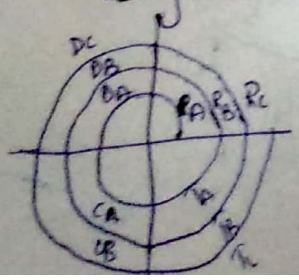
⇒ Testing happens only after coding currently.

⇒ Developers used to do testing before. Tester Engineers only will do the testing.

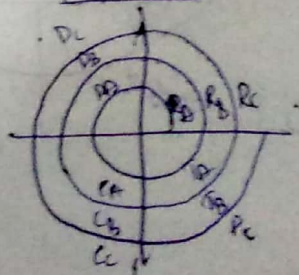
REQ changes can be two types:

1) Major
2) Minor

Major



Minor



Whenever their is a major ~~change~~, we can work on the ~~chang~~ current not on the new Req	whenever their is a minor change, we can work on the changes on new REQ & defect fixes

③ V-Model (or) V and V model :—

Q) When do we go for 'V'-Model?

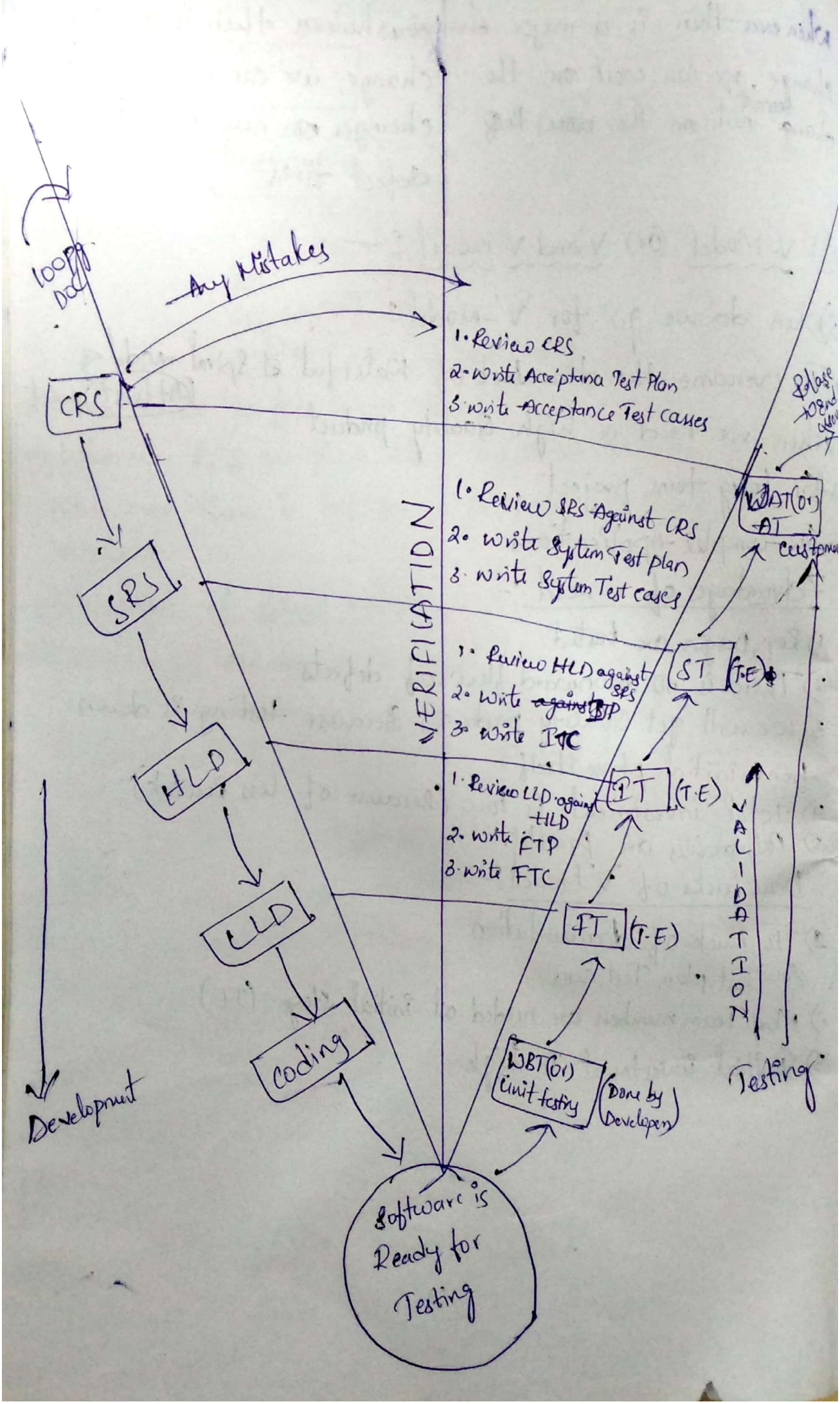
A) → To overcome the drawback of waterfal & Spiral model (At Least point)
→ when we need a high Quality product.
→ for long term project
→ for complex Applications

Advantages of V-model :-

1) Req, Design are tested
2) There is no downward flow of defects
3) we will get Quality product, Because testing is down from initial phase itself
4) Total investment is low (Because of less Rework)
5) Delivarables are parallel.

Drawbacks of 'V' Model :-

1) To much of documentation
   Eg:- Test plan, Test cases
2) More Team members are needed at initial stage (T-E)
3) Initial investment is high.



Proto Type Model :-

*When do we go for prototype Model?

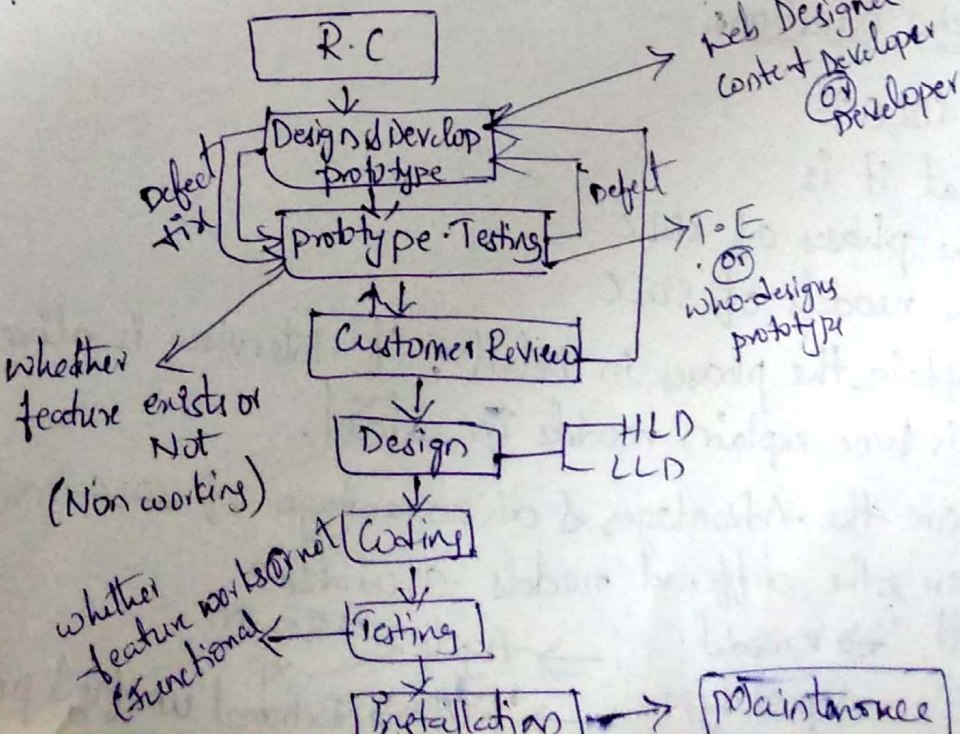
→ when customer is not clear about his REQ.
→ when software company is new to the domain.
→ when developers are new to the technology.

Advantages of prototype :

→ There is improved communication b/w customer & software company.
→ In the begining itself customer knows, what he gets on last day.
→ In the begining itself developer will know what he know will develop on last day
→ If customer wants any changes he gets a chance to ask it in the begining itself

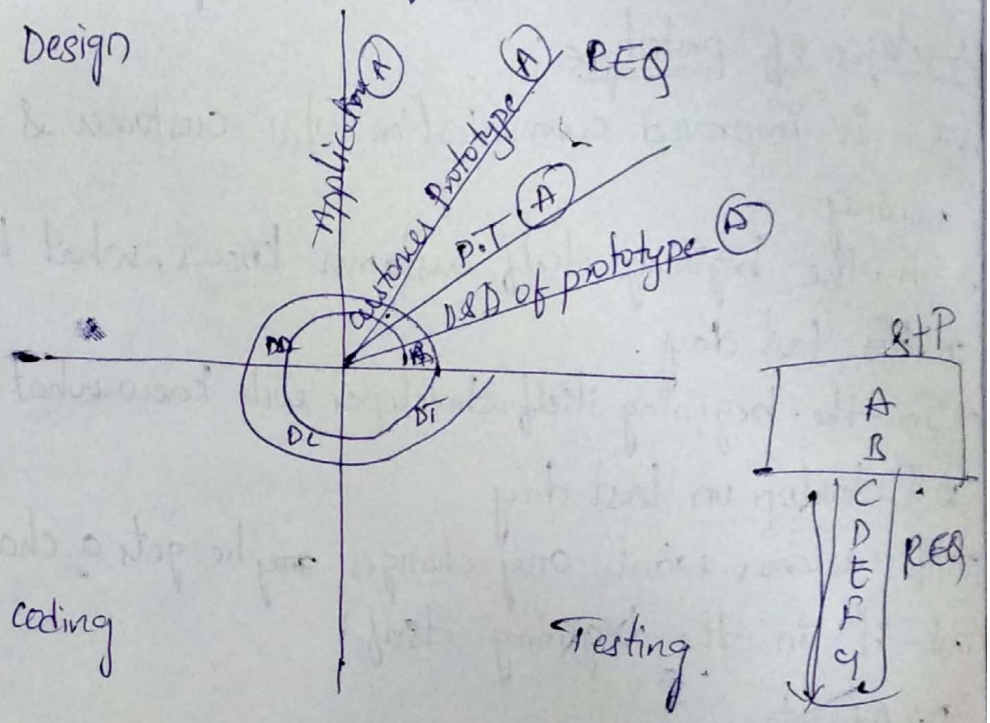
Dis-Advantages

→ Investment is done on the prototype.
→ There will be delay in actual start of the project.
→ Too many changes can distrib the rhythm of development team.



Note:- Once I get a job, I may be using any of these models, combination of these models (Hybrid model) (or) deriving a new from these existing models (Derived model) (or) I will not know which model I am following.

~~Combination~~ Combination of Spiral + Prototype



* When do we go for this model?
→ When the requirments are step by step process and ~~is~~ when the requirments are not clear.

Interview Questions:-

* Explain SDLC?
→ Tell what it is
→ Tell the phases of SDLC
→ Tell the models of SDLC
→ Now Explain the phases in Detail, if interview is alive continue, if there is time explain models in detail.
* What are the Advantages & disadvantages of ~~SDLC~~ models?
* What are the different models available?
→ waterfall → v model → Agile ← Scrum, XP
→ Spiral → prototype → RUP → Rational unified process

RAD → Rapid application development

W Model.

Project Specific Questions :-

1) which module you followed in the project?

2) what is your roles and responsibilities in your project

→ Explain with respect to module wise and use the word "we" while explaining.

Tools used in the industry :-

1) Automation tools

2) Functional Testing tools.

i) Selenium. → free of cost (Open source) → uses Java, phython, ROR..

ii) QTP → Quick Test proffession / UFT → unified functional Testing
(licensed) ↓ older version ↓ latest version
→ uses VBS (Visual basic scripting)

iii) Win Runner
(licensed) ↓ older ↓ older gen

iv) Test Patner

v) Silk Test.

3) Performance Testing Tool : → Response time

i) Load Runner } Licensed tool
ii) J Meter } Hp.
iii) Silk load.
iv) Neo load
v) QA load.

4) Test managment Tools :
licensed (HP)

i) QC → Quality center / ALM → Application life cycle Managment

ii) JIRA → best tool for Agile model.

iii) Test link.

iv) OTM → Oracle test managment

5) Defect/bug Tracking Tools :-

i) Bugzilla

ii) QC/ALM

iii) JIRA

Software Testing :— Testing functionality of an applic-ation with respect to given requirement by customer is called soft ware testing

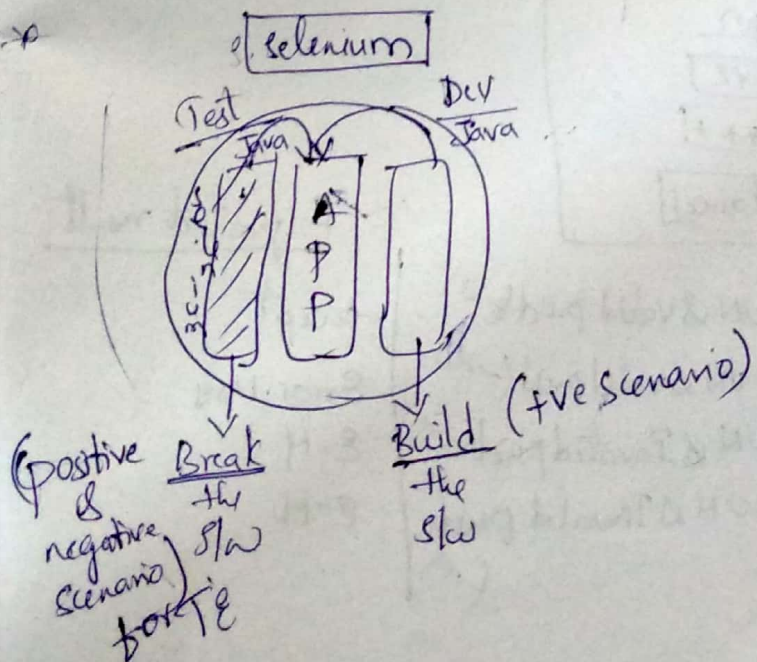
Manual Testing :— Testing the functionality in application wrt given REQ without using any tool is called Manual tesing

Automation testing :— Testing the functionality of an application wrt given requirment by using a tool.

eg:- Selinium.

*Whay testing is important?

Every application is developed for business. If there are defects in the application after releasing to the end user, it will affect the no. of users who are using that application and negative word will spread across. To avoid all these thing, Software testing must be needed. If it is not done it will effect the business hugley.



* Drawback of Software's Manual Test :- It consumes lot of time.

→ It is monotonous / repeatated in nature (because of Regression testing)

→ Turn around time is more

∴ To overcome the drawback of manual testing we go for Automation ~~testing~~.

eg:- Selenium.

*** Senario :- Testing the application in all the possiable ways.

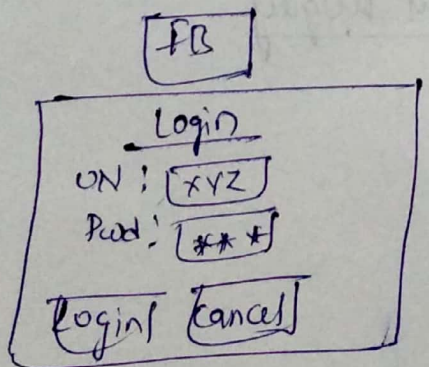
→ Senario is of two types :- positive senario
Negative senario

* Senario is not a defect

* Defect is not a senario

∴ To find defects, we have to identify senario

→ Eg:-

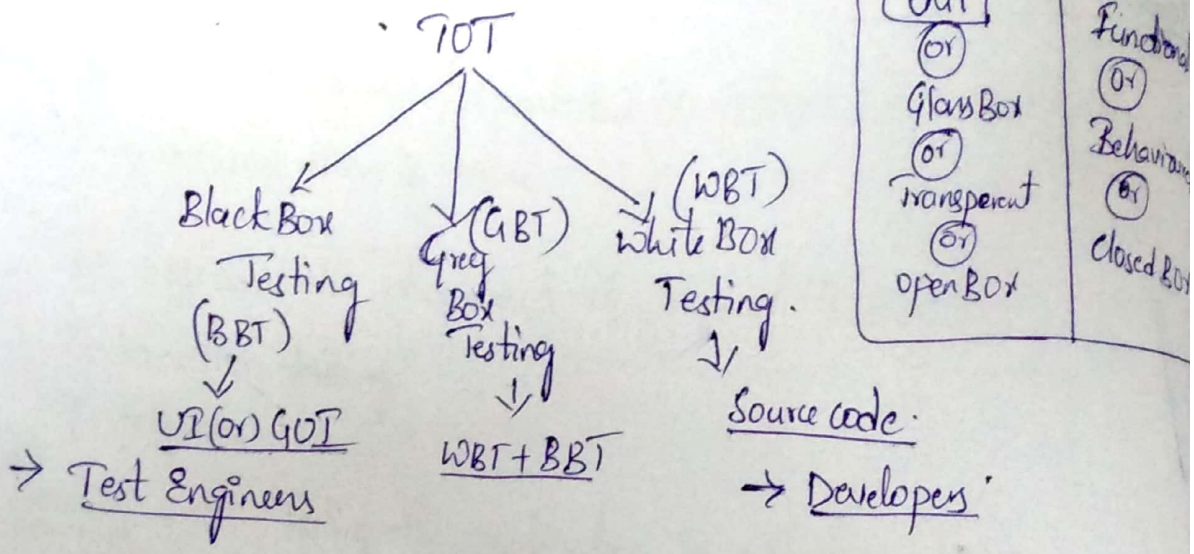


Scenario		Expected result
1 :- valid UN & valid pwd	(+ve)	Accept
2 :- Invalid UN & valid pwd	(-ve)	Error Msg
3 :- valid UN & Invalid pwd	(-ve)	E.M
4 :- Invalid UN & Invalid pwd,	(+ve)	E.M.

Note:- There need not be negative scenario in all cases.

Scenario for Escalator.

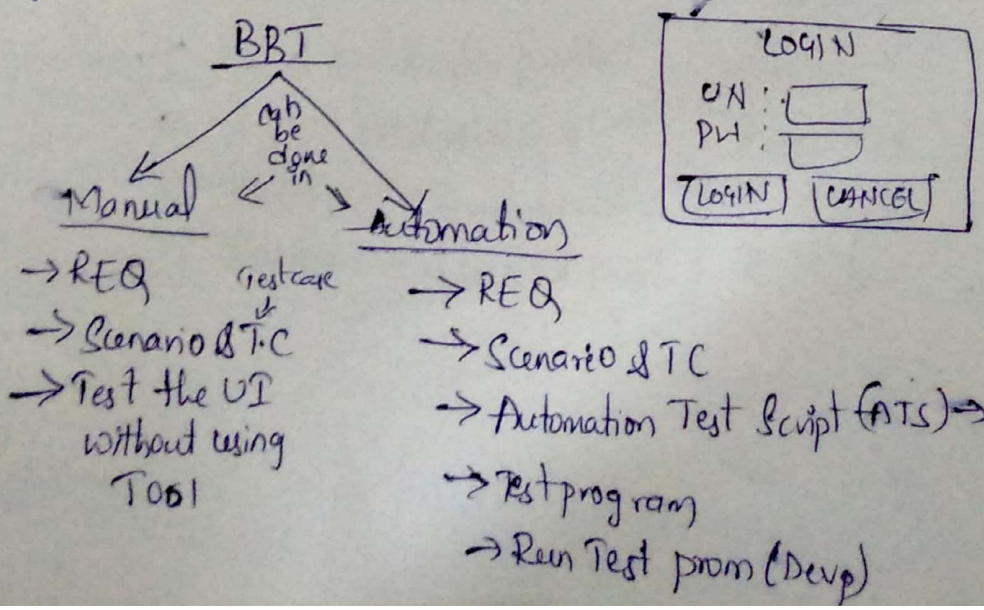
Types of Testing :-



* Black Box Testing :
→ Testing the UI or GUI of an application is BBT
→ It is usally done by test Engineers

* White Box Testing :
→ Testing the source code of an application is WBT
→ It is usally done by developers.

* Grey Box Testing :
→ Testing the UI and source code is GBT
→ Those who have knowledge on UI and source code they will do this GBT.



WBT

(ab) be done

Manual

→ REQ
→ Design & code
→ Test each & every LOC manually

Automation

→ REQ
→ Design & code (pgm for Development)
→ write prgm for Testing (JUNIT / NUNIT)
→ Run the progms written for Testing.

JUNIT :- The code which is written in Java code.
↓
Automation tool.

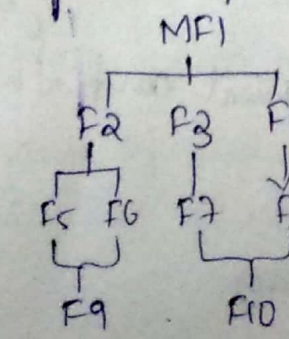
Eg:-

REQ	Prg (Development)	Test prgm JUNIT	Run (1)	Run (2)	Run (3)
A	(A) 100	TestA	pass	pass (Reg)	P (Reg)
B	(B) 100	TestB	fail	pass (Retest)	P (Reg)
C	(C) 100	TestC	Pass	fail (Reg)	P (Retest)
D	(D) 100	TestD	Pass	pass (Reg)	P (Reg)
E	(E) 100	TestE	Pass	Pass (Reg)	P (Reg)

Types of white Box Testing

1) Path Testing : Identifing all the independent paths and testing those paths

Eg:-



Paths

- F2 - F5 - F10
- F2 - F6 - F9
- F3 - F7 - F10
- F4 - F8 - F10

→ This will done done by developers by sometimes they will not do properly.

⇒ Developers may test few paths and they may not test few paths

⇒ This becomes advantage for testing team to find d

2) Condition Testing :

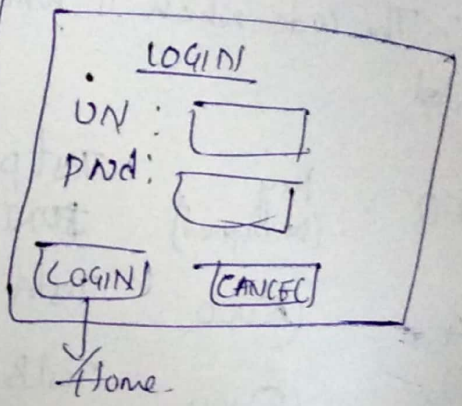
⇒ Testing for all True and false conditions i.e if else condition is Condition Testing.

Eg:

Source code

```
if (valid)
{
  sucess;
}
else
{
  sucess
}
```

UI



⇒ Developer ~~Developer~~ may test and may miss to test else

⇒ This is ~~what~~ becomes advantage for test team to find d

3) Loop Testing :

Testing for all the loops before break statement and after break statement is loop Testing.

```
while(10)
{
  Break;
}
```

Eg: Mobile passwords like ↓

⇒ Pwd, pin, finger print, pattern .. etc.

⇒ Developers may test till break statement and they miss to test after Break statement

⇒ This becomes advantage for testing teams

White Box Testing from Memory Point of View :-

⇒ What are the typical mistakes done by developer because of which memory is consumed.

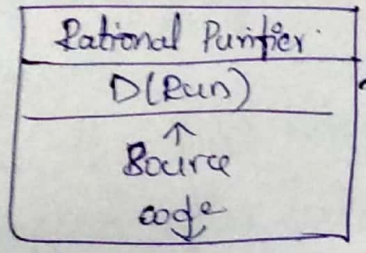
Ans (i) Because of not using inbuild function

eg:- sort() → 25MB → Inbuild function
Mysort → 256MB

(ii) Because of unused variables and functions

eg:- int i=10; → used
int j=20;
String s="Hello" } not used → waste of memory

Because of
Rational Purifier : Tool



⇒ It is a tool to verify the unused variables and function are used are not.

(iii) Because of logical mistakes.

White box testing from Performance Point of View :-

⇒ What are the mistakes done by developer:
Because of which performance is reduced?

Ans i) Because of logical Mistakes.
ii) Because of not using 'switch case' instead of 'if else'.

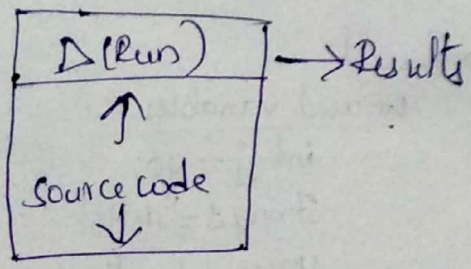
iii) If we are using 'OR' between two conditions, use that condition in the begining which majority of times becomes true

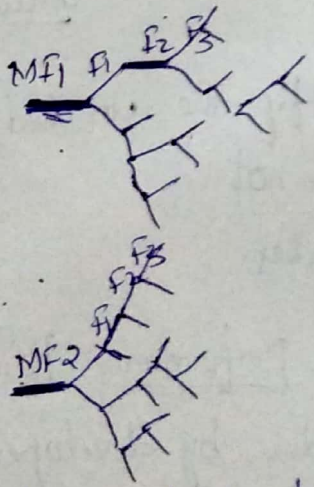
eg:- if (Cond1 OR Cond2) → Increases Execution Spead
     if (cond2 OR cond1) → Decreases Execution Spead

iv) If we are using 'AND' between two conditions, place that condition in the begining which majority of the times becomes false

Eg:- if (cond1 AND cond2) Increases Execution Spead
     if (cond1 AND cond2) decreases Execution Spead

→ Rational quantifier :-





Thick line indicates, that function is taking more time to execute

Rational quatifier is a tool which is used to identify which part of the program is taking more

→ Performance Tuning :-

In an Application if some portion is not working the developers has to identify it & make the relevant changes so that the application performes well.

→ Difference between blackbox & white box testing

white box testing :-

1) Testing the source code of application

2) Usually, done by developers Rarely test Engineers

3) Need to have programming knowledge

4) This is the first which should happen after coding

I/P → O/P

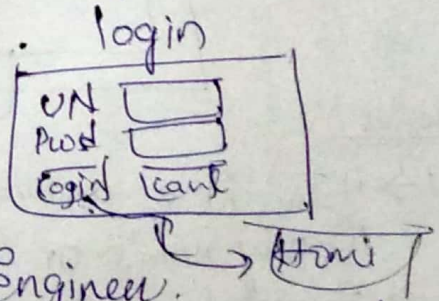
Blackbox testing :-

1) Testing the UI/GUI of an application

2) Usually done by Test Engineers

3) No need of programming knowledge

4) This is the testing which has to done after white box testing



21/5/18

Characteristics of Good Test Engineer.

→ Test Engineer will always like to break the product

→ Test Engineer should think in all different purspective (different Think)

Eg:- Think from customer point of view, Developer, Enduser, Buriness ...etc point of view.

→ If a Test engineer know Programing skills ; it will help to identify more scenario

→ Test Engineer should have good domain knowledge

→ Patience is very important.

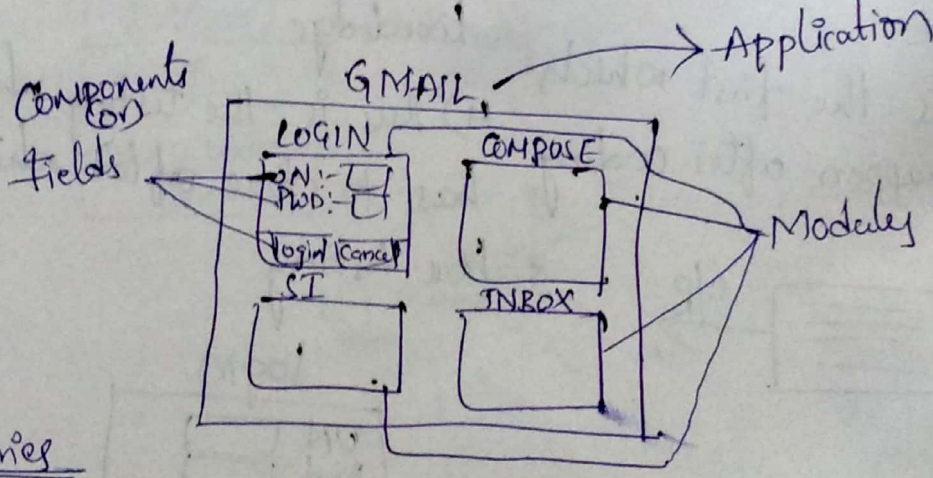
Black Box Testing :—

Functional Testing (or) Field level Testing (or) Component Testing

⇒ Testing the functionality of an application independently and thoroughly with respect to given requirments is called FT.

⇒ That is (i.e) Testing each component separately with all positive and negative scenario's.

eg.



Queries

1.1 → i) Is it a combination of Alphabets or num or both
ii) only num's or only Alphabets / uppercase or lower case

1.2 → Encrypted or not

1.3 → country code

1.5 → what all the options should be there

1.6 → what are the languages should be displayed

1.7 → Other options should be (or) not

1.8 → which formate (like calander)

1.10 → What are the options should be mandatory.

2/5/18

CRS / BRS

SRS

1.0 There should be a option to Add New user

1.1 User Name Text Field: It should accept 6-32 characters (Alphabets & Numbers)

1.2 Password Text Field: It should accept 5-10 characters (Any)

1.3 Mobile Number: It should accept 10 digits numbers

1.4 Email ID: It should accept valid Email ID

1.5 Designation: It is Drop down field

1.6 Language: It should have 3 options

1.7 Gender:

1.8 Date of Birth: It should accept valid Date

1.9 Address Text Area: It should accept maximum 2000 characters (Any)

1.10 SUBMIT Button: It should Add user

1.11 CANCEL Button: It should go to Home page

REQ Gap Analysis Document

SrNo	REQ No	REQ	Query
1	1.1		1/2
2	1.2		1.
3	1.3		1.
4	1.5		1.
5	1.6		1.
6	1.7		
7	1.8		1.
8	1.10		1.

Add user

Add user

*User Name:

*Password:

*Mobile Number:

*Email ID:

*Designation:

Language known: ☐ ☐ ☐

*Gender: ○ ○

Date of Birth:

Address:

SUBMIT CANCEL

→ Once the Customer gives the Requirements to the BA. BA will go through Requirements and then BA will share this requirements for Developer & Test Engineers. They will also go through through the Requirements and ask the ~~doubt~~ doubts as quires to the BA. Then BA will inform this to the Customer clarify those doubts. Then developers and Test Engineers will do the projects by analysis +ve and -ve Scenario's format

Scenarios for UserName Text field :-

Requirements :-
- 6-32 (A,N)
- Alphabets < upper / lower
- S.C (+, -, /, %, space)

Valid (+)	Invalid (-)
Should Accept the Data	Should get Error Message
• ABCDEF	• ABCD
• abcdef	• abcde
• 123456	• 12345
• AaBbB2	• Aa126
• ABCDEF..... 32	• *BA
• abcdef..... 32	• @+-
• 12345678910..... 32	• ABC..... 33
• Aa12BbCc6..... 32	• abc..... 33
	• 123..... 33
	• AB1..... 33
	• &**..... 33
	• Blank (donot enter anything)
	• abB..... 33
	• ABCD** → 6 but not accept
	• ~~abcd~~ *9II*?#! => "

26/05/18

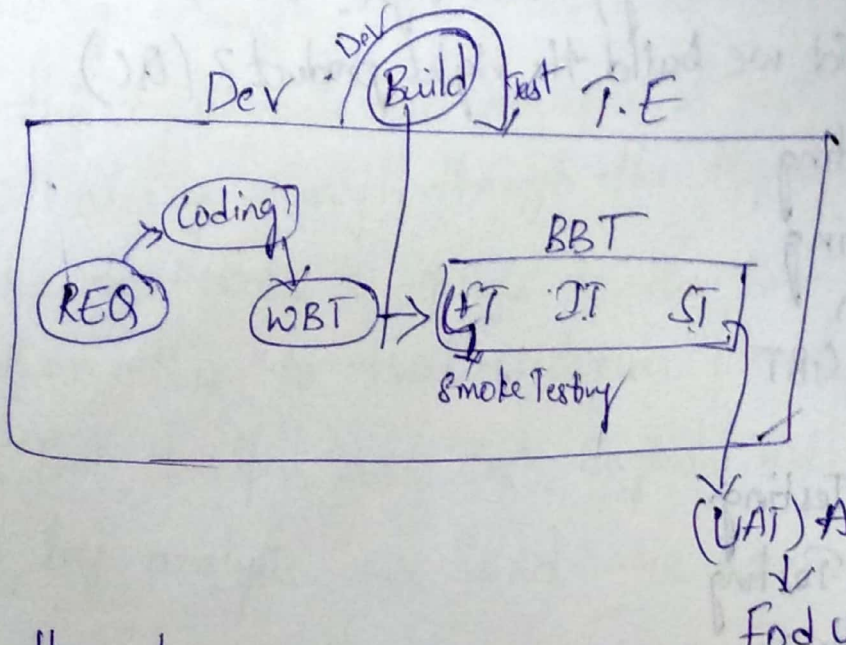
Grooming class

Verification - are we building product right? (QA)

Validation - Did we build the right product? (QC)

- Software Testing
- Manual Testing
- Automation
- WBT, BBT, GBT
- Scenario
- Functional Testing
- Integration Testing
- System Testing
- Acceptance Testing
- Smoke Testing & sanity Testing
- Adhoc
- Exploratory
- Compatibility
- ~~Compatibilit~~ Performance (& calability)
  - Load
  - Stress
  - Volume
  - Soak
- Globalization
- Usability
- Accessibility
- Retesting
- Regression
- Test scenario
- Test cases
- Types of servers
- Test case
- Design techniques
- ISTQB

SDLC → STLC → DLC



Agile model:- to overcome the drawbacks of all others m

i) Scrum
ii) Sprint > Speak something about this.

→ Built → Built is part of Application, product

Smoke Testing:- Testing the basic features of an applicat
↓
will get confidence on Application (only +ve scenario) is called smoke testing.
* It is partial Testing.

Integration:- Testing the data flow from one module to other module is called Integration Testing.

→ At least two dependent modules are needed to test integration testing.

eg:- pen parts
Gmail modules.
cap & bottle.

System Testing:- It is an end to end testing it is like a end user/real user in a tester server similar to production servers.

Acceptance :- Testing the buisness scenario's for an Application. which usal done by customer

Types of Acceptance Testing
i) Alpha → done in development place
ii) Beta → done in customer place.

Sanity Testing :- This is done on new features and the bug features.
Narrow Testing → Sanity Testing
wide Testing → Smoke Testing.
→ Testing is done on relatively stable build.

Adhoc :- Testing the Application Randomly
Monkey Testing (or) eg:- Throw pen & write

Exploratory :- without requirments. Exploring the applications and understanding scenario's

Compatability :- Testing the application in different h/w & s/w environment, eg:- watsapp in iphones, anroids, windows phone.

Performance :- Testing the response and stability of an application by applying load. (Dec 31)
↓ no. of users.
Tools used here → load runner, Jmeter

Types
- Load → App b/w limit is load
- Stress → if cross load limit if is stress
- Volume → Transfer huge no. of data & see if it works
- Soak → Apply load continues eg:- FB

Globalization :- Testing an application for multiple languages :

Types of globalization :-
1) I18N - Internationalisation
2) L10N - Localization.

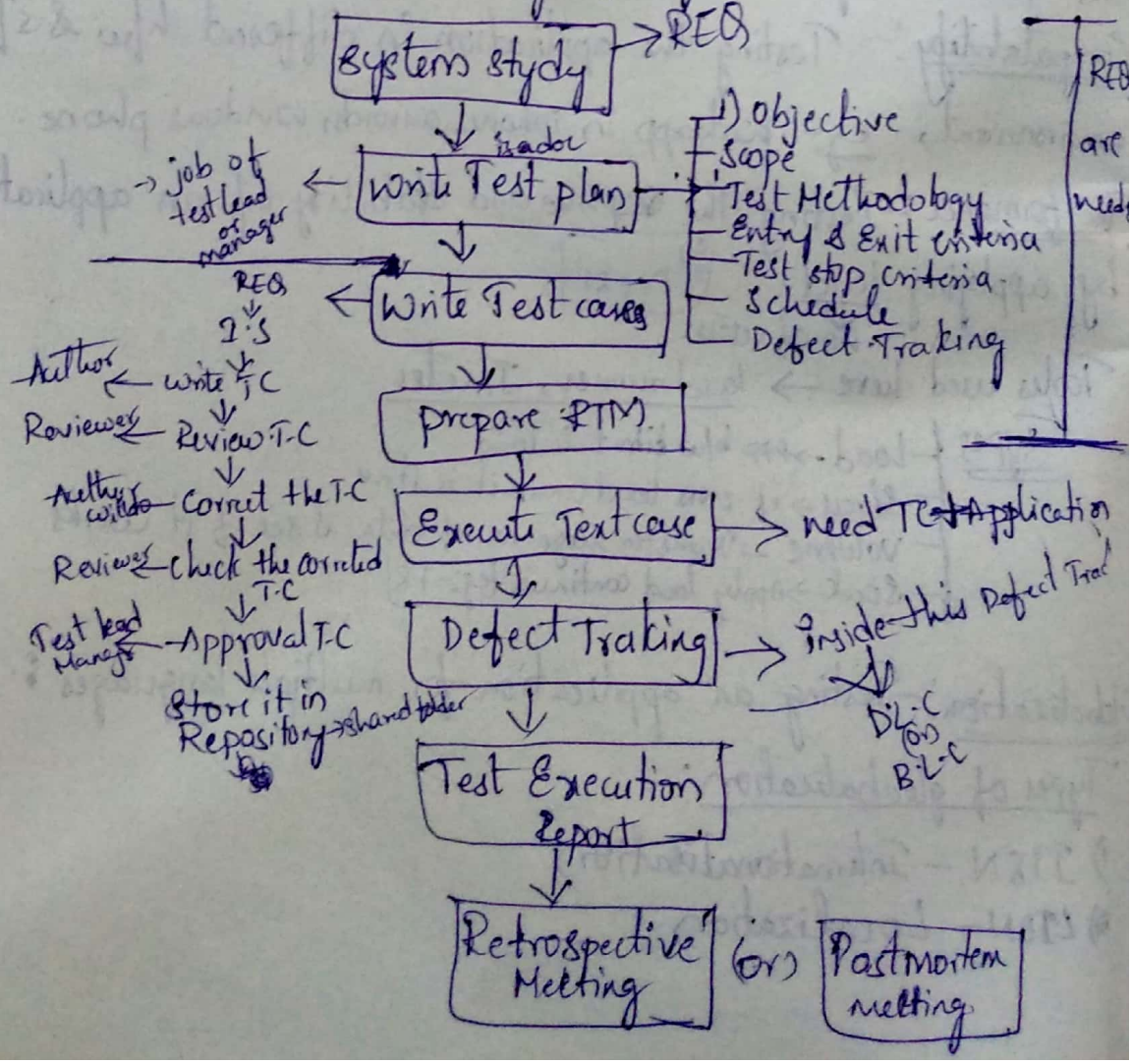
Usability :- Checking whether the application is user friendly or not.

Accessability :- Testing an application from physically challenging people point of view.

Difference b/w Scenario and Test case :-

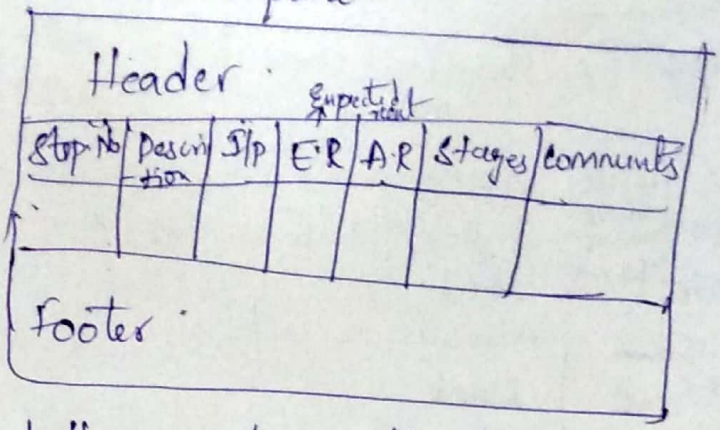
Scenario :	Test case :
1) What to do	1) How to do
2) Scenario will not tell navigation steps	2) Test case will tell navigation steps.
3) Scenario takes less time to write	3) Test case takes more time to write
4) It is high level	4) low level

* STLC → Software testing life cycle.



2) Testplan is the doc which is written for future activities

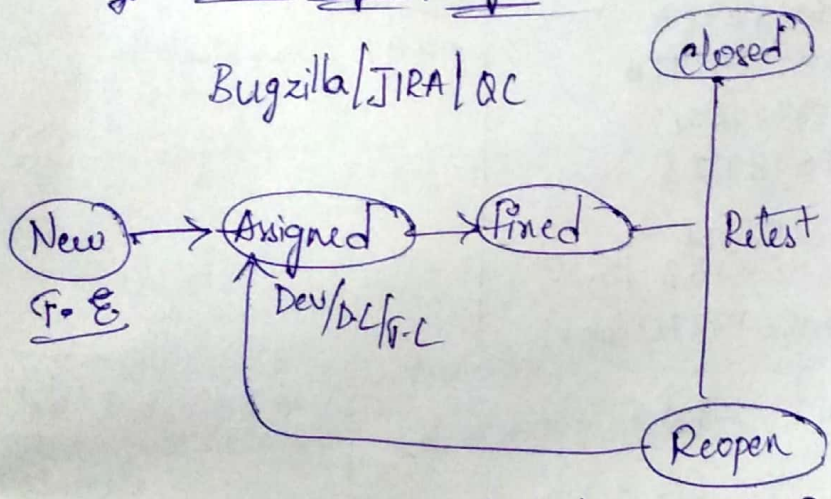
3) Test cases → Test case Template



RTM → checking whether we have atleast one test case for each requirement.

*** Defect/Bug life cycle

Bugzilla/JIRA/QC



Anything which is deviating from a REQ is a defect (Real name)

Informal name of a defect is a bug (Nickname)

When you Raise defect you need → Defect ID
States
Severity
Priority
Brief Desc
steps to Reproduce.
Screenshot (or) video

Severity	Priority
• Blocker / Show stoppers • critical • Major • Minor	Urgent • High • Medium • Low
It is the impact of the defect on customer Business	→ how fast or How fix that the defect has to be fixed

Scenario's for password :-

Valid (+)	Invalid
Should accept the data	Should get error message
• $-12A • $+Aab • $Aabcde $-12A3a456... • abcde • ABCDE	• abcde • ABCDE • 12345 • $#+-* ABCD $$-A | <5 abcd A12345... || >11 Blank

Scenario's for mobilenumber :-

Valid (+)	Invalid (-)
• 0123456789 • 9848111222 • +91 9848612345 • 0123456789	• 1234567891 • 12345667891 • B056789671 • $*6789123 • 12345678 • 9812345167 |<10 • 9848697870 (oh!)

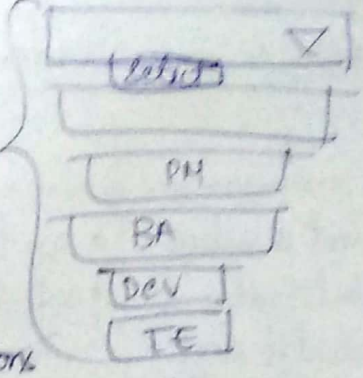
• Blank
• +91 (leave blank)

Scenario for Email -id

valid (+)	Invalid (-)
abc@gmail.com 123@gmail.com $#123@gmail.com abc_123@gmail.com	abc gmail-com abc@gmail.com @gmail.com abc@.com abc@gmail.co.in abc@gmail.com abc@ abc@gmailcom abc@gmail..com

Scenario for Designation

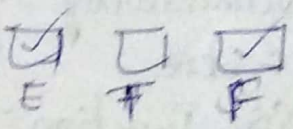
Valid (+)	Invalid (-)
any one value: BA, PM, Dev, TE	• Select "Select" • Select [ ] • Blank. • Try to select multiple options • Try to add new value it should not show error



Select 1 value & change to another value

Scenario for languages

Valid (+)	Invalid (-)
• Select any 2 values any one value: French, English, Telugu • select all values • select 1 value & deselect it • select values in reverse order	[redacted]



(+)ve scenario
• donot select anything (because not mandatory)

Scenario for Date of Birth

Valid (+)	Invalid (-)
[illegible] • Select 1 date & change to other date • Blank. • Select calender.	[illegible] [illegible] [redacted]

Scenario for Address

valid (+)	Invalid (-)
• ABS-## .... 200 ABS-HD 2001	>2001
• Blank.	[redacted]

Scenario for Gender :-

Female ○ male ○

+	−
• select 1 value. • select 1 value & deselect it and change to other value.	• Don't select any option. • Select value & deselect the same value.

* Security Testing (May ask

→ Vulnerability
→ Cookie
→ SQL Injection
} → browse

* ISTQB → Certification for Testing (SEI) Specially for Manual.
* COTS – Commercial off the selff (Comes under Integration testing)
↓
Third party software.

* Scenario for DOB :-

if 14-06-1997

DD	
+	−
1	0
31	00
30	10 (oh!)
29	−1
28	•1
09	Blank
	1@

MM	
+	−
1	0 → 2016 ✓
01	00
12	13
	−1
	10 (oh!)
	•1
	Blank
	1@

YYYY	
+	−
1	0
01	000
001	00
0001	0000
2018	2019

→ without giving slash test (eg – 01111996)

for leapyear

Date	Month	Year	
29	02	2016	→ valid ✓
30	02	2016	→ Invalid.
28	02	2017	→ valid ✓
29	02	2017	→ Invalid.
31	08	2016	→ valid ✓
32	08	2016	→ Invalid
30	04	2016	→ valid ✓
31	04	2016	→ Invalid.

Monthly 28, 29, 30

Scenario's for [submit] Button :- click on

→ Enter all data and submit button (+ve)
→ leave all the feilds blank and click on submit (-ve)
→ Enter only mandatory things (+ve)
there if you get any error it is a "defect"

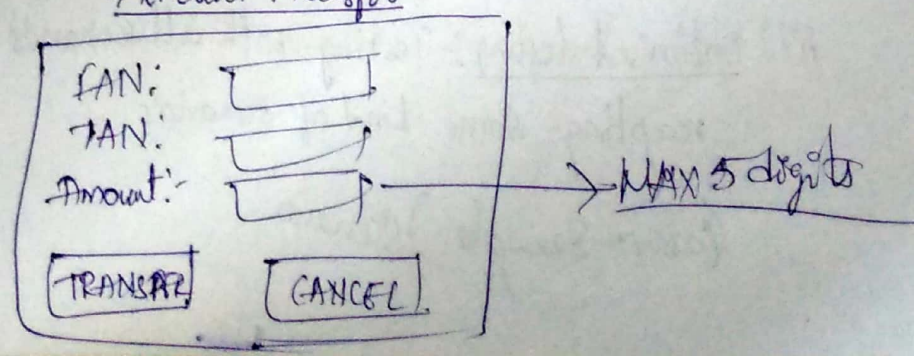
Scenario's for [cancel] Buton :-

→ leave every blank and click on canal button (+ve)
there if you get any error like (fill data & cancel) it is defect
→ Enter all the data & click on cancel (it return to home page) (+ve)

* for Cancel button there is no negative Scenario's.

Application for eg :-

Amount Transfer



Scenario's for Amount :-

ti	—
1	
01	100000
001	9990 (oh!)
0001	blank.
00001	1000/-
99999	100.00
	10,000
	Rs.100
	₹.100 , -100
	₹0.5
	Ten-thousand
	-100
	-10x-10

→ under testing

→ optimised testing

over testing.

Over

200 ×
(a+b)² ×
sin ×
tan θ ×

:- Points to Remember (in FT)

→ All ways testing starts with +ve scenario and later perform testing for negative scenario's

→ if we find a defect, Raise it, continue testing for other scenario's.

→ Don't Assume while doing testing. when confusions are there among the team members.

→ Testing can be done with three different categories

i) Over testing :- Testing an application with reapted kind of scenario's

ii) Under testing :- Testing an application with insufficient set of scenario's.

iii) Optimised testing :- Testing with all scenario's without reapting same kind of scenario's.

(above Example) Scenario

⇒ Testing with expected data from the end user is positive testing.

⇒ Testing with unexpected data from the enduser is negative testing

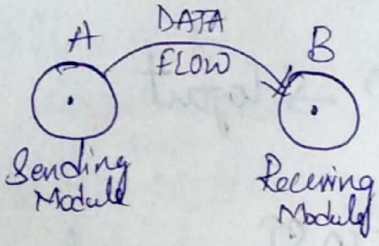
30/5/18

Integration Testing :—

* Testing the data flow between two or more dependent modules.

* Write scenario's for GMail-Application. (FT & IT)

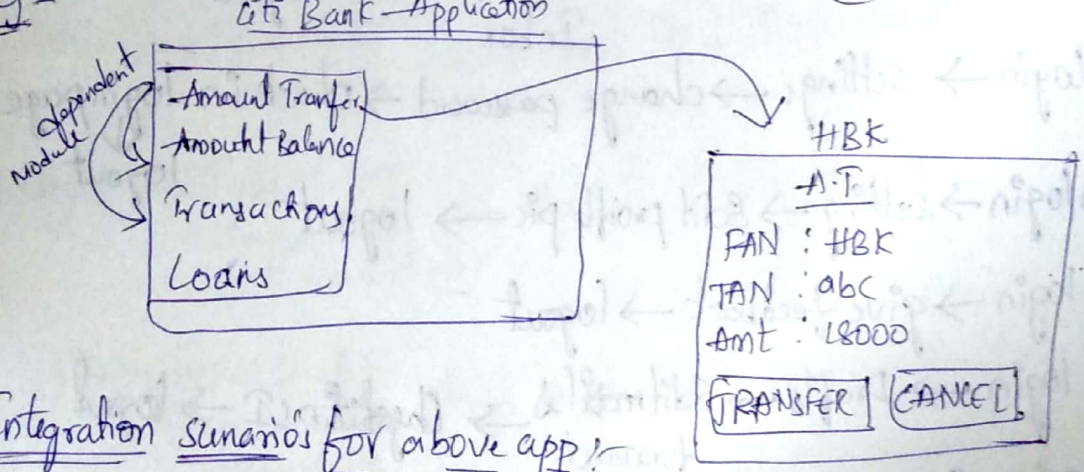
⇒ We need atleast two or more modules to perform integration testing.



eg:
- Gmail
- Watsapp → timings
- FB
- Flipcard
- Red bus

(defect)

eg:-



Integration scenario's for above app:-

1) Login HBK → AT (abc) → check AB → logout
2) login HBK → AT (abc) → check Transaction → logout
3) login as abc → check AB of abc → logout
4) login as abc → check Transaction → logout

Scenarios for Gmail Application:—

1) Login → Inbox → forward mail → check in send Items →
2) Login → Compose mail → check in S.I → logout
3) login → Compose mail attach photo → check in S.I → logout
4) login → Compose mail attach doc → check in S.I → logout.
5) Login → Compose mail with incorrect mail id → saved to drafts → logout.
6) login with correct mail Id → Homepage → logout
7) login → from Send Items forward mail → check in SI → logout
8) login → give self mail Id → check in SI & Inbox → logout
9) login → settings → change password → check in login page ↓ logout
10) login → setting → Edit profile pic → logout
11) login → give feedback → logout
12) login → Drafts → Edit mail & forward → ~~Check~~ in SI → logout.
13) login → Compose mail with attach doc & Incorrect Id → saved in drafts → logout
14) login → Compose mail with attach pic & incorrect Id → Saved in drafts → logout
15) login → Compose mail with attach ~~song~~ video & incorrect Id → saved in drafts → logout

16) login → compose mail with attachment of vedio → check in SI → logout

17) login → check Inbox → logout (F.T)

18) login → check Sent items → logout (F.T)

19) login → check Drafts → logout (F.T)

20) login → To address → Type cc → check in sent Items → logout

21) login → Type CC & BCC → Send → check in S.I → logout

22) login → Type BCC → Send → check in S.I → logout

23) login → Type CC, BCC subject, compose mail → check in S.I → logout

24) login → Type CC, BCC subject → check in S.I → logout

25) login → Type CC, BCC compose mail → check in S.I → logout

26) login → Type CC, BCC compose mail with incorrect pw → check in Drafts → logout

27) login → Type CC, BCC compose mail, subject Incorrect ID → check in Drafts → logout

28) login → Type BCC Incorrect pw → check in Drafts → logout

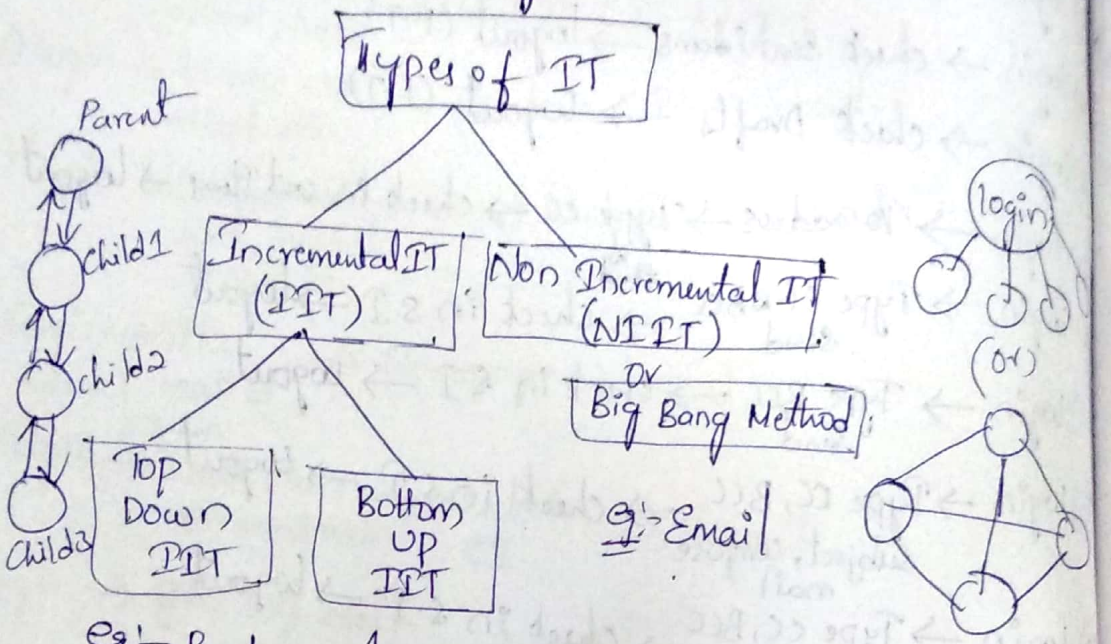
29) login → Type BCC, CC, Subject attach file, doc, pic vedio → check in SI → logout

29) || incorrect pw, ID → check in Drafts → logout

Assignment:- Write only 200 scenario's for gmail account.
Use → Verify, Validate, To check.

1/6/18

Types of Integration Testing:-



Eg:- Email

Eg:- Book my show
Filp card.

* Paytm comes under bot IIT & NIIT

Stubs & Drivers:-

* What is stub and Drivers?
(or)
* If one module is ready and other module is not ready how will you the testing?

→ Stub is dummy module which acts like a child module. It will receive the data from the parent module and send an acknowledgement.

→ Driver is also a dummy module which as like a parent module. It will send the information to the child module and analyses the behaviour of the application.

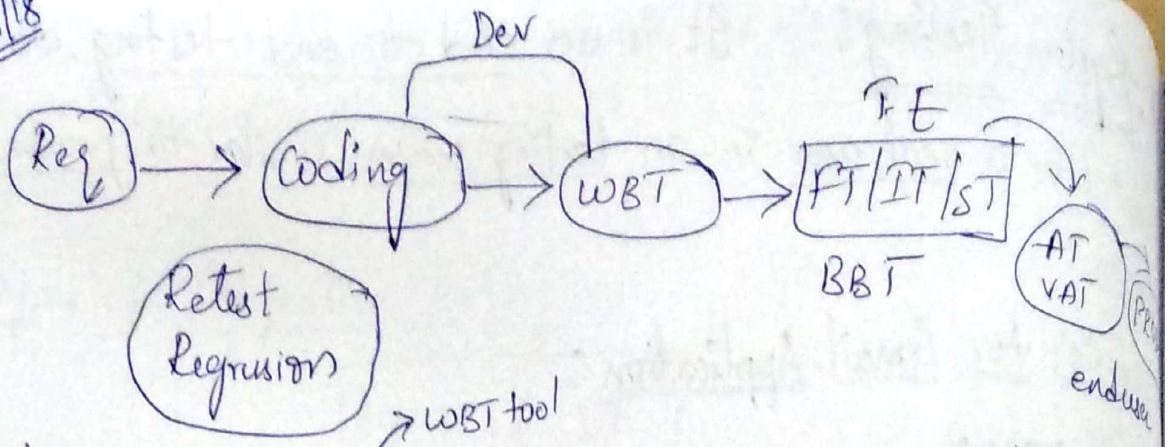
* System Testing :— It is an end to end testing do just like a real user. in an testing server similar to production Servers.

Scenario's for Gmail Application :-

Login page :—

1) Validate the correct userid and password and login to gmail account then home page will open.
2) To check password given invalid password and correct password and and check whether account login (or) not
3) To check Id give invalid mail-id and check whether account login (or) not
4) To validate give correct password and login
5) To validate give correct mailid and login.
6) Verify if we enter wrong mailid id password is enter forget password to relogin
7) Verify by entering correct mailid id and password and cancel it
8) verify by entering invalid mailid id and correct password cancel it
9) verify by entering valid mailid and invalid password and cancel it
10) verify by entering both invalid mailid and password and Cancel it.
12) verify by entering only mailid and click on login
13) verify by entering only password and click on login.

4/06/18



→ WBT tool

→ purpose of Junit testing the Java source code.

Important Note:-

→ In real time the people who are loging to the application are physically different people while testing, T.E will login as different users and perform the testing.

→ In real time the amount entered is the real money while testing it is just a number

→ In real time whenever They is a need to wait to repay the ~~whenever~~ money. Eg:- One month, people have to wait for one month. where as while testing, we will be changing the data in testing server according to our needs.

Scenario's

→ Verify by applying OD and check whether activation fees AF is charge for first time

a) Login as user A, apply OD (20,000) Logout

b) Login as manager, approve pending OD of customer of A after that logout

c) Login as user A. check OD balance (20,000) logout

d) Change the server date to next 30 days. in Testing Server [for user A in database].

e) Login as user A, check OD balance (20650)

f) Login as manager. deposit 20650 in to user A's Account

g) Login as user A Repay OD [20650] check OD balance logout

h) login as user A, apply OD (20,650) logout

i) login as manger, approve pending OD of customer of A after that logout

j) login as user A, check OD balance (20,000) logout

k) Change the server date to next 30 days in testing server [for user A in database]

l) Check the OD balance (20,400]

Requirment change:-

→ If customer change the Req from 250 to 50

Scenario's are

Set AF=50	Set AF=250	Set AF=50	Set AF=250
Apply OD	Apply OD	Apply OD	Apply OD
Set AF=250	Set AF=50	Approve OD	Approve OD
Approve OD	Approve OD	Repay OD	Repay OD
Repay OD	Repay OD		

50 250

Based on customers

REQ for Insurance

New Insurance → 10,000
1st Renewal → No discount
10,000/—
2nd Renewal → 15% discount
if No claim
8000.

Scenario 1

New Insu → 10000
1st Renewal → 10000
2nd " → 8000.

2nd

New Ins → 10,000
1st renewal → 10,000
Claim Ins → 5000
2nd renewal → 10,000

3rd

NI → 10,000
Claim Ins → 5000
1st Ren → 10,000
2nd " → 10,000

4th

New Insura → 10,000
claim Ins → 5000
1st Renewal → 10,000
Claim → 5000
2nd Ren → 10,000

5th

2016 Dec → New Insuran → 10,000
2017 → passed
2018 Jan → claim Insurance. (Then)
Your insurance has Expired. Please Renew
(or)
else → it is a Defect

on myaccount we again navigates to
account is F.T
192) click on Add account, & fill the username
Password is F.T
193) click on google maps & navigates the map p
is F.T
194) All these modules are called system Testing
195) This modules & F.T, I.T, S.T comes under Appl
196) click on I/ps tools & select one character is F.
197) when the character is on/off is I.T
198) click on close option of I/p tool is F.T
199) click on singout is F.T
200) we enter from Homepage to login page is
F.T.

Test cases:

*If we don't write the scenario's what is the problem?

1) There is no consistency in test Execution
2) Quality of Testing dependable on mood of T.E
3) Quality of Testing dependence on memory of T.E

Solution :- Document the scenario's

What is Test case :-

=) Test case is a document which contains all possible scenario's. It contains coloums like step number, description, expected result, input, Actual result status and comments.

Test case Template

Test Case Name :
Project Name :
Module Name :
Release Number :
Requirment Name :
Test data :
Precondition :
Test Case Type :
Priority :

Test Case Description :

StepNo	Description	Input	Expected Result	Actual Result	status	Comments
1.	Open the browser enter the URL	www.gmail.com	login page should be displays	login page is displayed	pass	
2.	Enter username in usesname textfield. Enter password in password textfield.	UN: ab@.. PSS: xyz.	Home page is displayed.	Home page is not displayed	fail	

don't write actual result & status while writing Test case

are not matched then the defect is locked.

Author :
Reviewer :
Approved by :
Approved data :

When we should write test cases?

When developers are developing the application. T.E should write Test cases.

→ Before testing team receives the 1st build test Engineer should be ready with the test cases for all the requirements because once test execution starts test Engineer will never have time to write test cases.

→ During execution when customer changes the requirement developers will be involved in changing the source code. Simultaneously testing team is involved in changing the test cases, Here changes means add feature, modify feature deleting feature

What is difference b/w Testcase & Scenario

Scenario	Test case
1) High level document	1) Low level document
2) It doesn't have navigation steps	2) It consist of navigation steps
3) Scenario doesn't contain navigation steps	3) Test case contains navigation steps
4) It doesn't tell where the deffect is	4) It will tell where exactly defect is.
5) Scenario says what to do	5) Test case says how to do
6) Scenario gives less time to write	6) Test case gives more time to write

Note:-
During test case preparing we should feel the following columns

→ Steps no
→ description
→ Input
→ Expected Result

During testcases execution we should fill the following columns

→ Actual Result
→ Status

When expected result accepted result matches status pass
When expected result accepted result doesn't match status will fail

⇒ Test case Execution means, seeing the testcase, testing application & ap updating actual result accordingly

* Test case Name:

* while writing test case below formate is followed

⇒ project Name – Module Name – Type of Testing – Testcase Number

eg:- Gmail – login – FT – 01

Realease Num:- if we should no clear abaut Realease No. we should contact test lead.

Req No:- Every Req will have a num to know that we should reffer req document and we should update the num in the test case

* Test data:- It is a data that we should create before executing the test case scenario

→ Test data can be crated internally & Externally.

⇒ internally means Test Engineers, for eg:-

for eg:- Checking the integration flow b/w compose & inbox.

⇒ To execute abaut scenario test data is cradentions of A and cradentions of B.

⇒ External means third party creating Test data

for eg:- 1) Credit card dummy no. by bank with cvv, Expired date

2) License no.

* Pre Condition:- It is an action that should be done before executing the test case.

There are two types of pre conditions.

i) Generic precondition

ii) Specific precondition.

eg:- If we cancel the order, then first we have do the order it is specific precondition.

eg:- If we logout from fb, first we have to ~~taking~~ login action can be done.

Priority:- Generally there are three priority.

i) High
ii) Medium
iii) Low

* ~~for~~ the important test cases we will give high priority
* for the notso important test cases we will give medium priority
* for least important test cases we will give low priority

⇒ Giving priority to the test cases will help the test engineer ~~to~~ prioritize his test execution

Test case type:- This section specifies what ~~are~~ types of scenario are written in the body, if the

⇒ if the body become function scenario T.C type become functional test case
⇒ if the body becomes integration scenario's test cas type will become integration test case
⇒ if the body becomes system T.C type becomes system test case.

Test case Description:-

⇒ ~~the~~ Test case Description roughly informs about the scenario which is present in the body.
⇒ When we are writing the description we should use words like check whether or verify wether.

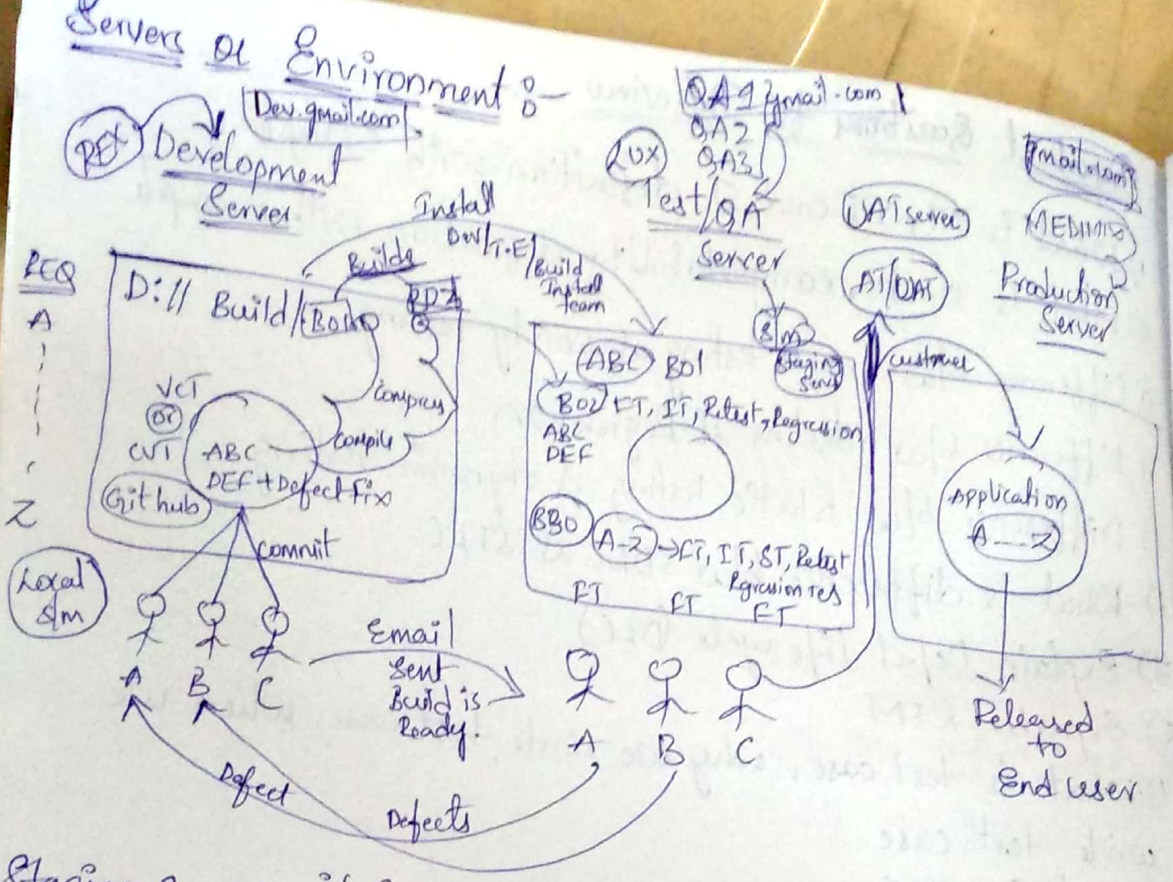
Important Questions for Interview

1) What is FT, IT and ST definition with example
2) What is smoke, campatability definition with example
3) Difference b/w smoke testing & sanity testing
4) Difference b/w retesting & Regrestion
5) Difference b/w static testing & dynamic testing
6) What is difference b/w SDLC & STLC
7) Explain Defect life cycle (DLC)
8) Explain RTM
9) What is test case, why we write test case, when we write test case
10) What is STLC
11) Give brief information about Test plan
12) Explain Agile Methodology.

*Why we write test cases ?—

i) We write test cases to achieve consistency in test Execution
ii) We write test cases to achieve better coverage > possible scenarios
iii) If we write test cases we are depending on the process not on the person
iv) If we write test case we can avoid giving training to the resource. (For new employe)
v) Test case is only document which acts as a proof to the customer that you really tested the application.
vi) If we write test case no need to remember all the scenario's
vii) if we write test case testing will happen in very organised ways
viii) If we write test case, test execution time will be less.

Servers or Environment :-



Staging Server :- it is similar to end user tested by test server or production server.

⇒ Usually/Generally, there are three servers Development server, Testing server, production server. In Development servers, Developer will write the code, In testing server T.E will do the work, In production user will check the product

⇒ In Development server, Dev A, DevB, Dev C will develop the application modules and commit the code into version control tool, then compile the combined code and compress into a build BO1.

⇒ Dev/pment sent mail to testing team to test the Application

⇒ After the build was ready either T.E/dev/Build into team will install the Build BO1 into Test server,

Then

⇒ Then BO1 starts first FT, IT and, ST, Retest and Regression testing.

⇒ Next BO2 has DEF application code, it is also test as like as BO1 i.e; FT, IT, ST, Retest and Regression testing.

⇒ At last BO3 will do the FT, IT, ST, Retest and Regression testing. If there are any defects T.E inform to Dev.

⇒ Then Now Acceptance testing / user acceptance testing will done by the end user. This testing will be done in Test server or in user own server

⇒ Now, then ~~End user~~ customer will Relase the product in to market/end user.

Development Server:- It is a setup which is done mainly for developing an application.

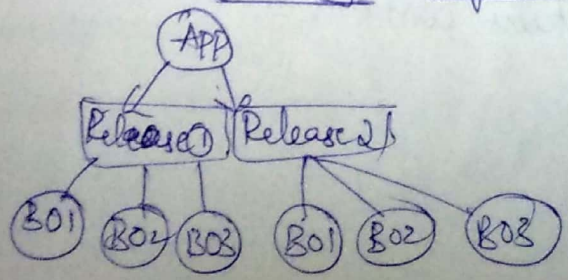
Testing Server :- It is a 2nd setup which is done mainly for testing the application

Production Server :- It is a final setup which is done mainly is used to release the Application to end users and customer can start the business.

Questions

1) How does the real-time product release

2) When we can do Retesting & Regression testing?



⇒Expect Build1 of release 01, we can do Retesting and Regression testing for all the remaining builds of all release.

11/6/18

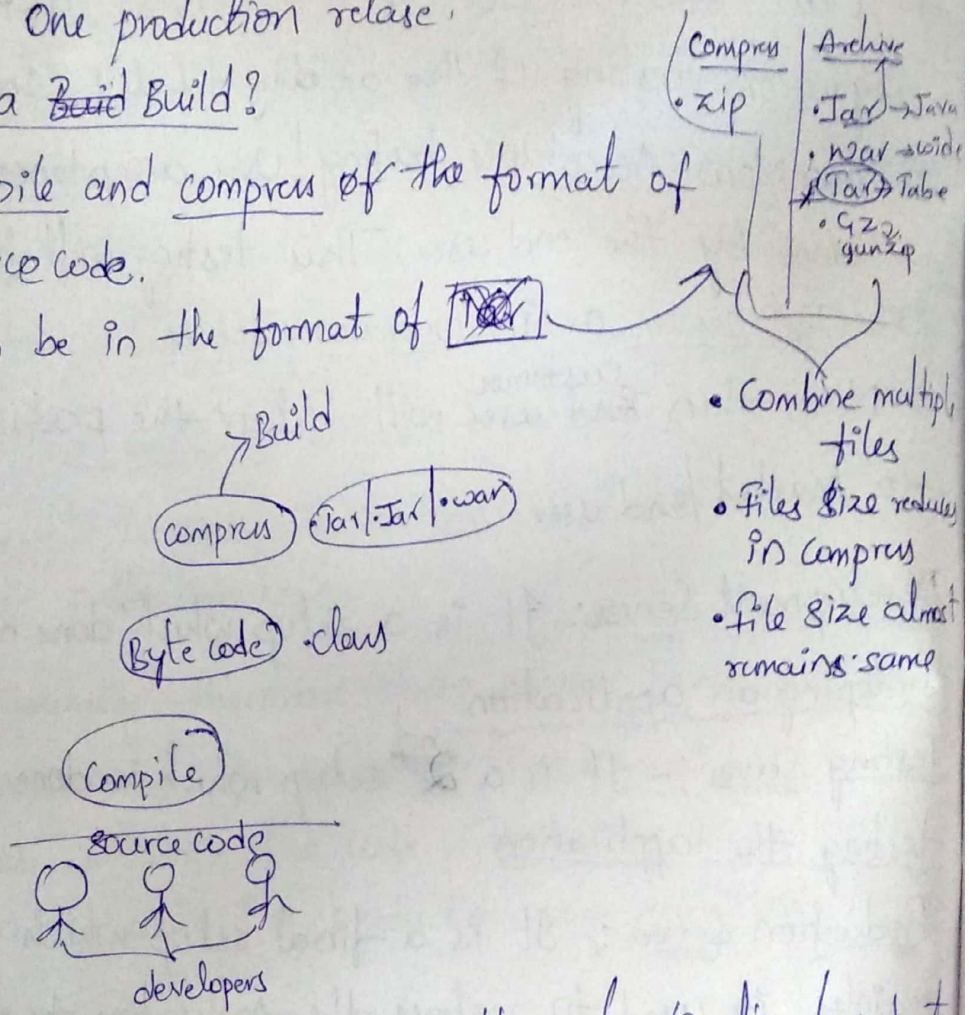
What is a release?

⇒Starting from gathering the requirement, developing application and testing it and finally releasing it to the end users is called one production release.

What is a ~~Build~~ Build?

⇒ The compile and compress of the format of the source code.

⇒ Build can be in the format of [illegible]



⇒A build is also called as a software/Application/product depending upon the contents of the build.

Eg:- If all features contains in the build it is an complete application.

If few of the features contains in the build it is partial applications.

What is test cycle?

→ The effect (or) duration taken to test application once we get a build given is one test cycle.

→ If we have 35 builds test cycles will be 35.

→ What does the build contain?

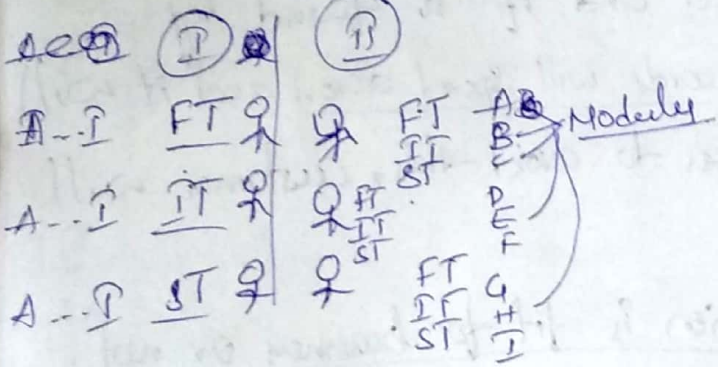
A build can contain

a) New feature (or)
b) old feature (or)
c) defect fixes (or)
d) All of the above (or) it may contain only defect fixes (or) it may contain old feature (or) it may contain new features

* How work allocation should happen

↓

Manager



→ Always testing should happen module wise. not testing type wise.

14/06/18

**** Note :- Once i get a job, always take Responsibility & whenever team Members are not available. Show the proactive. proactiveness during the absence of Manager this will help us to have a better appraisal rating & survive in the company

* "DONOT ESCAPE FROM WORK"

Acceptance Testing:— It is a testing which is done by customer in which they will test all the business scenario's and see whether application is bussiness or not

(or)

Testing the business flow from end to end just before launching the software to end users.

Who will do Acceptance Testing?

Ans:- Usually it is done by customer. If customer need any help T-E will do Acceptance Testing.

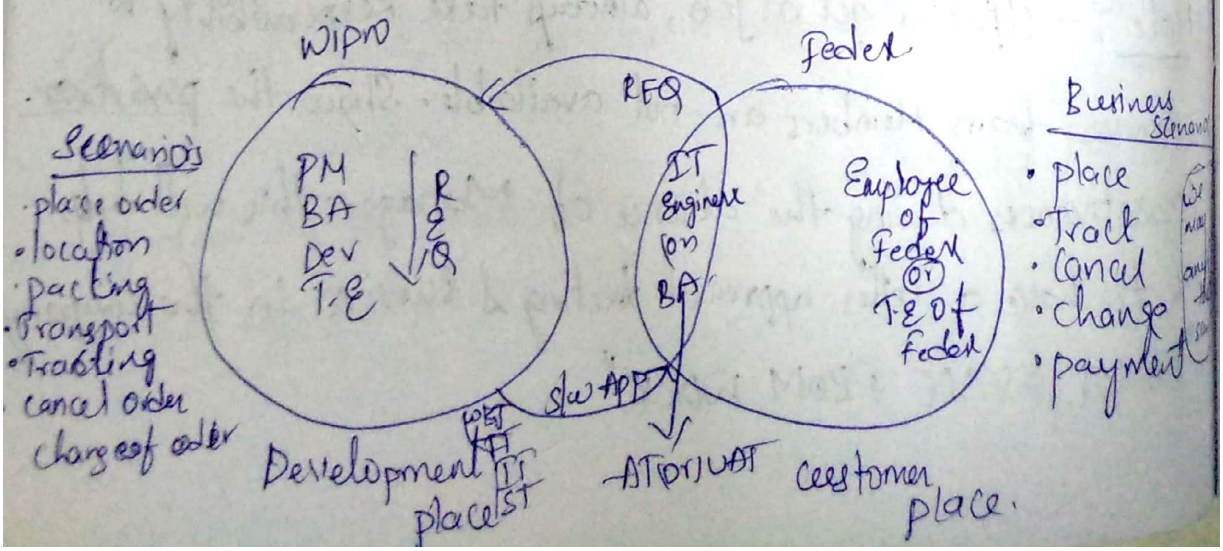
Why Acceptance testing has to be done?

⇒ Because of Business pressure, S/w company might release the product with defect and if it found by end users, Negative words will spread across and it will be loss for a customer to avoid this, customer will do acceptance testing.

⇒ To see whether application is fit for bussiness or not.

Approches of Acceptance Testing:—

Approche No-1:- IT Engeer (or) BA of customer will do A.T. sitting at customer place.



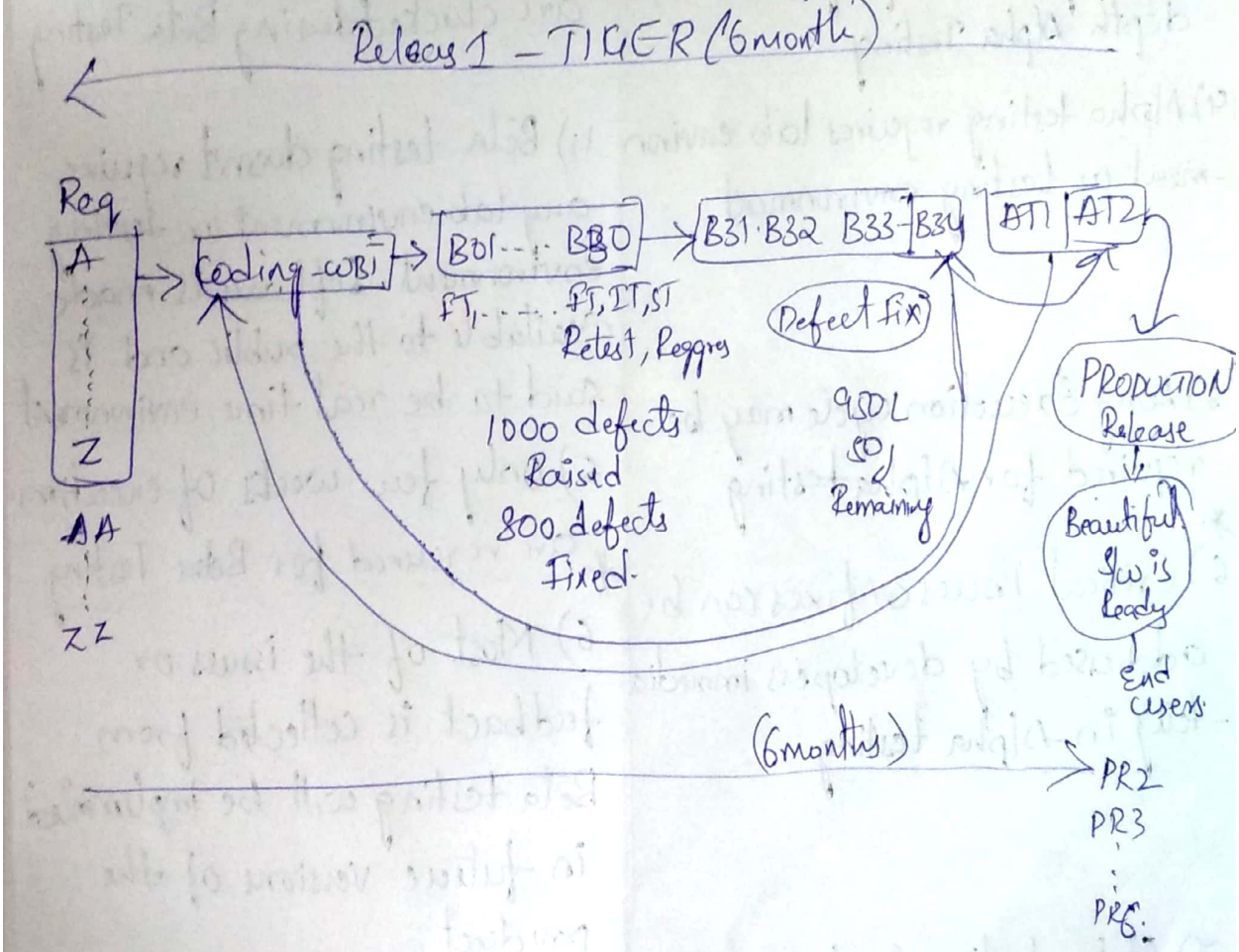
Approach 2:- Empolyees of customer (or) T.E of customer will do A.T. sitting at customn place.

Approach 3:- Empolyees of T.E of software company will do A.T sitting at Customer place.

Approach 4:- Customer [IT Engeneer or BA of customer] will come to developement place and will make test Engener or BA of software company to do Acceptance Testing

Assignment

Difference between α-testing & β-testing (Or) Types of ~~A.T~~



Alpha Testing	Beta Testing
1) Alpha Testing performed by Testers who are casually internal employee of the organization	1) Beta Testing performed by clients or End users who are not employees of the Organization
2) Alpha Testing performed at developer's site	2) Beta testing is performed at client location or end user of the product
3) Reliability and Security Testing are not performed in depth Alpha Testing	3) Reliability security, Robustness are checked during Beta Testing
4) Alpha testing requires lab environment or testing environment	4) Beta testing doesn't require any lab environment or testing environment. Software is made available to the public and is said to be real time environment
5) Long Execution cycle may be required for Alpha testing	5) Only few weeks of execution are required for Beta Testing
6) Critical issues or fixes can be addressed by developers immediately in Alpha testing	6) Most of the issues or feedback is collected from Beta testing will be implemented in future versions of the product
7) Alpha testing is to ensure the quality of the product before moving to Beta testing	7) Beta testing also concentrates on quality of the product, but gathers users input on the product and ensures that the product is ready for real time users.
8) Close for public	8) open for public (9) happens in uncontrolled
9) happens in controled Environment	

What is Hotfix? (or) Incident management system

Whenver there is a production release if the customer comes across a blocker defect. The defect should be fixed immediately and it should be released

Smoke Testing :- (or) Build verification Testing

Testing the basic or critical features of an application before doing a through testing is Smoke testing.

Q) Why do we do smoke Testing :-

A) i) To ensure that basic features are working fine.

ii) Inorder to find whether the Blocker defects (or) basic (or critical) defects. initially only in the basic features

iii) If defects are there in the basic feature, it can be fixed as early as possible by developes, if we raise the defect in the early stage.

Q) When/who do we do smoke Testing :-

i) Test Engineers will do Smoke Testing as soon as they get a build from development team.

ii) Developers can do smoke testing before giving the build to Testing team

iii) Customers can do Smoke Testing before they do acceptance testing.

iv) Build team can also do Smoke testing after installing the build to testing server.

Note (a) In smoke testing, check basic or critical features

→ Here we do testing only on positive scenario, not on negative scenario

→ If is a partial testing, not a complete testing

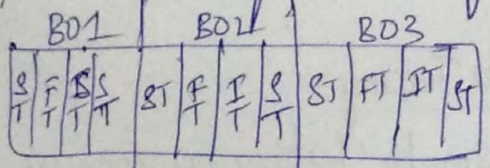
Assignment :- What will you test as a part of smoke test for a) gmail b) facebook c) watsapp d) QspApp.

→ How will you do some testing / How will you manage Smoke Testing

⇒ Once we identify scenario we will write test cases from the test cases which is return, (functional Test case, Integration Test case, System T.C) we will identify the basic features and the positive scenario's of those features and will create another test case document (Smoke T.C)

Once build is give, i will execute smoke Test cases before F.T.C, I.T.C, and S.T.C.

⇒ for every build i will start with smoke testing and since it is a repeated activity, we can go for automation



Note:-

→ It is also called as health check of S/w

→ It is a kind of health check of S/w

Assignment:-

a) Gmail

Smoke testing

1) login
2) Compose
3) Inbox
4) Sent Items
5) logout

Non smoke testing

1) Settings
2) Spam
3) trash
4) drafts
5) Archive
6) Message
7) snoozed
8) Starred message
9) Block
10) junk
11) important
12) Social
13) promotions
14) Bin
15) Calender
16) Contact
17) Help & feedback

b) Facebook

Smoke testing	Non smoke testing.
1) login	1) find Request
2) post	2) Search
3) profile pic	3) Most Recent
4) Message	4) Groups
5) Request	5) pages
6) Notification	6) find friends
7) likes, comment, share	7) Discover people
8) logout	8) Create new page
	9) language
	10) Help center
	11) Activity log
	12) News feed preferances
	13) Account Settings
	14) Terms & policies
	15) privacy
	16) Report a problem.

c) Whatsapp :-

Smoke testing
1) Chats
2) Vedio calls
3) profile pic
4) Audio calls
5) Status
6) Contacts
7) Camera
8) Search

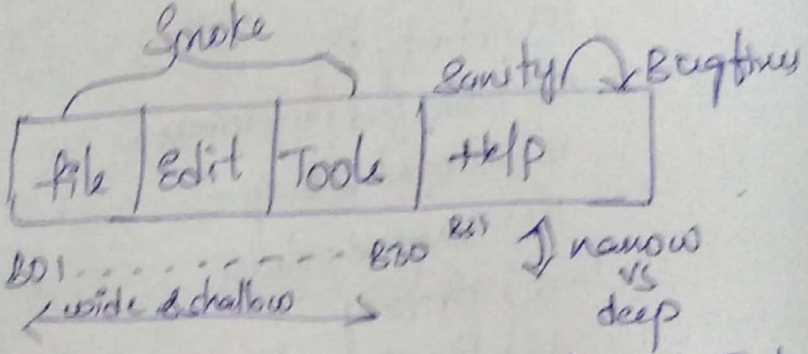
non smoke testing
1) Groups
2) Stared messages
3) pinned
4) delete chats
5) Settings
6) notifications
7) Date storage
8) Invite a friend
9) Help
10) Account
11) privacy
12) Two step verification
13) Security
14) payment
15) mute
16) Archieve
17) Message Info
18) group Info
19) Add shortcut.

Smoke Testing

1) Testing the basic(or) critical features
2) It is an wide vs shallow testing
3) It is positive testing
4) It is done on the initial bills on any new application (Usually it will be unstable build)

Sanity Testing

1) Testing the bug fixes & the new features of the application
2) It is a narrow & deep testing
3) It is positive & negative
4) It is done when we are nearer to the stable build



Formal smoke test

1) It is documented
2) There is a proof

Informal Smoke Test

1) Informal not documented
2) There is no proof

Note:- Do not speak about sanity unless they ask.

Exploratory Testing:-

Exploring the application and understanding the scenario and testing based on the understanding is called Exploratory testing

→ When we do exploratory Testing when there is no Req
→ When there is a Req, but there is no time to go through

*What are the drawbacks in Exploratory Testing?

i) There will be misunderstanding of the REQ
ii) If there are any features which are not developed, we may not come to know it is missing.
iii) If any extra features are added, we will not come to know that it is additional feature

*How will you overcome the drawback of Exploratory testing?

⇒ Interact with ~~der~~ customers, Dev, leads, manager, B.A closely
⇒ Based on the previous domain knowledge.
⇒ Based on the application which is already existing in the market.

*How will you do exploratory Testing?

→ After exploring the App, document the scenario's which are identified. This will help us in future or this will help other team members when we are unavailable

Adhoc Testing:— / Monkey Testing / Gorilla Testing:—

Testing an application Randomly without following any of the formal documents is Adhoc Testing.

Examples of formal doc:- REQ, Testcase, scenario's

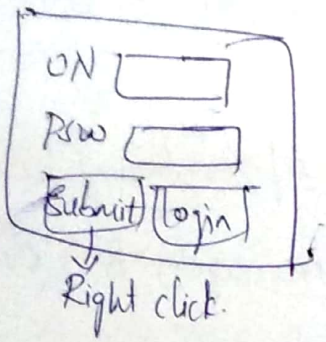
Why we do Adhoc Testing:

→ End users might use the Application Randomly and find defects to avoid that we do adhoc Testing.
→ If we test an Application within the REQ, the number of deff what find will be less. If App is tested outside the REQ we get more deffect

Examples:- ① log in FB (chrome)
log in FB (firefox)
logout from FB (chrome)
Continue using firefox (FB)

② login Gmail (chrome)
login to Gmail (I.E)
Change pwd in chrome
continue using (I.E)

③



④ Road Rush

Gaming application
1) Moving Back
2) Hitting Building
3) hitting tree
4) Sitting in the bus
5) Moving out of the Road

⇒ Adhoc testing has two meanings:-
i) Come up with creative scenario's and test the Application (eg:- Road Rush)
ii) Do the testing without any logic
eg:- click space bar continously for some during ~~is Not~~
eg:- hrs (pressing)

Note:- i) All the defects which are found during adhoc testing need no be fixed unless it is major ~~class~~ / critical / blocker defects

When we do Adhoc Testing?
→ Whenever we reach stable stage of an application (After FT, IT, ST) at last we will think about adhoc testing.
→ This is an optional testing and it will be done if customer demands
→ It is a ~~pref~~ferable testing for Gaming Application

**Compatability Testing:- Testing the software with different hardware and software is Compatability testing.

Why we do Compatability Testing?
→ Once the Application is being build in one platform it has to be tested throughly and should be relase to end users

End users may use the application in different platform and they may come across the defects

→ Negative words will spread across and the no. of users who uses the application drastically ~~ridecess~~ reduces. To check this we have to do compatibility testing.

→ Developing and testing will be happing one environment & if the end users does not use the application in same environment they may find defects.

→ To check whether the app is ~~work~~ consistently working for all the platforms.

For what kind of application we do Compatability testing?

→ for any application which is build for multiple variety of end user.

→ for any app where we expect good revenues [ROI → Return on Investment]

Eg:- Customer will decide the platform based on the no. of users who are using that particular platform. Based on ROI

When do we do Compatability Testing:-

→ When application is developed for multiple platform/environment

→ When application is stable in the main or base platform, then we think doing compatability testing for other platforms.