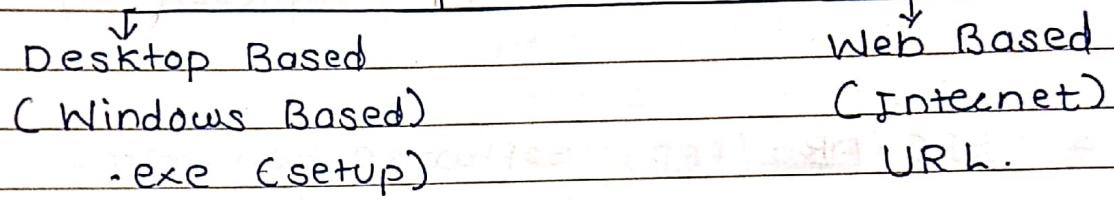


Functionality : Core behaviour of Application

Testing : To test whether application works according to its functionalities.

Production : End users will start using that particular application.

Application .



Selenium : Automation testing tool for web Based Application.

Client / Customer : Gives Business Requirement / client Requirement / customer Requirement.

* In maintenance Project the change is called as Change Request or RFC (Request For change)

* Scratch : Application development from zero level.

* Business Analyst : Person who collects requirement from Client & he is non-Technical person.

Client → BA → BRS / BRD → SRS → Technical Team

* BRS / BRD : Business Requirement specification / document .

- This document describes customer requirements to be developed ,
- It's a bridge between technical people and client .
- It consist High level Business language .
- This document is prepared By BA .

* SRS / FRS / FRD : Software Requirement specification
Functional Requirement Specification
Functional Requirement Document

- This document is prepared By BA
- It consists Functional Requirements to be developed and System Requirements to be used .

* What SRS consist of ?

- 1> Functional Requirements
- 2> Functional Flow Diagrams
- 3> Use cases
- 4> Snap shots
- 5> System Requirements .

Use cases : Defines Functionality in terms of Input , Output & process .

e.g. login : I/P : UName Process : click login
O/P : login success / unsucces .

SRS Format: .doc or .pdf

eg.

<u>BRS</u>	<u>SRS</u>
• <u>What to develop.</u>	How to Develop.
Addition of two Numbers.	FRS: Functional Req Specif. * 2 inputs * + Operator * 1 output

NFRS: Non-Functional RS.

- * Blue color in screen
- * Run on Windows & Linux
- * Addition within 0.5 sec.

* SDLC : Software Development Lifecycle.

LCD [Lifecycle Development] LCT [Lifecycle Testing]

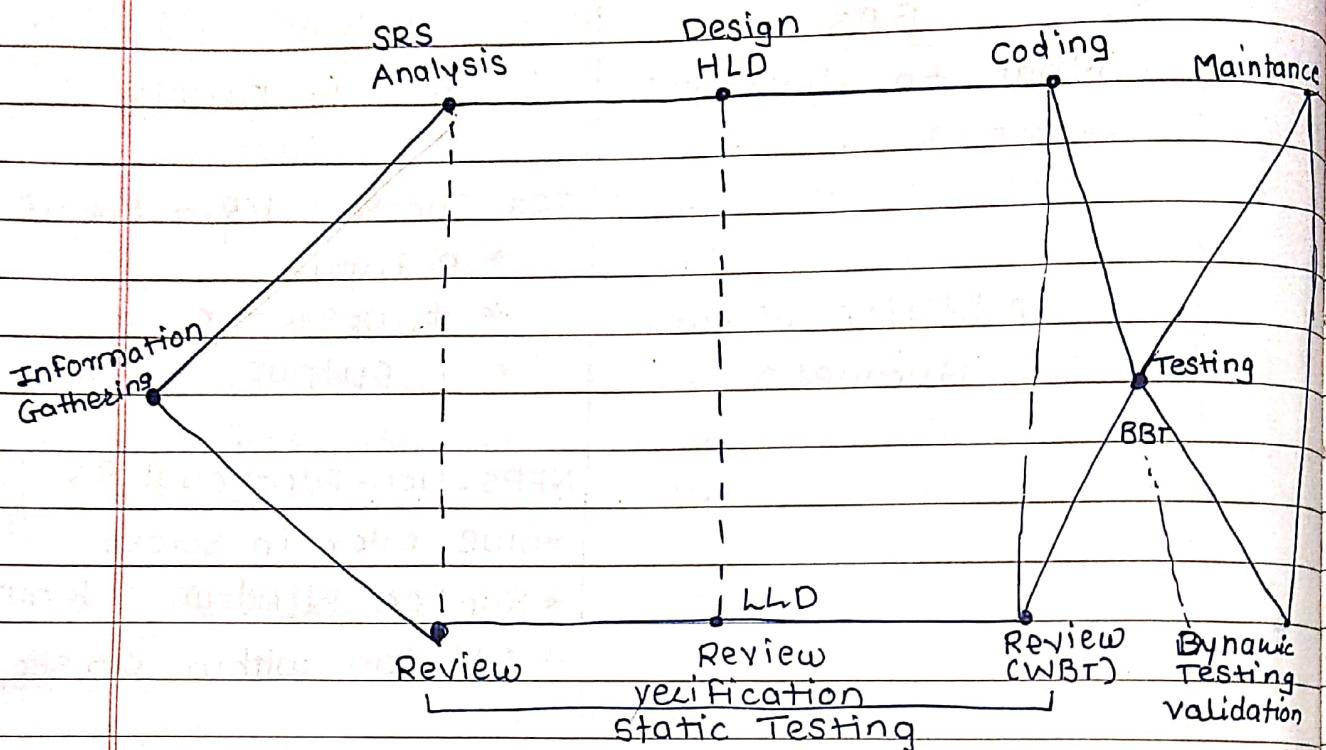
Team Size : $\frac{\text{LCD} \text{ [No. of Developers]}}{\text{LCT} \text{ [No. of Testers]}}$

Normal Project Ratio : $\frac{3}{1} \frac{\text{Dev}}{\text{T}}$

No. of Testers : Manual Testers : 5
Total : 7 Automation Testers : 2

Total Team Size : * No. of Testers \times 3 No. of Dev.
 $= 7 \times 3 = 21 + 1 \text{ BA} = 22 - 24$.
 $8 \times 3 = 24$ Around.

SDLC Phases : Graphical representation of SDLC is called Fish Model.



1) Information Gathering

- BA is going to collect the customer requirement from client.
- After that 2 documents will be prepared: BRS and SRS. [Translates Business lang. into technical lang.]
- BA is going to pass SRS to technical team.
- SRS will be shared with developers and testers on same day.

2) SRS Analysis

During this stage, issues/query in requirements will be solved.

Query - Action items

- * How to solve the issue or to whom you will interact or Have you ever interacted with client?

As a Testee, I am going to follow hierarchy.

- Test Team Members
- Test Team head
- Development Team
- Business Analyst

Lastly, if query is still not solved then with the permission of project Manager, I am going to interact with the customer or with client.

3) Design

There are 2 stages in design : High level Design
Low level Design

HLD :- Gives the architecture of the software product to be developed.

- It contains Main Modules which are going to develop.
- Done by architects and senior Developers.
- Data flow, Flowcharts.

LLD - Done by senior developers.

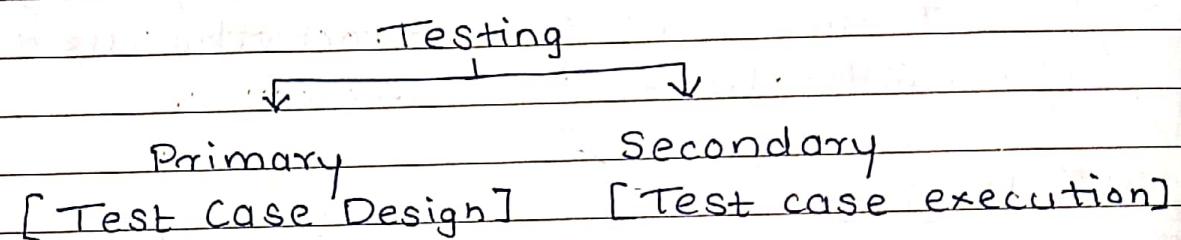
- It describes how each and every feature in the product should work and how every component should work.
- Contains detailed description of each & every module or component.
- Class diagrams.

4) Coding | Programming :

- Done by all Developers : Seniors, Juniors.
- This is the process where we start building the software and start writing the code for the product.

5) Testing :

- Done by test engineers.
- It is the process of checking for all defects and rectifying it.



Test data : Input data for testing.

Build : Development team will provide this to Testing team (URL / setup).

Test case design : Test cases are written in.xls.

Step Name	Design step	Expected
Step I	Launch the browser & Enter the URL. http://-----	Login Page should open successfully.
Step II	Enter valid vName.	Text box should accept input.
Step III	Enter valid Password.	TextBox should accept sP.
Step IV	click on Login Button	It should redirect to Homepage.

WBT : White Box Testing

- Developers perform this testing.
- Code-Level Testing
- Unit Testing

Static Testing

- Testing was done without executing the program.

- This testing does the verification process.

- Static testing is about prevention of defects.

- BA is going to review his own SRS document, Design team will review their own design (HLD + LLD) and after that Coding team are going to check their own code i.e. White Box Testing.

- Review is process of static testing.

Dynamic Testing

Testing is done by executing the program.

Dynamic testing does the validation process.

Dynamic testing is about finding and fixing the defects

During this, Test engineer is involved & he is going to check complete/whole functionality of the application.

- Black Box testing is also called as dynamic testing.

Verification

- It is a static practice of verifying documents, design, code and program.
- It does not involve executing the code.
- It is human based checking of documents & files.
- It uses methods like inspections, reviews, walkthroughs & desk checking.

Validation

- It is a dynamic mechanism of validating and testing the actual product.
- It always involves executing the code.
- It is computer based execution of program.
- It uses method like black box (Functional) testing, gray box testing & white box (Structural) testing.

- White Box Testing
Glass box / Clear box Testing
- Developers
- coding level testing techn.
- In this, developers are going to concentrate on only positive scenarios.
- Developers will find defect in presence of code.

- During WBT, developers going to concentrate on front end (UI)

Black Box Testing
System & Functional Testing.

- Testees
- Build level testing tech.

- In this, testers are going to cover positive & negative Scenarios.

- Testees will find out defect in presence & absence of code.

- During ABT, Testees is going to concentrate on internal functionality that depends on external interface.

* Gray Box Testing :

- Combination of white Box testing and Black Box testing.
- During this, only those testers are involved who are having some knowledge of coding.

* Red Box Testing :

- Combination of hardware and software. [embedded]

* Process / Models / Methodologies :

- To implement SDLC, Organization have to follow some process/Model/Methodology.

- 1) Waterfall Model
- 2) V-Model
- 3) Agile Methodology
- 4) Spiral Model
- 5) RAD Model.

Duration / Release / Sprint :

1) Waterfall : 3 Months.

2) V-Model : 3 Months.

3) Agile Methodology : 1 Months / 15 days.

Productivity OF T.E

Test case Design : 15 - 20 (per day)

Test case execution : 20 - 25 (per day).

* Which Model to choose For Software development?

Who is going to decide which Model to Follow?

1. When the client is new or having no knowledge of process model then organization can decide which model to be used.

2. When client is having its own technical team then organization & client will mutually decide which model to refer.

* There are 3 Factors using that organization is going to decide how many requirements we need for specific release / sprint:

1) Complexity of Requirements [More complex requirement then less will be taken]

2) knowledge: [Previous experience of same domain or new domain research]

3) Efforts : Time duration for Dev + Test.

This process is called estimation process.

* Who is involved in estimation Process?

3 persons : Business Analyst [BA]

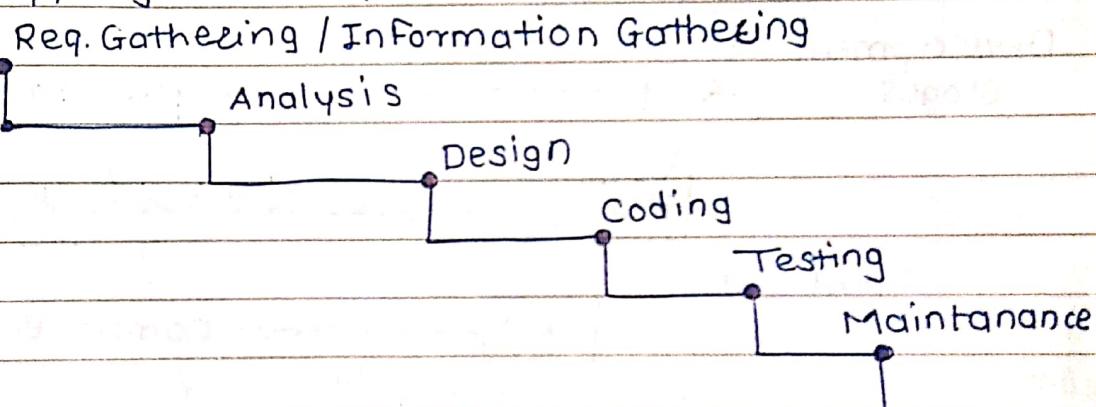
Development Head

Testing Lead

- * One Model will be followed throughout development of One Project.

↳ WaterFall Model: linear Sequential Model

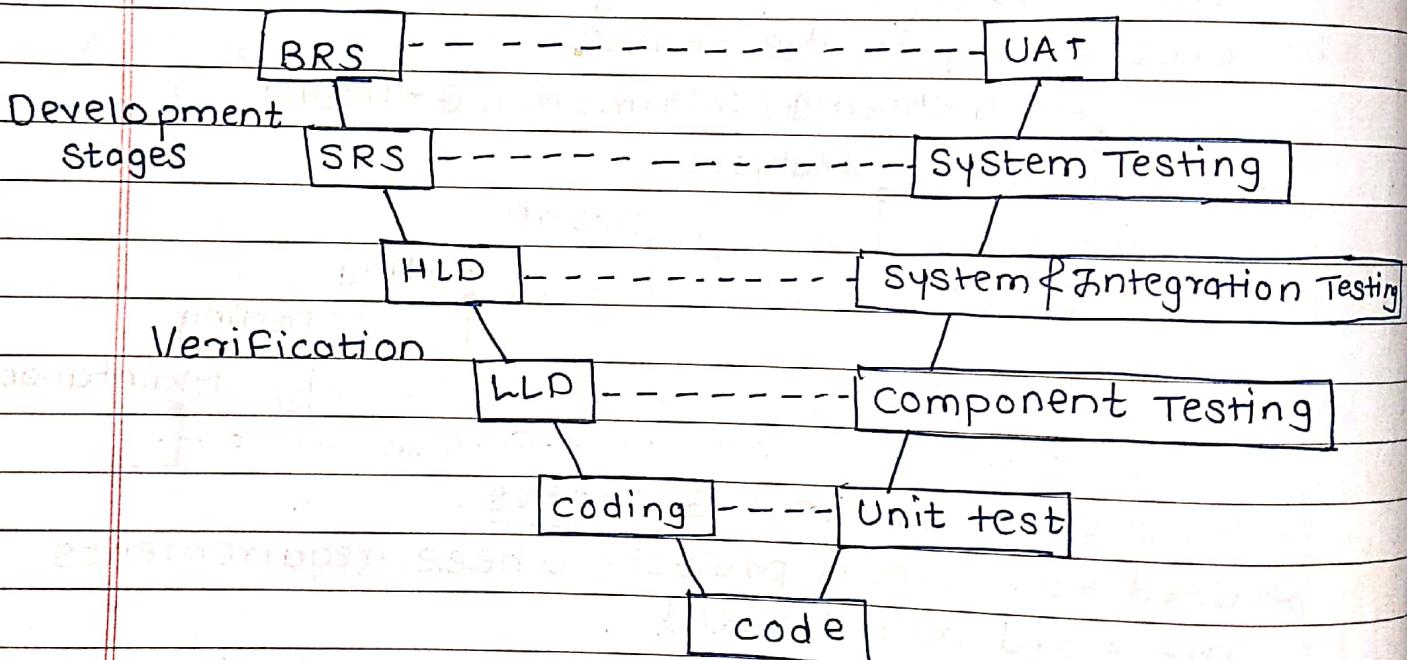
- First Model which was being introduced in the organization in 1996.
 - It is a Cascade SDLC model in which development process looks like the flow, moving step by step through the phases of analysis, projecting, realization, testing, implementation and support.
 - In this, each phase must be completed before the next phase can begin and there is no overlapping in the phases.



- Release Duration : 3 Months
 - Used for small projects where requirements are fixed and freezed.
 - In this model, we cannot accept the change in requirement because it works in linear fashion. So if we change requirement then again we need to move from first stage to last and then it will affect delivery deadlines and planned process.

2) V Model : Verification & validation Model

- Release duration : 3 Months.
- It is an expansion of classic waterfall model.
- It's based on associated test stage for the every development stage.
- This is a highly-disciplined model & the next phase starts only after completion of previous phase.
- There is mapping between development stage & Testing phase.



BRS \longleftrightarrow UAT

SRS \longleftrightarrow System Testing

HLD \longleftrightarrow System & Integration Testing

LHD \longleftrightarrow Component Testing

Coding \longleftrightarrow Unit Testing [WBT]

- This model is costly than waterfall model. So For this model, client will decide to use or not.

Component Testing :

Test engineer is going to check each and every module separately.

Integration Testing :

T.E will test modules after combining.

UAT : User Acceptance Testing

Customer Acceptance Testing

During this, client is involved. Before production UAT will be done.

* Agile Model

- We have used this model for Banking / Telecom project.

- Agile is a software development approach where a self-sufficient and cross-functional team works on making continuous deliveries through iteration and evolves throughout the process by gathering feedback from the end users.

- Sprint wise Delivery.

- Flexible Model and costly than previous models.

- Usage of Automation Testing.

* Types of Agile :

•Lean • Kanban • Scrum • XP • DSDM • FDD.

- In my previous organization and for past projects, we were working on Agile methodology.

* Actors in the Agile

Developers + Testers Team = Scrum Team

<u>V Model</u>	<u>Agile Model</u>
Client	Stakeholders
BA	Product Owner
Project Manager	Scrum Master
Release	Sprint
SRS	User Stories
Release Duration : 3 Months	1 Release : 1 Month Sprint 1 + Sprint 2 [15 Days] [15 Days]

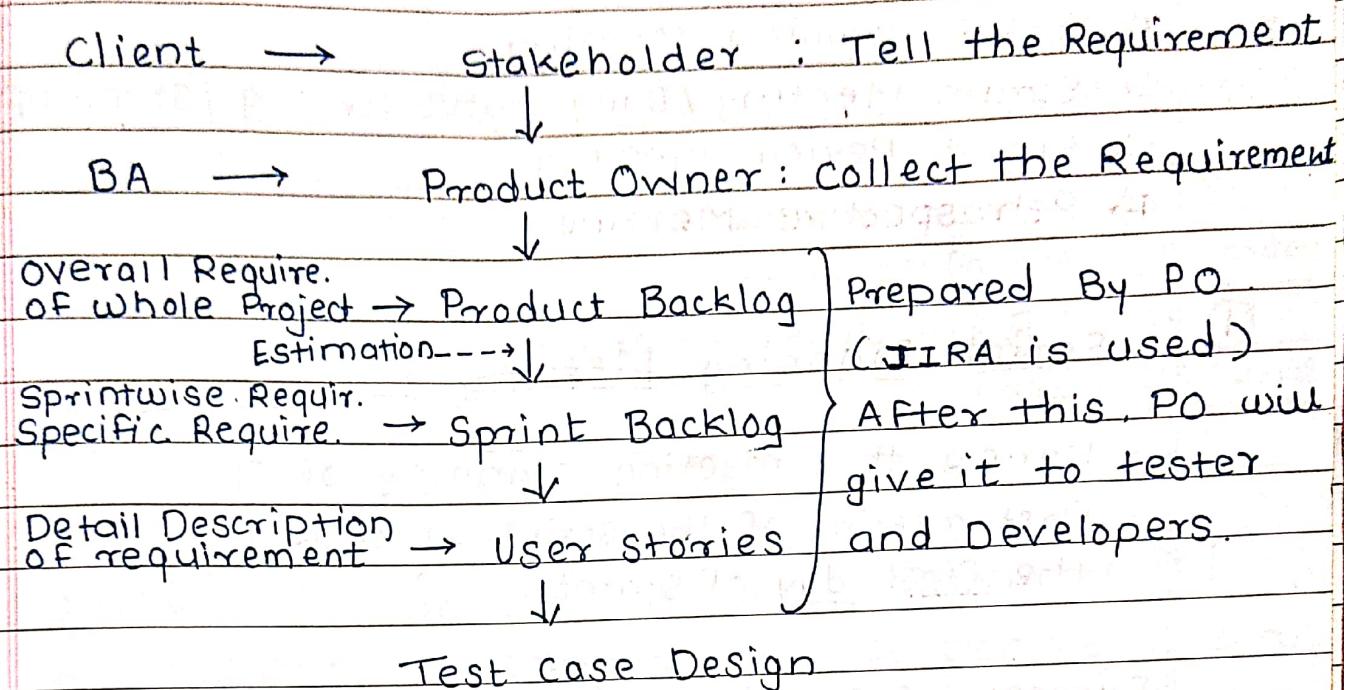
JIRA : Automation tool for project Management and defect tracking.

Backlog : Requirements.

Sprint : Predefined interval or time frame in which the work has to be completed and make it ready for review or ready for production deployment.

2 Weeks : 1 Week Development + 1 Week Testing

* Architecture of Agile : Working of Agile



Estimation :- Converting Product Backlog into Sprint Backlog.

- Product owner, Testing & Development head are involved in this Meeting.

User Stories :- Consists of detailed description of each and every requirement which is being mentioned in sprint Backlog.

User Stories Analysis : Tester and Developer.

* Meetings in the Agile: Ceremonies in Agile. Events in Agile.

- 1) Sprint planning Meeting
- 2) Scrum Meeting / Daily status Meeting / Stand-up
- 3) Sprint Review Meeting
- 4) Retrospective Meeting

① Sprint Planning Meeting

- During this meeting, planning of sprint is done.
- First meeting of the Sprint which will happen on the first day of Sprint.

Purpose : Discussion and plan of requirement and completion, Test and development lead will do resource and Job allocation.

Product owner (PO) will explain user stories.

Involvement: Testing team + Development team + Scrum Master + Product owner.

Frequency : Once per sprint

Time : 2-4 hours. (Day 1)

Day 2: User stories analysis by Developers & testers [2-2 Days]

- After this testers will start identifying test scenarios and start test case design.

2 Week Sprint: 1 Week - User Stories Analysis, Test scenarios identification, Test case Design & Review

2 Week - Start executing test cases on Build.

② Scrum Meeting / Daily status meeting / Stand-up Meeting

Purpose - Every day testing team & development team will update to Scrum Master.

about :

- 1) What did the team members do yesterday?
- 2) What did the team members plan to do today?
- 3) Any roadblock.
- 1) Today's action plan
- 2) Whatever we've performed yesterday
- 3) What are problems we are facing in requirements / Test case & scenarios.

Frequency - Every day / Daily.

Duration - 30 - 45 minutes.

Time - 9:30 am - 10:15 am. [Regular shift]

Involvement - Testing team + Development team + Scrum Master + Product Owner.

Scrum Master - The person who is going to keep track of the schedule & plan of sprint.

- Scrum Master is the chair Person for this particular meeting.

③ Sprint Review Meeting

- In this particular meeting, the task we have completed in this sprint that will be reviewed by stakeholder.

- Frequency - Once per sprint

Duration - 2 - 4 hours.

Involvement - Testing team + Development team + Scrum Master + Stakeholder.

④ Retrospective Meeting / Improvement Meeting

- At the end of the sprint. (Happens after the review meeting)

Purpose : To discuss openly what went well and what didn't during the sprint, so that the team can togetherly find better ways to meet the project's goals. Here the team can discuss internal processes as well.

- What went well during the sprint.
- What did not go well in the sprint?
- Lessons learned
- Action items

Frequency : Once per sprint

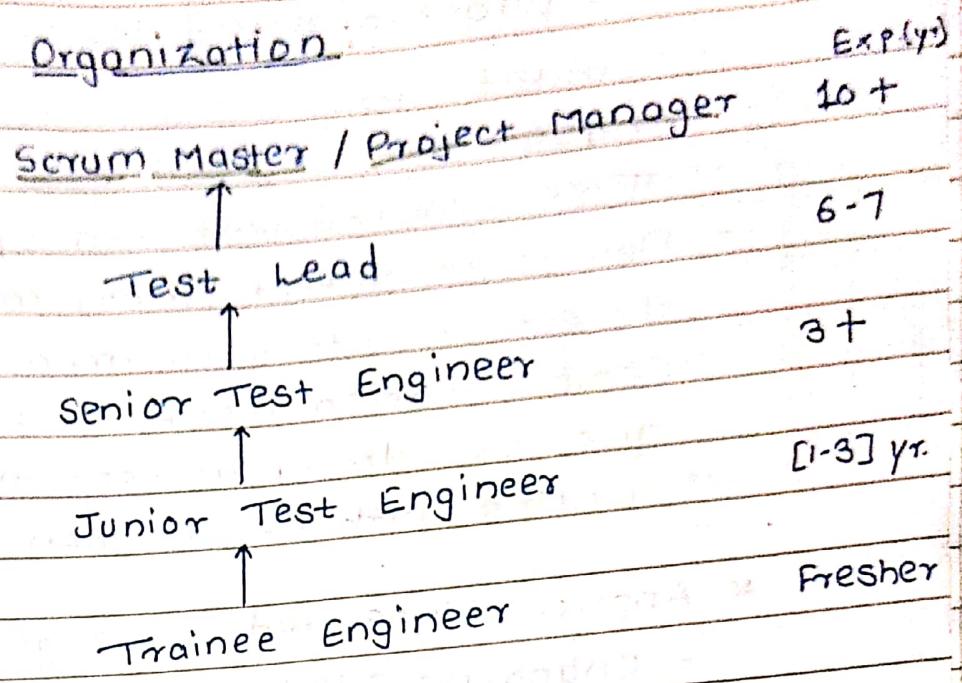
Duration : 2 - 4 hours.

Involvement : Scrum Master + Test team + Dev. Team

* Features / Characteristics of Agile :

- 1) Sprint wise Delivery : Deliver product to Stakeholder sprintwise.
- 2) Daily Scrum Meeting
- 3) Use of Automation : To reduce time and manual efforts, it uses automation tools. (Regression testing).
- 4) Quality - Testing is integrated throughout the lifecycle, enabling regular inspection of the working product as it develops.
- 5) Accommodating change due to volatile requirement (Flexibility / Agility).

* Hierarchy in Organization



* Roles and Responsibilities of Test Engineer :

- 1) SRS Analysis
- 2) Creating Test Scenarios
- 3) Designing Test Cases
- 4) Creation of Test Data
- 5) Execution of Test cases
- 6) Traceability Matrix.
- 7) Defect logging & Reporting & tracking till closure.
- 8) Client Interaction.

Defect Log : Functionality / code issues are logged through tools : JIRA & HPLM.

Defect Report : Environment related issue through E-mail.

* What challenges and difficulties you have faced during 3 years of experience ?

As a tester, there are different challenges we face while working in organization :

* STLC process related challenges :

- limited time available for testing , late engagement of testing team
- Test environment issues / Build installation.
- Delays in defect fixing / improper Fixing of defects resulting in impacts.
- Communication issues like ~~no~~ unresolved queries , delay in response , communication hurdles with development team .

* Application related challenges :

- Changing requirements .
- Non-availability of right set of test data .
- Build quality issues .

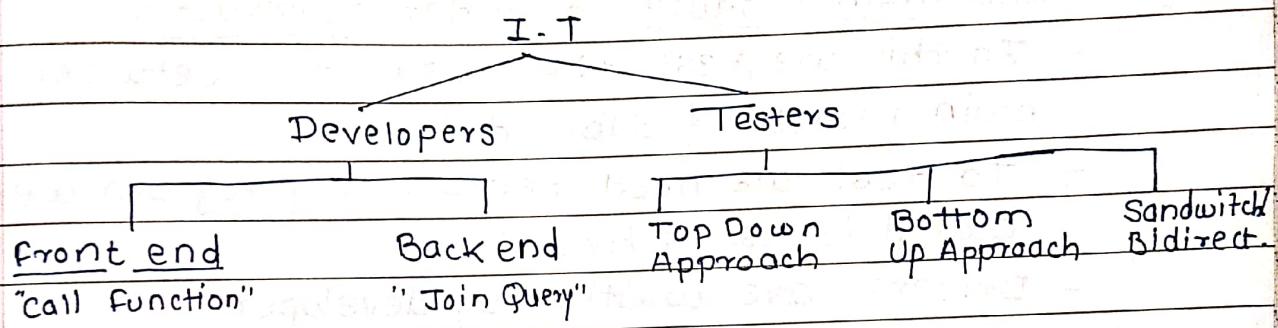
* What will be your daily Routine ?

- 9:00 - Read emails for any important issue or announcement.
- 9:15 - Go through the allocated tasks by the team Lead on email or xls sheet or whatever tool used for communication .
- 9:30 - Scrum Meeting :
- 10:15 - Plan for test cases or test execution for the day

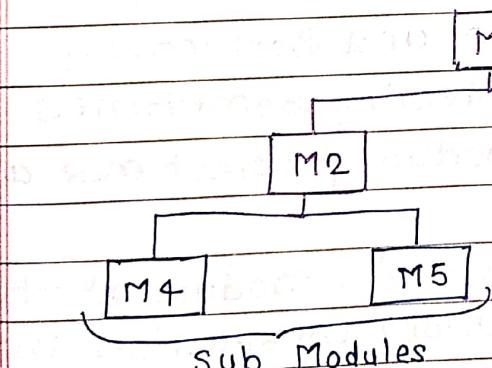
★ Integration Testing

- Interface Testing: During this, we check whether there is flow of data or resources between two modules.

- During IT, different and independent modules are combined together and tested as unit.
- This testing is performed by both testers & developers.



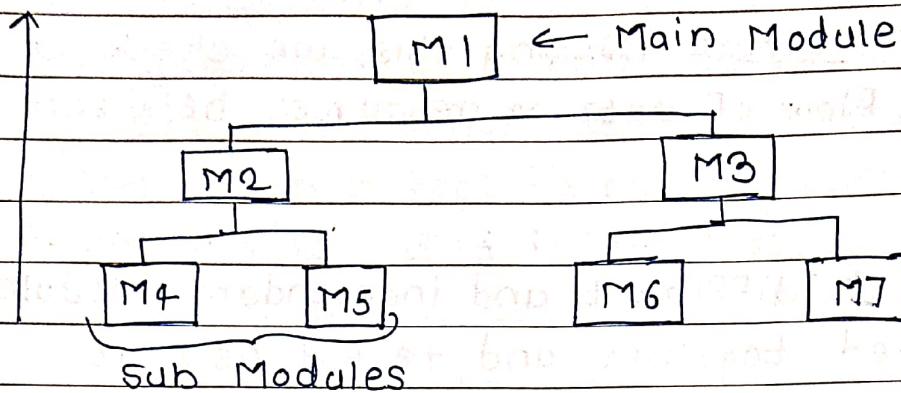
1) Top Down Approach



- Main modules are already developed & tested
- Sub modules are under constr.
- Physically not ready with us.
- Temporary Prog. to test modules

- Testing will be performed from top to down.
- M1 will be tested first then M2 & M3 and Lastly all the modules M4, M5, M6 & M7.
- stub - Temp. prog. written in XML lang. & prepared by Dev.
- XML: Tags & values, <tag> Value. & Universal lang.
- XML Edit: Notepad ++.
- FTP: File Transfer Protocol.

2) Bottom-Up Approach

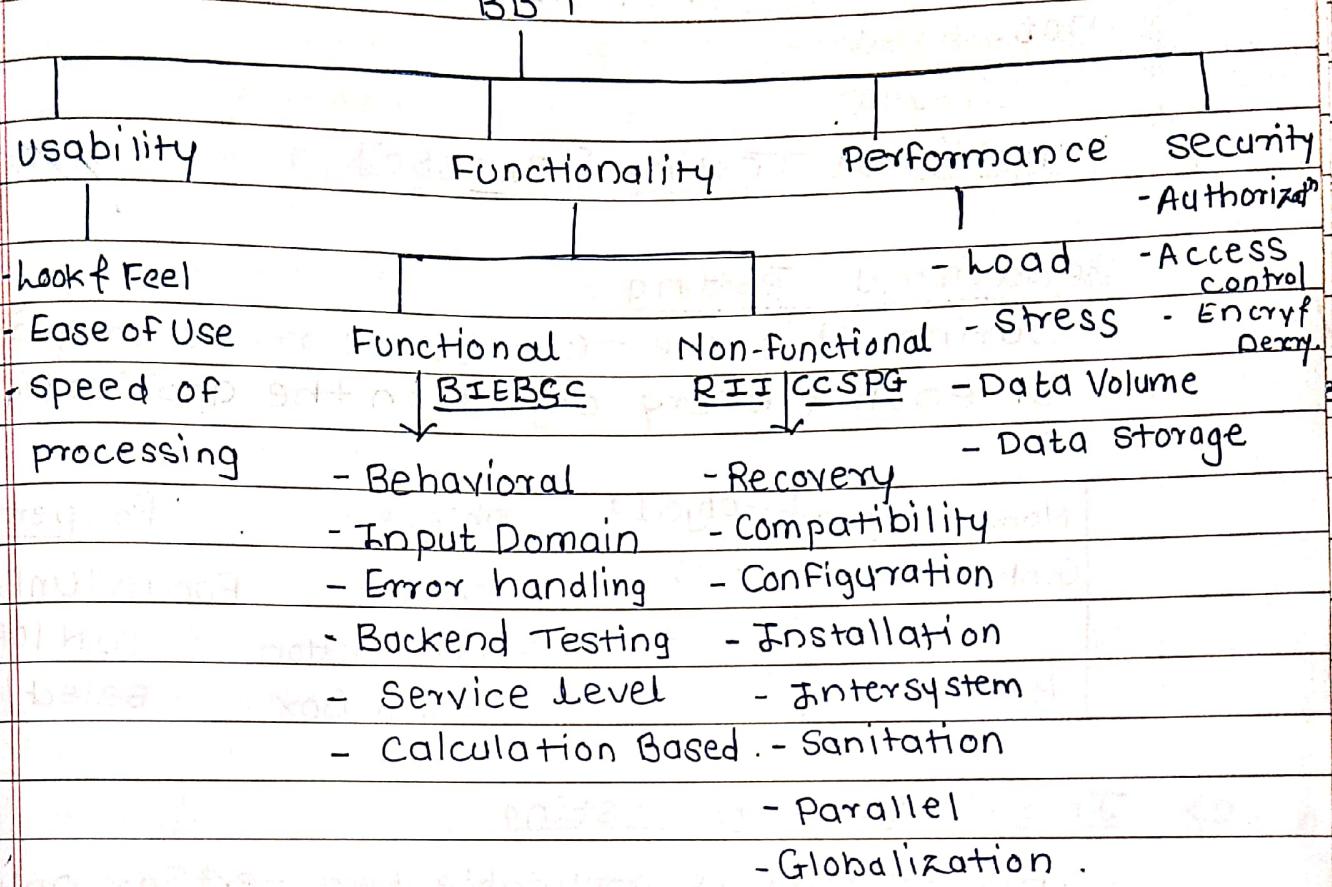


- During this approach submodules are ready and main module is under construction.
- In this we test the integration between main module & submodule.
- To test we need temporary program we called it as "Drivers".
- Drivers are written by developers in XML.

3) Sandwich or Bidirectional Approach

- Combination of top-down and Bottom-Up.
- Module which we are testing sometimes it will work as main module & sometimes as sub module.
- When it is working as main module at that time submodules are under construction then top-down will be used.
- When it is working as sub module at that time main module is under construction then bottom-up will be used.

* Black Box Testing Techniques :



17/5/19 - Screenshots of Testing : Proofs

Usability Testing

- This testing is related with GUI part of the end users.

1) Look f Feel : pleasant or attractive for the end users.

2) Ease of Use : User friendly

Application or screen which are developed, it should be user friendly.

3) Speed of processing : App. should take minimum no. of steps to perform task ~~are~~ operation.

* Functionality Testing

- During this, test engineer has to validate whether we fulfill customer requirement or not.

1) Functional Testing [BIEBSC]

1) Behavioral Testing

During this, we test behavior or property of each & every object in the application.

Name: <input type="text"/>	Object ¹	Object	Property
Gender: ON <input checked="" type="radio"/>	2	TextBox	Focus / UnFocus
OFF <input type="radio"/>	3	Radio Button	ON / OFF
<input type="submit"/> <input type="button" value="Cancel"/>		Combo Box	Select / Unselect

2) Input Domain Testing

- This testing is applicable for TextBox only
- We validate test Data by 2 Factors :
 - 1) Length (Size)
 - 2) Data type.

Two methods :

1) BVA [Boundary Value Analysis]

2) ECP [Equivalence class Partition]

Example : Customer Requirement

There is one Password textbox which contains 3-5 Uppercase characters only. Find BVA & ECP.

- 1) BVA -
- 1> min - 3 - valid
 - 2> min+1 - 4 - Invalid
 - 3> min-1 - 2 - Invalid
 - 4> max - 65 - valid
 - 5> max+1 - 66 - Invalid
 - 6> max-1 - 64 - valid

2) ECP : Blank spaces are not part of special symbol.

A-Z - valid

a-z - Invalid

0-9 - Invalid

(@, *, #, ...) - Invalid

Blank space - Invalid

2. There is one password textbox. It should contain 4-8 Alphanumeric lower case characters.

BVA (4-8)	ECP
min = 4 ✓	
min+1 = 5 ✓	Valid
min-1 = 3 ✗	(a-z)
max = 8 ✓	(0-9)
max+1 = 9 ✗	(*, #, @...)
max-1 = 7 ✓	Blank spaces

3) Error Handling Testing / Exception Handling

- During this, T.E has to validate or He has to test with valid as well as invalid Test Data.
- For Invalid Test Data application should Show error or exception that we are going to check.

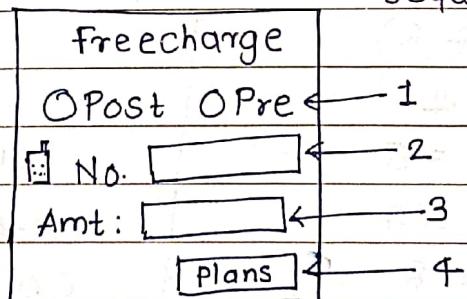
4) Backend Testing / Database Testing

- Validating the data into database.
- eg. Gmail signup Details and then login.
- crud : Create, update, retrieve, delete.

5) Service - level Testing

- Functionality Should be tested in a sequential manner or there should be proper order of functionality.

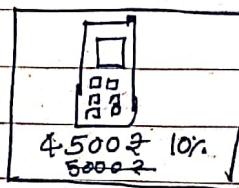
Sequence of Test



6) Calculation Based Testing

- To test arithmetic operation like:
ADD, subtract, MUL, DIV, %.

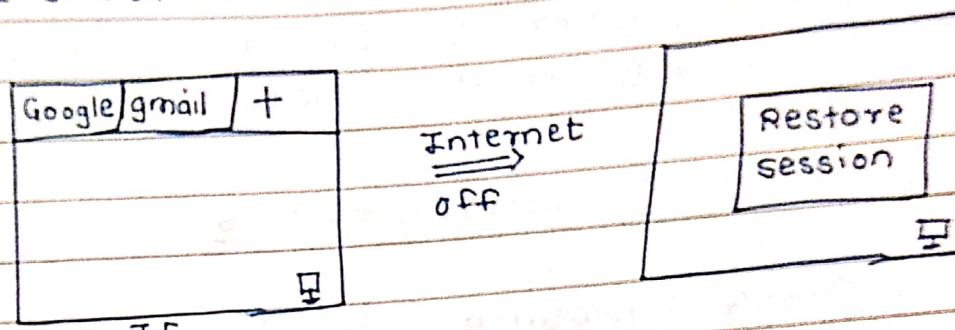
e.g. Ordering App



* Non-Functional Testing [RCIIISPG]

1) Recovery Testing / Reliability Testing

During this, T.E has to validate whether our application can cover from abnormal situation to normal situation.



- It will be performed only for abnormal conditions

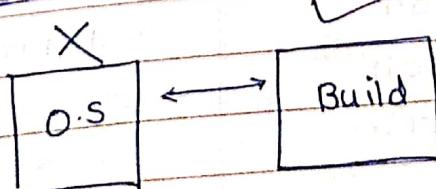
* 2) Compatibility Testing / Portability Testing

During this, T.E is going to check whether our application supports customer expected platforms (O.S & Browsers)

O.S Related Compatibility Testing

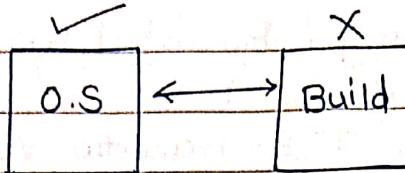


1) Forward Testing



- In this case, O.S is not proper, But build is proper
- We get less defect in this case.

2) Backward Testing.



- In this, OS is working properly and Build is not proper. So getting maximum defects is possible in this case.

Browser Compatibility Testing

1) Cross Browsing

- It is performed on different browsers like internet explorer, Google chrome etc.

- During this, we have to check whether our application supports different browsers.

T.E Check:

- URL Validation
- Link Validation
- GUI Validation
- Tab Validation
- Page Validation

2) Version Browser

- Same browser with different versions.

- During this, we have to check whether our application supports different versions of same browsers.

- URL, Link, GUI, Tab, Page Validation.
- It uses Virtual Machine.
- Start → Remote Desktop.

3) Configuration Testing

- It is hardware Related.
- During this, T.E will validate whether our application supports different hardware devices like printer, mouse, keyboard etc.

4) Installation Testing

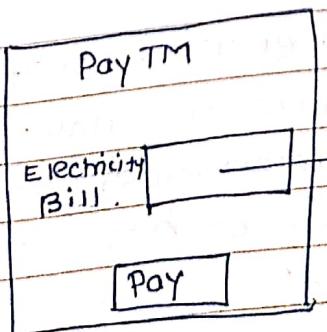
steps : T.E has to validate following things :

1. All the setup files should be available with us.
2. while Installation of our application, it should show the progress of installing application, time progress.
3. After installation we have to check how much space application has occupied and how much remaining space (Memory).
4. Easy Uninstallation .

5) Intersystem Testing

- During this , T.E validates whether our existing application (co existence of our application) with another application to share data of resources .

eg .



For this electricity Bill amount, PayTM will connect to MSEB application.

6) Sanitation Testing

- During this, TE has to identify extra features which are developed by Developers But not specified by the customer in SRS.

7) Parallel Testing / Reconciliation / Recon

- During this, T.E will validate new version of application with old version (Production) for data comparison.

We do it using tools:

- Recon

- Beyond Compare

*8) Globalization Testing / Language Compatibility

- In this the application will be validated For different language.
- T.E is going to test application to check it works for different languages.
- Google Translator is used to check for diff. language.

* Performance Testing

- This is advance level Testing technique.
- During this, T.E have to make sure that whether our application performs as per customer expectation or not.

Tools Used :

- LoadRunner
- Jmeter

Reasons of performance issue:

- 1) No. of users using application at the same time.
- 2) Wrong logic By Developers.

- Performance testing is not possible to do manually when no. of users are more. So it is done using Tools.

- Wrong logic By Developers:

Stopwatch will be used to calculate Performance (loading) time while doing it manually.

For e.g.

Prod

RedBus

loading.....
500 Records

1:13min

QA

RedBus

loading....
500 Records

45sec

Time duration changes for same no. of records then performance is said to be Better.

1) Load Testing

During this application will be tested by providing load / data in incremental order to check how much max. load our app. can handle.

1U → 2U → 15U → 20U crash
Load

2) Stress Testing

Application is tested by giving maximum load.

4) Data Storage Testing

Testing data storing capacity (size) of application.

size: KB, TB, GB.

3) Data Volume Testing

Volume - Quantity

Testing how much Records app. can store.

eg. 2000, 1000
Records

* Security Testing :

- Application provides access as per role of user. So testing the access of application.
- This testing is mostly done for Banking system.
- This is complex type of testing and advance level.

1) Authorization

- In this, T.E is going to validate user is authorized or not for the application.

e.g. login : 4 cases T.E will check.

	UName	Password	Output
	Valid	Valid	Pass
	Valid	Invalid	Fail
	Invalid	Valid	Fail
	Invalid	Invalid	Fail

2) Access Control

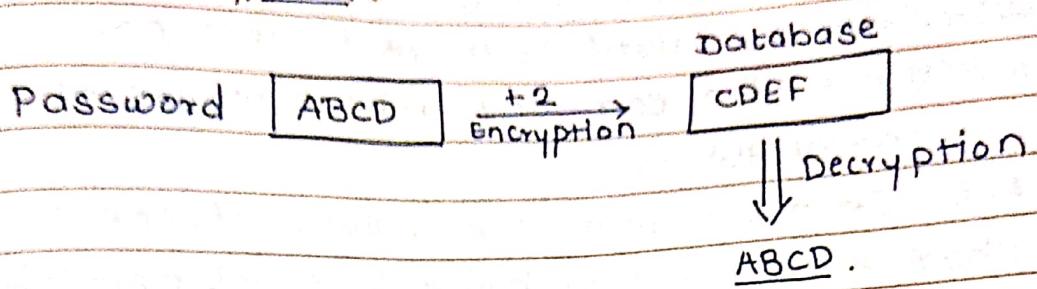
Precondition : Authorized User

- Then T.E will perform Access control & T.E will check this user have the permission to perform specific task.
- The Permission Table in database for permission & access permission for users.

3) Encryption & Decryption

- Done by Backend Developers (SQL Developer)
- Password or imp. data will be stored in encrypted format (Cipher text).

- Converting encrypted data to plain text is called as Decryption.



* Defect leakage

When a Bug (defect) is found by customer/end user, missed by the testing team to detect, while testing software.

* Defect Release

Bug Release - Release of the Software with the known bugs is called Bug Release.

When SW or an application is handed over to the testing team knowing that the defect is present in a release. During this the priority & severity of bug is low.

* Bug (Defect) Age:

- Bug (Defect) Age is the difference in time between the date a defect detected (confirmed & assigned) & the defect was fixed (verified & closed).

Bug Age = Bug fix Date - Bug detection Date.

* Different Environments In Testing

client



SRS

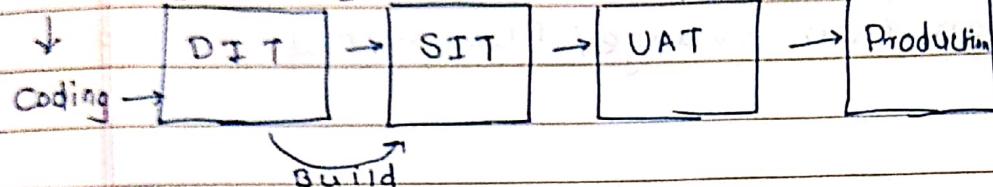
↓ [Developers]

SRS [Unit test] [T.E]

↓ Client + T.E

- End User

- Live.



1) DIT : Development Independant Testing

2) SIT : System Integration Testing

3) UAT : User Acceptance Testing / Customer Accept. Testing

4) Prod : Production

* SIT :

Initial Build :- Basic or First Build

1) Sanity Testing, 2) Functional, 3) Regression.
 Initial Build CR Build
 Modified Build Testing overall App for change

* When we can say testing is completed ?

- once all the designed test cases are executed.
- All the raised defects are fixed by developers & in retesting the defect is not occurring again then we "close" the defect & we can say application is stable.
- Tester will do "sign off" for closing testing.

* UAT :-

- SIT & UAT Testing teams will be different.
- In UAT, senior testers will be involved.
- Time will be less (3-4 hours) for UAT.
- Testing & test team will be less in size.
- SIT test cases will be executed by UAT team.
- Focus on New change / Functional testing

24/5/19

* Error :- Mistakes in the code.. (WBT).

- Developers will find errors.
- DIT environment.

Defect :- SIT environment.

- Found by SIT testers.
- Defect Report - Mail (Environment)
- Defect log - Tool through. (Functional regression).
- Retest.

Bug :- If defect is accepted by developers then it is called as Bug

Defect leakage :- Defects missed by SIT & found by UAT team.

- UAT team will inform defects to developers / Developers will fix it. Then UAT team retest it then it will go to production.
- These defects will be only inform to SIT but SIT will not be part of this defect fixing.

Production issue: The defects found at production environment. These defects are missed by DIT, SIT & UAT.

Hot fix: Production issue fixing by developer & CCB.

27/5/19

- When any production issue occurs then client is going to take penalty from organization.

Impact Analysis:

- When any production issue comes, there will be one team: CCB (Change Control Board).
CCB → BA, Development lead, Test lead.
- CCB Team will do first Impact Analysis
- Impact analysis is process of analysing product issue for criticality, priority, severity etc of impact of issue on the client's Business.

Escalation: kind of notice / Memo / Warning

* Testing Terminology; Scenario Based Questions

1) Monkey Testing [Speed Testing]

- less time and more no. of test cases to be executed.

2) Test case → Priority [Importance]

Defects → Priority + severity

Test Case Design : Priority will be given by Priority the T.E who have written the test case.

- In this testing, first T.E will start executing high priority test cases then medium priority and if time remains then low priority test cases will be executed.

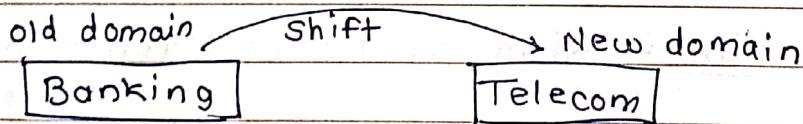
eg. 100 Test cases are there to execute but time is of only 2 days and as T.E can execute maximum 60-70 test cases then we will go for Monkey testing.

100 Test cases - T.E — 2 days.

Productivity: 20-25 per day, so out of 100 T.E can execute 60-70 test cases, still 40-30 test cases will remain without executing so in this scenario we will go for Monkey testing.

27 Exploratory Testing

eg.



- We have Test Data (test cases) but we are new in the domain. So T.E will follow test cases (steps) to understand the application or to explore the application.

This is called as exploratory testing.

- Mostly used when application in production and maintenance phase.

3. Ad-hoc Testing

- When T.E have knowledge about domain but we don't have test data (test cases) then using previous experience/knowledge T.E will write Test cases for application.

28/5/19

Build : MSI

Mon: Initial Build : 40 releases are already in production.
G-CR. 41.0 → current release & version 0.

- Sanity Testing will be performed for build stability and to check basic & core functionalities of application.
- Smoke test is performed after sanity test on initial build.

Tue: Incremental Build : 41.1 - 2-CR Deployed.

- Functional testing for 2 CR - change of Regression testing for checking impact of new change on old functionalities.
- If any defect is found, T.E is going to log the defect.

Wed Modified Build : 41.2 - 2 new CR : Total : 4 CR

- Defect fixed : Retest to check bug is fixed or not.
- 2 CR : Functional test.
- Regression Test : To check impact of new CR if to test fixed bug has impacted on old function

Thur 41.3 → 2 CR: Total : 6CR + Bug Fix.

- Incremental versions of Build : consolidated Build
- For ~~1~~ CR - Functional Test
- Bug fix - Retest
- Impact CR + Bug fix - Regression test

In this, if defect found ~~is~~ while doing regression test. T.E will log the defect :

Fri: 41.4 - Final Build : Retest + Regression Test.

- Bug Fixes : Retesting + Regression test for impact of bug fixes on other functionalities.
- No functional test will be performed. Bcz there will be no change deployed. Only Bugs are fixed.

Master Build : 41.5

- All bugs are fixed.
- T.E has done sign off then this version will be given to VAT.
- If VAT found some defect in this build then they will log it to developer and again new version will be release 41.6 and this build will be forwarded to production (without Bugs).

* Retesting : When defect is fixed then only T.E will do this.

After retesting, Regression test will be performed.

* Sanity Testing / Build verification Testing

- After receiving Initial Build from the development team, T.E has to check Build is stable or not (Build Stability) as well as core or Basic functionalities of application.
- As well as we check : - URL Validation
 - link Validation
 - Page Navigation
 - Tab Validation
 - GUI Validation
- Mostly we perform this testing on Monday (2 week), on the first day of test case execution .
- 2-4 hours are required / needed for this testing.
- Kind of defects :
 - Environmental issues : Run-time error, services are down, exceptions, page not found / loading.

* Do you log the defects in sanity testing ?

- No, Environmental issues are reported through mail & send ~~to~~ the screenshot.

- In this test, T.E will not troubleshoot the issue.

* Smoke testing

- Initial Build
- Sanity test + troubleshooting .
- Troubleshooting the defect :
 - check all the services are up (running) on server machine (services.msc)
 - If all ~~are~~ services are up then check for logs.

logs:-

- Server machine - particular path
 - extension - .log
 - events are stored: Date, time
- no - to I: Information E: Exception.
- IF there is exception for your (T.E) event then check into log file, take screenshot and report it to developers.

Package Validation : Package contains objects

- Sanity testing + Package validation.

Build

→ attachments → Package.

- IF o T.E gets defect in object then T.E will identify the package where the issue is and then report it to developer.

* Do we write test cases for sanity testing?

- No, We are going to execute test cases of previous release test cases.
- BCA Basic functionalities are already deployed in previous releases.

* To get know more about new project then refer Sanity test cases.

sanity Testing

- Build verification test
- Report the defect

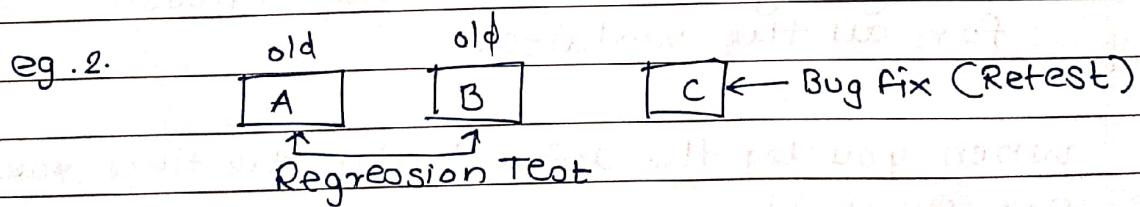
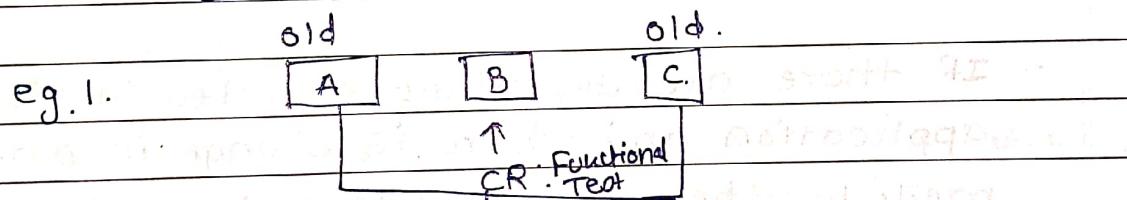
Smoke Testing

- sanity + troubleshooting
 - sanity + package validation.
- 1 - Check services
 - 2 - check logs
 - 3 - Report

* Regression Testing

- When there is a new change or bug fix in the application, we have to check, due to this new change or bug fix, existing functionalities should work as it is as it was working previously.
- We test bcz of new change or bug, existing functionalities should not be impacted.

code change → Regression Test



- Two Scenarios when we perform Regression :
 - 1> When new change in application
 - 2> Bug fixes.

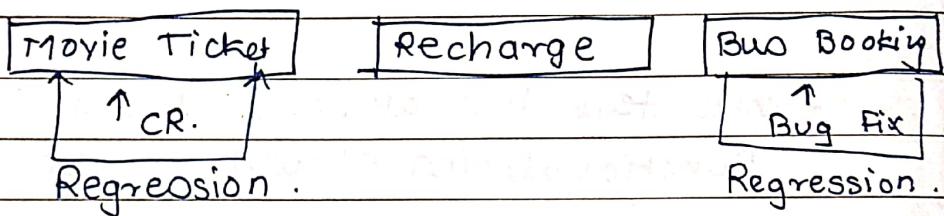
* There are 2 types of Regression Testing :

1) Regional Regression 2) Full Regression.

1) Regional Regression

- If there are independent modules in the application and there is change in one module then bcz of independency we (T.E) are going to perform Regression test only on particular module. and it is called as Regional Regression.

eg.



2) Full Regression

- If there are dependant modules in the application and there is change in one module then bcz of dependency , we (T.E) are going to perform Full Regression test for all the modules.

* When you log the defect & till the time you fix Dev. fix the bug, what T.E will do ?

- Execute other independent test cases.

For Blocker defect / show stopper defect :

- we (T.E) can not do further testing.

* How many times we perform Regression Testing?

- In each testing life cycle

2: For every release / sprint; 1: On consolidated Build times.

2. Master Build.

- During day by day on consolidated build.

- Before Build moves to UAT, on Master Build.

31/5/19

Test Suit / Test Batch / Test set / Test Pack

- Clubing of all test cases which we are going to cover in Regression:
- Previous failed Test cases
- Critical Test cases : High & medium Priority.
- If time is still remaining then we are going to cover all independant test cases.

*

Retesting

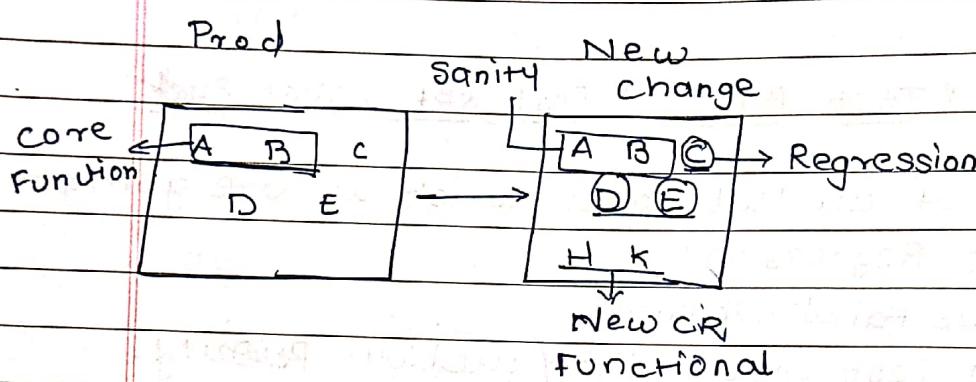
- During this, T.E is going to check whether Bug is fixed or not raised by T.E.
- Modified Build for Bug fix.
- T.E is going to test particular functionality where bug was occurred.

Two cases / scenarios / possibilities after retest :

- Bug is fixed if functionality is working as expected.
so T.E will close bug.
- Bug / Defect is not fixed or may occur again.
so T.E will Reopen Defect.

- * which is perform first: Regression or Retest
- IF Bug is fixed then Retest and then regression testing.
 - Retest is performed only for Defect fix.
 - Regression is performed when change is deployed in build and Bug is fixed.

Example:

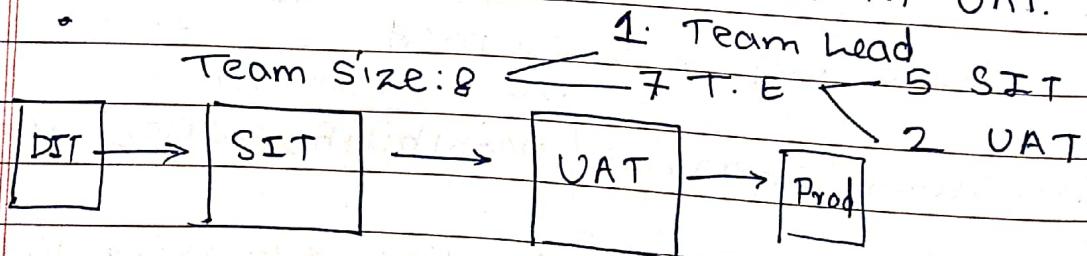


1/6/19

User Acceptance Testing [UAT]

Customer Acceptance Testing

- After Completion of SIT, Organization is going to follow VAT to receive feedback from client / customer.
- Customer / Client is involved in UAT.



2 week sprint : 1 week Testing.
 5 days :- 4 days - SIT
 1 day - VAT

* Traditional Project: Dev + Test + Support = 1 organization
 SIT UAT f Maintenance.

SIT

- Client is not involve.
- There is more concentration on Database (Backend). bcz time is more.
- It is core testing & mandatory.
 - Team size & duration: More
 - During SIT, sanity, function, Regression testing & retest is performed
 - SIT testers will not be part of VAT.

UAT

- client involvement.
- There is more concentrat^h on UI (Front end) bcz of less time.
- It is performed for traditional projects.
 - Team size & duration: less
 - During this, T.E will focus on new changes or functional Testing.
 - Seniors testers are mostly involved.

— OK —

- There are two types of VAT:
 1. Alpha Testing (A)
 2. Beta testing (B)

- Applicable for application. i.e Service Based company
- Performed within organization.
- It is done with the presence of Dev. & testers.
- It is performed in controlled environment.
-

- Applicable for products. i.e. for product based comp.
- Performed by End users at their own place of PC's.
- Without Dev. & testers.
- It is performed in uncontrolled environment.

* Entry and Exit criteria

Entry Criteria :

- These are the condition or the set of conditions which should exist or be met in order to start a process.
- Conditions :
 - Availability of test environment supporting necessary HW, SW, NW configuration, settings & tools for the purpose of test execution.
 - Availability of proper & adequate test data.
 - Presence of proper testable data.
 - Testers are trained & necessary resources are available.
 - Requirements should be clearly defined & approved.
 - Test Design & documentation plan is ready.

Exit Criteria

- Exit criteria is the condition or set of conditions that decides the completion or termination of the testing task as well as the accomplishment of the targets & goals. It is also defined & outlined during the test planning phase.

Following are some of the possible exit criteria which may be used to mark the end of the testing activities:

- All test cases have been executed.
- Sufficient coverage of the requirements of functionalities has been achieved.
- All high severity or top priority bug has been fixed.
- High risk identified area has been taken up & tested.
- Budget gets depleted.
- Deadlines reached.

* Showstopper / Blocker Defect

- Due Defect due to which we can't proceed Testing.
- Testing Halt.

e.g. GMAIL login not working.

* End to End Testing

- It is defined as a type software testing that not only validates the software system under test but also checks its integration with external interfaces.
Hence the name "End-to-End".
- Purpose - To exercise a complete production like scenario.

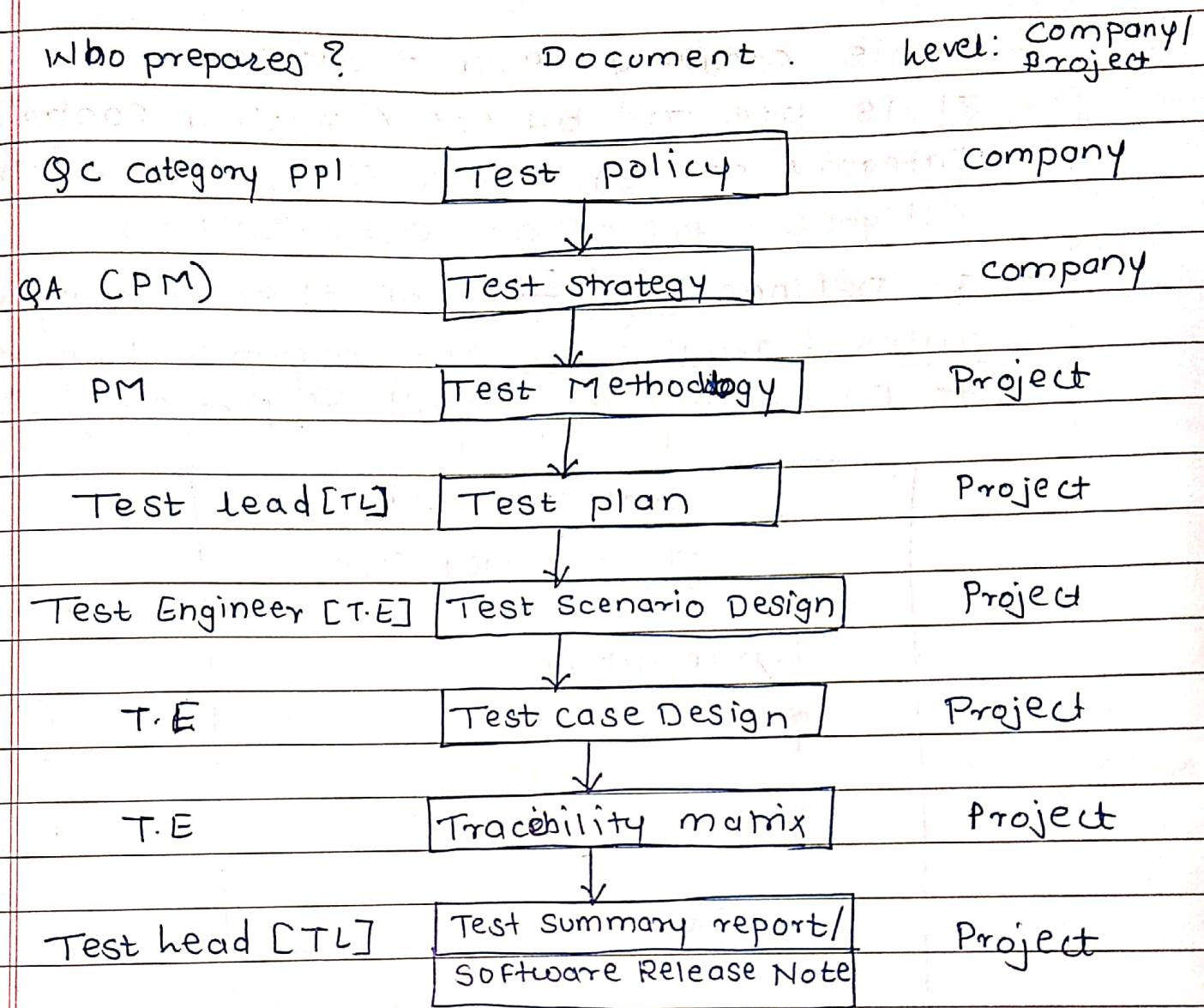
- End to End testing is actually executed after functional & system testing.
- It uses actual production like data & test environment to simulate real-time settings.
- It is called chain Testing.

Why End-to-End Testing?

- Modern Applications are complex & are interconnected with multiple subsystems.
- A subsystem may be different from the current system or may be owned by another organization.
- If anyone of the subsystem fails, the whole application could collapse. This is a major risk & can be avoided by End-to-End testing.
- End-to-End testing verifies the complete system flow.
- It increases Test coverage of various sub-systems. It helps detect issues with sub-systems & increases confidence in the overall application.

Depth Part

Test Documentation



Test plan :- Agile test plan.

- Estimation (Duration, resource / job allocation, start & end time).
- Approved by PM & client.

1) Test policy

- It is company level document.
- It is prepared by QC (Quality Control category people) means CEO, MD, Vice MD, CHM (Higher mgmt of any organization)
- It defines Objectives of the company or rules & regulations that organization has to follow for any kind of project.

Syntex pvt. ltd

Test Policy Doc.

1. Definition :-

Q

2. Process :-

3. Standard :-

4. Measurement :-

Salok
CEO

1. Definition: Verification + Validation.

Develop. Testing

2. Process: SDLC Model + Testing tasks. (Testing types).

3. Standard :

No. of employees → CMM level.

e.g. 1 LOC → CMM level 5.

std: 1250 LOC / 1 defect.

2. SQL queries.

3. 10 FP / 1 defect

FP: Functional point.

↳ SQL query

↳ web page

↳ objects.

4. Measurement :

- level of CMM

CMM : Capability Maturity Model (level 1-5)

- ISO will decide the CMM level of organization.

2) Test strategy

- Company level document.
- Prepared by QA (Project Manager).
- During this, PM is going to use different approaches to work on any kind project.
- This mostly included the approaches for different projects.

There are different approaches: [7-8 Preparation]

- 1) Scope of objective; What kind of project it is & what kind of testing we will apply?
e.g. Banking project → Security & performance testing

[T.E] RTM : Requirement Traceability Matrix

classmate
Date _____
Page _____

2. Business Issue:

- Cost Analysis
- How much money will be spent on Dev.?
- How much money will be spent on Testing?

* 3. TRM: Test Responsibility Matrix

- Prepared By PM.
- It's a mapping between Development stages & testing issues/factors.

e.g. Banking

6 Development Stages

	Test Issues	Dev. Stage	Info Gathering	Analysis	Design	Code	Test	Maint.
15- Testing issues	1. Authorization	X	X	✓	✓	✓	✓	✓
	2. Access	X	X	✓	X	✓	✓	✓
	3. Audit Trial.	✓	X	X	✓	✓	X	✓
	4. continuity process	X	✓	✓	✓	✓	✓	✓
	5. Coupling	✓	✓	✓	X	X	✓	✓
	6. Correctness	✓				✓		
	7. File Integrity			✓				
	8. File Reliability	✓			✓			
	9. Ease of USE			✓				
	10. Ease of process	✓			✓			
	11. Performance			✓				
	12. Portability	✓				✓		
	13. Service level							
	14. Maintainability	✓	✓	✓		✓		
	15. Maintainance				✓			

4. Test Deliverables: Required task to be completed before start testing : SRS Analysis, Test case design, RTM & then test case execution.
5. Roles & Responsibility: Name of the job who is going to perform it is decided by PM.
6. Communication status Reporting: PM has to check whether there is proper communication between all the teams.
7. Defect logging & Tracking: Developer + Tester Team. PM will keep track of defect.
8. Use of Automation : Test Automation. Deciding whether automation tool will be used or not.
9. Risk of Mitigation : Risk : Problems, Mitigation / Solution.

Risk : Project Related

- lack of Domain knowledge
- lack of Test data
- DB connectivity
- Budget

Non-Project related - Hardware related, network related.

- 3) Test Methodology : Finalize TRM. for specific project.

- Prepared by Project Manager.
- It is project level document.

TRM: Testing Issues/Factors

A³

- 1) Authorization : To check whether user is valid or not.
- 2) Access Control : permissions for valid user.
- 3) Audit Trail : eg. mini-statement in Banking project for generating it for multiple transactions.

C³

- 4) Continuity of Processing : Maintaining communication between two processes via XML.
- 5) Correctness :
 - Functional Testing
 - To check correctness & completeness of application for customer requirement.
 - Mostly, for all application.
- 6) Coupling : Inter system testing.

E²

- 7) Ease of Use
 - Usability testing
 - App. should be easy to use & friendly.

- 8) Ease to operate;

App should be easy to operate.

F²

9) File integrity : Ance system testing creating Backup files.

10) File Reliability (Recovery) : Recovery Testing.

P²

11) Performance : Performance Testing

12) Portability : Compatibility Testing

S

13) Service level :- service level Testing.

M²

14) Maintenance : Enhancement / service .

15) Methodology

* Test Methodology

- Prepared by PM .

- Project level Document .

- It is specific for project .

- The objective of this document is to prepare Finalize TRM document.
- * On what basis, PM is going to prepare the Finalize TRM?
 - Project type
 - Project Requirement

Steps to prepare finalize TRM:

1. PM is going to acquire Test Strategy document.

5 Types of Project:

1) Traditional - Dev + Test from Scratch -

2) off the shelf - Only testing -

3) Maintainance - Enhancement + Bug fixes -

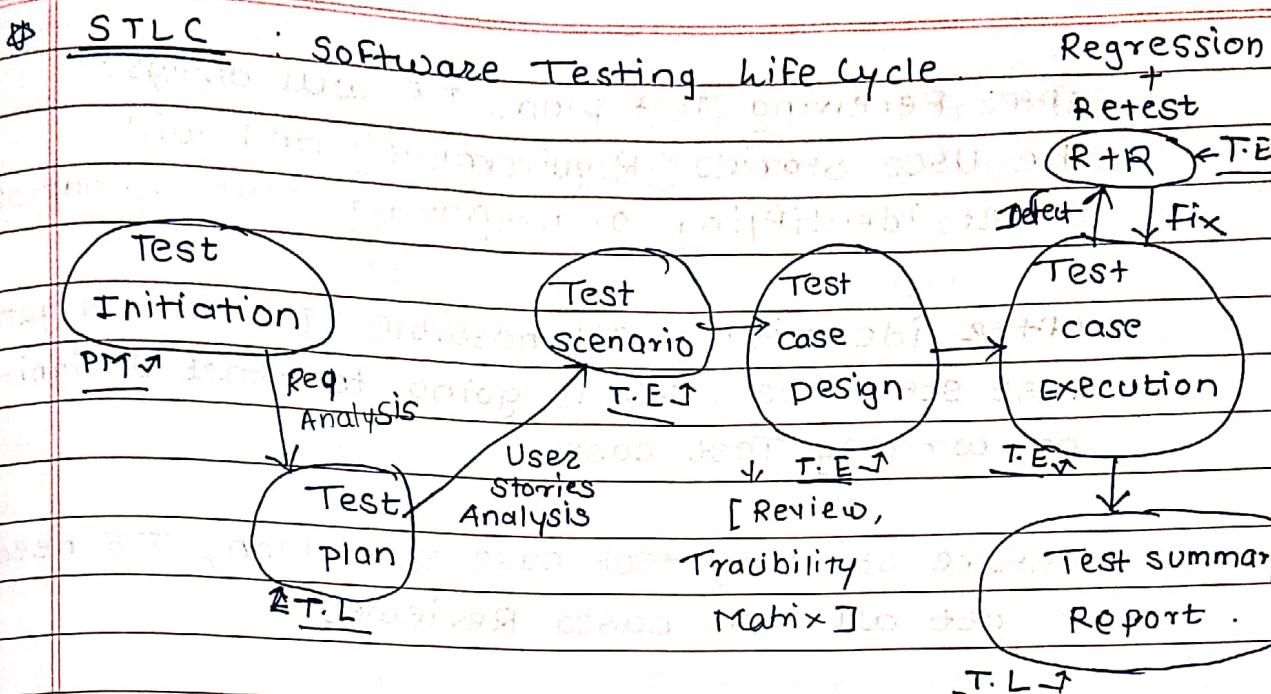
2. According to project type, PM is going to remove some columns (Development stages).

3. According to Project requirement, PM is going to remove some rows (Testing issues).

4 - For Future enhancement, PM can / may add ^{some /} new testing issue (Row) now.

5. To avoid future risk, PM may remove some testing issue (Row).

PM - This Finalize TRM will be pass to Testing head as input to next document i.e. Test plan.



1) Test Initiation:

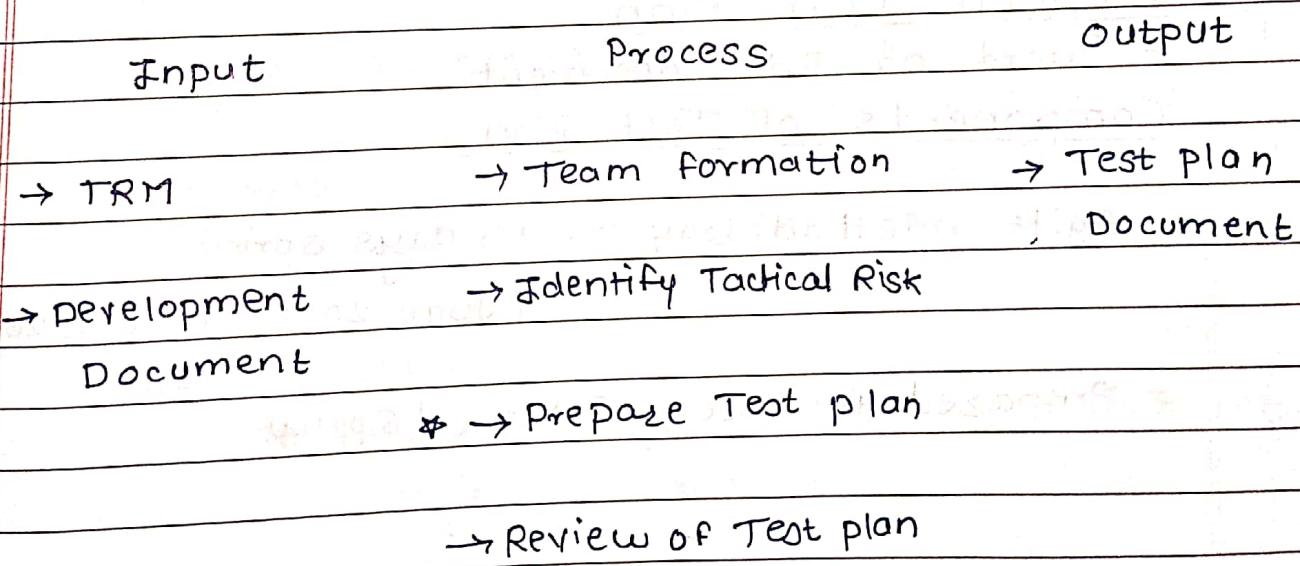
- PM (Project Manager)
- Scope, Requirement, Risk.
- Finalized TRM (Test Plan) and Test initiation document.
- In my organization testing process starts with Test initiation stage where my PM basically concentrates on Scope of Project, Requirement of Project and Risk involved in Project.
- In this stage, PM also prepares Finalised TRM document and pass it to Test head [T.L] for further process.
- After test initiation, the test plan is prepared by T.L. and T.L mainly focuses on things like Job allocation [Task Allocation], Resource Allocation and Estimation [Start & end time of each & every Testing activity]. Test plan is done for every sprint.
- This Test plan will be Reviewed by PM + client & final Test plan is going to pass to all testing team Members.

- After Receiving Test plan, T.E will analyse the User stories (Requirement) and will start identifying or preparing test scenarios.
- After identifying all possible [positive + Negative] test scenarios, T.E is going to start designing or writing Test cases.
- Before starting test case execution, T.E need to get all test cases Reviewed.
- In the Review of test cases, there might be some changes [priority, missing test cases] & will be commented by review members.
- After Test case Design, T.E will prepare Requirement traceability Matrix [RTM] for finding out any missing test case.
- The Test cases found in RTM will be included in final Test case Design.
- This final Test cases will be used by T.E for further Test Execution process where T.E will perform Sanity + Smoke Testing, functional Testing, Regression Test & for defects Retesting.
- once all the Designed Test cases are executed & All the raised defects are verified & closed then T.E will Stop Testing.

- After completion of Test case execution the finally summary of all the testing process will be prepared by T.L and it is called as Test Summary Report or Software Release Note.

* Test Plan Document

- It is project level document if prepared by Test Lead.
- During this T.L is going to focus on 3 factors:
 - 1) Job allocation [What to do]
 - 2) Resource Allocation [Who is going to do]
 - 3) Estimation [Start end day of Testing activities]



1) Team Formation

- This is first stage while preparation of Test plan document.

- During this T-L has to concentrate on team formation & will consider following:
 - * Availability of Resources [Human Resource]
 - Availability of Environment [H/w & S/w]
 - Estimation [start date & end date]

2) Identify Tactical Risk Risks

- 1) Lack of Domain Knowledge
- 2) Lack of Test data
- 3) Lack of Resources
- 4) Build Stability
- 5) Lack of Budget

3) Prepare Test Plan

- word or PDF document

Components of Test Plan

Agile Methodology :- 15 days sprint

1 June 2019 - 15 June 2019

- Prepared for every Release / Sprint.

Components of Test Plan

- ① Test plan Id : Unique Identity for Test plan.
eg. 2019.6.15_PayTM Release #
R1_PayTM Release.
- ② Introduction : This Release is planned for 6 functional changes (JIRA) & 4 Regression testing.
- ③ Iteration : What functional changes will be performed in this Release ?
- ④ Test items : Modules & component which we are going to test during this Release.
- ⑤ Features to be Tested : Out of 15 CR : only 10 features will be tested bcz of risks.
- ⑥ Features Not be tested : Out of 15 CR : 5 features will not be tested bcz of risks .
- ⑦ Test Approach : SRS Analysis, Identifying Scenario, Test case design, Traceability matrix, Test case execution .
- ⑧ Schedule : Estimation .

Task	Start Date	End Date	Environment
SRS Analysis	2 JUNE	3 JUNE	-
Test scenario	4 JUNE	5 JUNE	-
Test case Design	5 JUNE	8 JUNE	-
[SIT] Test case exec.	9 JUNE	13 JUNE	SIT
UAT	14 JUNE	14 JUNE	UAT

(9) Responsibilities:

Resource Name	Responsibility / Task

Approved By :

(10) Risk - Schedule, Technical, Management, Personnel, Requirements .

4) Test plan Review

- Once the T.L prepared test plan document then he is going to pass this to PM & Client for Review.
- After approval of these people, the final Test plan will be shared with Testing Team.

* Agile Test Plan / Complete Agile.

[1 June - 30 June 2019] - Sprint Duration .

4 weeks.

Week 1 - 1 J - 5 June

Week 2 - 8 - 12 June

Week 3 - 15-19 June

Week 4 - 22 June - 27 June .

Week 1 : 1 June 2019 - 5 June 2019

1 June 2019 - Sprint Planning Meeting

2 June 2019 }

User Stories Analysis

3 June 2019 }

4 June 2019 }
5 June 2019 } Test scenario identification/Design

Week 2: 8 June - 12 June 2019

8 June - 12 June 2019 : Test case Design

- Iteration Test case Design (Functional).
 - Review of Test cases. (2-3 Hours)
 - If Regression test cases are not available then T.E is going to design Regression Test cases.
 - Review of Regression Test case Design.
- 12 June 2019 - Traceability matrix for missing Test cases from iteration of Regression test cases.

Week 3 : 15 June - 19 June 2019

15 June 2019

- On Monday Morning, T.E will get initial Build.
- Sanity Testing will be performed on initial Build.

16 June 2019 - 18 June 2019

- Test case execution for functional Testing, Regression and Retesting.

19 June 2019 : Final Regression on Master Build.

Week 4 : 22 June - 27 June

- Build will be move to VAT.

- UAT will be performed.
- UAT is happen in Sprint Review Meeting.
- Client / Stakeholder will give feedback regarding Sprint.
- Then Retrospective meeting will be conducted.
- Then Test Summary Report will be prepared by Test lead.

* On which phase you're currently working in your Sprint?

- As my organization follows 1 Month or 15 days sprint, accordingly we are currently doing SIT. for this sprint.

For 1 Month - 3rd week

15 Days - 2nd week.

SRS	Use Cases	Test Scenario	Test cases
- Req. Doc.	BA	T.E	T.E
- BA	IIP, OIP & Process	- Defines the functionality of Application	- Navigational statement to define functionality
		- It means "What to test"	- It means "How to test"
		- One Test Scenario consist multiple Test cases	- 1 Test case consists of multiple steps
		- These are derived from Use Cases	- These are derived from the Test Scenario.

eg login Page

SRS	Use cases	Test Scenario	Test case															
login Page	IUP: UName & Password	verifies the UName & Password	- Navigate steps to test the functionality.															
- Username																		
=	Process: Click on login	for valid UName & Password.																
- Password	OIP: Navigate to Homepage																	
=																		
		<table border="1"> <thead> <tr> <th>UName</th> <th>Pass</th> <th>OIP</th> </tr> </thead> <tbody> <tr> <td>valid</td> <td>valid</td> <td>✓</td> </tr> <tr> <td>invalid</td> <td>valid</td> <td>✗</td> </tr> <tr> <td>valid</td> <td>invalid</td> <td>✗</td> </tr> <tr> <td>invalid</td> <td>invalid</td> <td>✗</td> </tr> </tbody> </table>	UName	Pass	OIP	valid	valid	✓	invalid	valid	✗	valid	invalid	✗	invalid	invalid	✗	
UName	Pass	OIP																
valid	valid	✓																
invalid	valid	✗																
valid	invalid	✗																
invalid	invalid	✗																

Test case :

Test case Name	Step Name	Design STEP	Expected	Actual	Priority
TC001 - Gmail - login - Valid UP.	Step 1.	Launch the URL	Should show login		
	Step 2.	Enter valid UName	Accept		High
	Step 3.	Enter Valid Password	Accept		
	Step 4.	Click on login Button	Should Navigate to Homepage		

Test Scenarios : Functionality

1) ATM Machine

- Withdraw
- Magnetic clip
- Physical presence (Human)

2) Human Being

- Walk, Talk, Read, Listen, Think,

3) Table / chair

- 4 legs (length should be same)
- Base for sitting
- Handle, Back support

Non-functional

- color, cushion

4) Keyboard :

- keys & its function.

-

5) Remote :

- On/Off
- channel & volume change
- Back
- Source

6) Pen

- Ink should be there (as fuel)
- Paper Color should not match
- It obeys gravitational Rule

Not for
pencil

- written line is not erasable.

- It obeys Capillary Tube mechanism.

7) Eye

- Watch, expression of emotions.
- Whenever body needs the rest then eyes will be closed automatically.

8) Fan

- Rotate (Anti clockwise)
- 3 plates

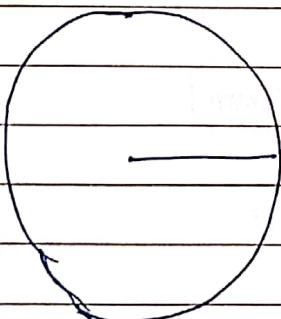
9) Lift

- Up / Down
- Floor number
- load / capacity
- lights / Fans
- security

10) White paper

- Use to write / Paint
- opposite ink will reflect.

10) Circle:



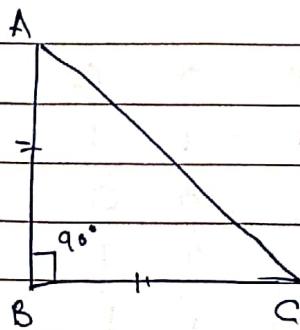
$$\text{Area} : \pi r^2$$

$$\text{Circumference} = 2\pi r$$

$$\text{diameter} = 2r$$

$$\text{Angle} : 360^\circ$$

11) Right Angle Triangle



$$\text{Area} = \frac{1}{2} \times AB \times BC$$

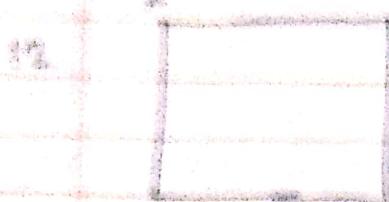
$$\angle B = 90^\circ$$

$$\angle A = 90^\circ$$

$$AC^2 = AB^2 + BC^2$$

$$\text{Circumference} = AB + BC + AC$$

Square



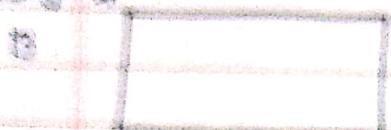
$$\angle A = \angle B = \angle C = \angle D = 90^\circ$$

Circumference : $4 \times AB$

$$AB = AC = BD = CD$$

$$D \text{ Area} = (AB)^2$$

Rectangle



$$AB = CD \quad AC = BD$$

$$\angle A = \angle C > \angle B = \angle D = 90^\circ$$

$$D \text{ Circumference} = 2AB + 2BC$$

Area : width \times length

Application :

Bus Booking App

Start City : Mum, Pune, Nagpur

Dest. City : Mum, Pune, Nagpur

Date of Journey : Today Tomm

Passenger
No. of Buses : upto 7

Search Bus

Conditions.

A

f

B or C

OK

A

B

C

0

0

0

0

1

1

0

1

0

1

0

0

A

B

C

1

1

1

1

0

0

0

1

1

1

0

1

HDFC Banking

<input type="checkbox"/> Download Statement	Select Account Type : <input checked="" type="checkbox"/> Saving <input type="checkbox"/> Current
	Selected Acc. No. : 4391 3232 7093
Period : <input type="checkbox"/> Mini Statement (Max 31/7/014) <input type="checkbox"/> Specific Period <input type="checkbox"/> Start Date <input type="checkbox"/> End Date	
Transaction Type: <input type="checkbox"/> Only Deposit <input type="checkbox"/> Only withdraw <input type="checkbox"/> Both	
Format :	.word .pdf .excel
<input type="button" value="Download"/>	

Description	Acctype	Acc. No.	Period	Type	format
Verify that user should able to download statement for different data or selected specific data. (Account type, AccNo, period)	Saving —l— —ll— —ll— —l— —l—	4391 —l— —ll— 4391 —l— —l—	mini Statement —l— —ll— Specific Period —l— —l— Specific period	Only Deposit —l— —ll— Only withdraw —l— Both —l— Only deposit	.word .pdf .excel .word .pdf .excel .word .pdf .excel .word .pdf .excel

MakeMyTrip Flight

Description	Journey Type	from City	To City	Dept.	Return	Travel Buses & class
Verify that user should able to search flights for diff. locations	Oneway	Mum.	Delhi	Today	NA	A/I/D/I & E/P/B
	→ ←	→ ←	→ ←	→ ←	→ ←	2/1/10 & Eco.
	→ ←	→ ←	→ ←	→ ←	→ ←	1/1/10 & Eco
	→ ←	→ ←	→ ←	→ ←	→ ←	11/11 & Eco
	→ ←	→ ←	→ ←	→ ←	→ ←	max 9/16/16 & Eco
Max: Price & Month 1 year						
Round Trip Delhi Bangalore Today Tomorrow 2 Buses.						

Multi City	Delhi Mumbai	Mumbai Chennai	Today Tom	Tomm 18 June	2- f Pre.
		Chennai Pune	18 June	19 June	
Max: 5 Cities					

PayTM Recharge Application

Desc.	Mobile No.	Operator	Circle	Amount
				Browse plan

HPALT		Test Case Design	
Test	Application	Test	Design
14 ✓	Stadio Application	3 Subject	1 TC0001 - Test case ID: N/A
Domain	Path of	4 Verify Step 1.	Stepname: Launch the login page Expected result: Browser should display URL: Gmail.com successfully
HPALT	HPALT	5 DESC.	Stepname: Enter Username Priority: High Design: S.N: Autodesk Max 9.0
Tool	Valid Username	6 Stepname	Stepname: Enter Priority: Medium Design: S.N: Autodesk Max 9.0
Design	Valid Passwo.	7 Stepname	Stepname: Enter Priority: Medium Design: S.N: Autodesk Max 9.0
GMAIL	Valid Passwo.	8 Expected	Expected result: Login page should accept valid Username and Password
Communication	Subject / PAYTM/ May release	9 Actual	Actual result: Step 3. Valid Pass. Accepted.
into	Step 3	10 Click on user	Click on user
Gmail	into	11 Step 4	Step 4. Click on user
with V.P.	with V.P.	12 Click on user	Click on user
V.P.o.s.	V.P.o.s.	13 Click on user	Click on user

Test	Application	Test	Design
14 ✓	Stadio Application	15 DESC.	1 TC0001 - Test case ID: N/A
Domain	Path of	2 Subject	Stepname: Launch the login page Expected result: Browser should display URL: Gmail.com successfully
HPALT	HPALT	3 Stepname	Stepname: Enter Username Priority: High Design: S.N: Autodesk Max 9.0
Tool	Valid Username	4 Stepname	Stepname: Enter Priority: Medium Design: S.N: Autodesk Max 9.0
Design	Valid Passwo.	5 Expected	Expected result: Login page should accept valid Username and Password
GMAIL	Valid Passwo.	6 Actual	Actual result: Step 3. Valid Pass. Accepted.
Communication	Subject / PAYTM/ May release	7 Click on user	Click on user
into	into	8 Click on user	Click on user
Gmail	with V.P.	9 Click on user	Click on user
V.P.o.s.	V.P.o.s.	10 Click on user	Click on user

* Test Case design Techniques :

- 1) Boundary Value Analysis 2) Equivalence Partitioning
- 3) State Transition diagram 4) Use case diagram
- 5) Decision Table.

* Test Case Review

- After Test Case Design phase, there will be Review of test cases & then we will prepare Traceability matrix.

Different types of Test Case Review :

- 1) Self Review 2) Peer Review
- 3) Internal Review 4) External Review

Reviewer will Review for :

- 1) Whether all the requirements are covered or not.
- 2) Whatever scenarios we have written, Have we covered Test cases for it?
- 3) Spelling mistakes ...
- 4) If there are any steps missing in Test Case.
- 5) Priority of the Test case.

* Which kind of Review happens / followed in your organization?

→ Internal type of Review is followed in my organization.

1) Self Review

Test engineer Himself/Herself is going to review his/her own Test cases.

2) Peer Review [Testing Team]

Testing Team members will review Test cases.

3) Internal Review [Test team + Dev Team + BA]

- Done by Testing Team + Dev. Team + BA of same organization or it happens within organization.

4) External Review/Walkthrough [Client]

- Client will Review the Test case of Test team + Dev Team + BA will also be present.
- T.E is going to Present/give demo of Test cases whatever he have written to the client.
- Client may pass Review comments.
- This is also called as "Walkthrough"

MOM : Minutes of Meeting

- Prepared by Test head.
- All discussed points, Agenda
- Attendees, Time, Duration
- Summary of Walkthrough

* Traceability Matrix

- Requirement traceability Matrix or Requirement Mapping Sheet.
- Prepared by Test Engineer.

Definition : It's a mapping between Business Requirement & prepared Test cases.

Purpose : After Test Case Review, if any Test cases are missed then we are going to find out by preparing Traceability matrix.

Two ways to prepare : 1) Excel 2) HPALM TOOL

Two Types : 1) Forward RTM 2) Backward RTM.

1) Forward Traceability Matrix

- Prepared Before Test case Execution .

Sr.No.	Requirement ID	Requirement Name	Prepared Test cases T.C. Name / ID
1.	HPALM-ID	login with valid UserName & Valid Password	TC0001-HDFCNet-ValidUsername-ValidPassword
2.		Fund transfer NEFT missing Requirement	No Test case.

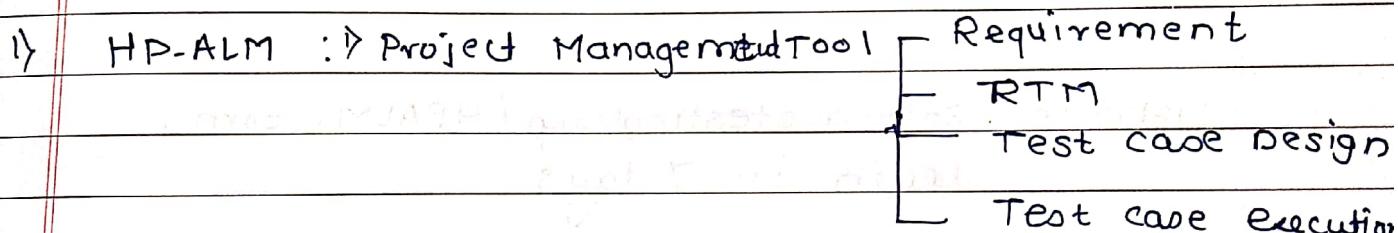
2. Backward Traceability Matrix

- Prepared after test case execution.
- It is a mapping between Business Requirements and Defects.

Sr.No.	Req_ID	Requirement Name	Defects
1	HPALM Generated ID	Login with Valid User Name & Valid Password	Defect ID 0001

21/6/19 Project Management Tool / Defect Management tool

- HPALM -> JIRA



2) Defect Management Tool - Defect log f tracking.

History : Initially Tool name "Mqc" developed by Mercury.

- qc - Quality center.

- HP company takeover the Mqc & renamed HPqc 9.2 (2009-10)

- HP-ALM - HP Application Lifecycle Management.
versions - 11, 12

latest version - 12.53

- * Which version of HPALM you've worked?

- HPALM 12 & HPALM 12.53

- * Why QC/ALM is used?

Many things we can do in single tool:

- We can generate progress report of any Release
- Can create/generate requirements.
- We can prepare Traceability Matrix.
- We can create release & cycle
- We can design/write Test cases
- We can execute test cases.
- We can log the defect.
- We can link the defect with test failed test case.

Website: softwaretestinghelp / HPALM.com .

Learn in 7 days.

Enter Username & password.

Project: Specific project assigned to T.E

Domain: Specific Domain of Project.

- * Components of HP-ALM tool:

- Dashboard
- Requirements
- Defects.
- Management
- Testing

HP-ALM 12

- Test Resource

HP-ALM 12.53

- Test Run

1) Dashboard : Manager, Test head.

- T.E is not involved
- Graphical Representation of Progress report like graphs, pie-chart.

2) Management [Releases

Releases [libraries (T.E not involved)

Releases → cycle → Sanity Test

→ Functional Test

→ Regression Test

→ Create Release :- Release Name

- Start Date of release
- End Date of release.

Create cycle - Cycle Name

- Start Date of cycle

- End Date of cycle

3) Requirements

- Create Requirements - Business Models.

* Traceability Matrix (we can create)

4) Testing

1) Test Resources / Test Run

2) Test plan

3) Test lab

Test plan - planning - designing : T.C. Design

Test lab [lab - execute: T.C. execution

[linking of defects with failed T.C.

5) Defects : log the defects. for failed Test cases.

* Release :

Release Name	what is to be tested	Start date	End date	Cycles
1. May Release	Feature 1 Feature 2	1 May 2019	30 May 2019	1. Smoke Test (1-May - 31May) 2. Sanity (4 May - 10 May) 3. Functional (11-May - 30May)
2. June Release	Feature 3 Feature 4	1 June 2019	30 June 2019	4. Smoke test 5. Sanity 6. Regression for May features. 7. Functional Testing.

Management:

↳ Releases. → By Default folder.

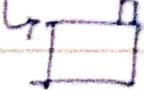
↳ New Release folder.



Name: ALM .



Releases



ALM Training

Releases

↳ PayTM → New Release
Right Click

Name: May Release

Start Date: 5/1/2019 END Date: 5/30/2019

Description:

May Release.

Releases

↳ PayTM

↳ May Release → Right click → New cycle.
New cycle.

Name: Sanity Testing

Release Start: 5/1/19 Release End: 5/30/19

Start Date: 5/15/19 End Date: 5/15/19

Description:

overlapping
of dates
is allowed.

eg. 16/5 - 17/5
sanity.
14/5 - 18/5
functional.

Date change :- Right click on cycle → Reschedule.

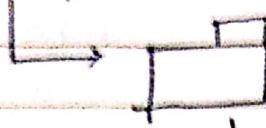
* Requirements

Requirements

→ Requirements.

→ Business Models

* Requirement



Requirements (By Default)

↳ Right click

↳ New folder Pay TM.

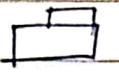


Pay TM

↳ Right click.

↳ New folder

May Release



May Release. (10 Requirements)



↳ Right click

↳ New Requirement

* Name : login Credentials. Requirement Type : Testing

Details :
Author :

Creation Date :

Creation Time :

Direct cover :

Modified :

Priority :

product :

Reviewed :

Target cycle :

Target Release :

Description :

Detailed description of requirement.

→ After clicking on Submit button
the requirement will be
automatically created. (Auto increas

* Traceability Matrix.



↳ May Release

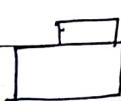
↳ Requirement author IP.

↳ Double click on Requirement

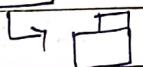
Name of Requirement	select	Test plan Tree
click on → Test coverage click	pull ↗	<input type="checkbox"/> Subject <input type="checkbox"/> Pay TM <input type="checkbox"/> May Rel <input checked="" type="checkbox"/> TC001 <input checked="" type="checkbox"/> TC002 <input type="checkbox"/> TC003 <input type="checkbox"/> TC004
	OK cancel Help	

* Testing → Test plan.

- To design the Test cases. in HP-ALM.



Subject (By default).



↳ Pay TM



↳ May Release

→ Right click "New Test". [Test case]

* Test Name: [Login with valid data.] * Type: [Manual]

Creation Date: []

Designer: [S.N.Autod]

Status: [Design]

Description: Verify that user should be able to login with valid data

[OK] Click on / Test ID autogenerated

Select the test case.

↳ Design steps.

↳ New step [Alt + N]

Step Name : Step 1			
Description : launch the URL : gmail.com			
Expected Results : gmail.com page Should open			
File	Edit	OK	Cancel
Help			

* Test plan lab: Execution of Test Cases

Root (Default Package)

↳ Paytm

↳ May Release

↳ Right click

↳ New test set [Test Suite]

Name : _____ Type : _____

Details

Baseline :

Modified :

Status :

↳ [Sanity Test · [sanity suite]]

↳ click of set click on 'select tests'

Select Tests .			Test plan Tree.
Details	Execution Grid	...	
Name	Test Name	Status	
login	login with valid data.	No Run	<input checked="" type="checkbox"/> Subject <input checked="" type="checkbox"/> Priority <input checked="" type="checkbox"/> May Release <input checked="" type="checkbox"/> login Valid data

- By default status of all test cases will be "No run".

- Passed - All steps in the test cases will be passed then complete test case will be passed. If any step gets failed then complete test case status will be "Failed".

Not completed - suspend / paused.

⌘ Defects : For Failed Test cases .

Test lab :

linking of test case to defect : Failed Test case .

- double click on ~~TS~~ Failed Test case .

	Defect ID : <input type="text"/>	
Linked Defects	<input type="button" value="OK"/> <input type="button" value="Cancel"/> <input type="button" value="Help"/>	

* Defects

→ New Defect [Template of Defect]
[Components of Defect]

New Defect

* Summary :

Details

* Type : Defect

* Severity :

* Process :

* Sub process...

* Detected By :

* Detected on

* Detected In : Release

* RICEFW ID

* Defect Category :

* Status :

Disposition : Pending Priority :

Description : Response to last reported defect

OK

→ Defect ID will be auto generated.

- After clicking on submit or OK button then mail will be generated by tool & will send it to the test engineer who have detected the defect & also to the developer to whom it is assigned.

► Summary : One lines Description.

SIT-Gmail Login - After click Not working properly

<u>Env. Name</u>	<u>App Name</u>	<u>Module Name</u>	<u>issue</u>
------------------	-----------------	--------------------	--------------

2) Type : Defect.

3) Detected By : Test Engineer.

4) Detected on : Date on which defect found & logged.

5) Detected in Release : May Release (Release Name)

6) Detected in Cycle : Functional (cycle Name)

7) Priority : What is priority of defect. (Importance)

8) Severity : Severity of defect. (seriousness)

9) Status : By default "New" for new defect.

10) Assigned To : Developer whom the defect can be assigned.

Q How to find your defect in the 1000 of entry in defect?

- Apply filter on "detected on" & then "detected by".



Defect.



Priority

Severity

Def^P: - Importance of the defect w.r.t. customer requirements. - seriousness of defect w.r.t. functionality.

Eg. 1. Requirement: logo should → All the functionality be in Blue color. working properly.

But developed in Red color.

Severity: Low

Priority: High

2. Bottom line: spelling → severity: low mistake.

Priority: low

High priority ←→ High Severity

High priority ←→ low Severity

low priority ←→ High Severity

low priority ←→ low Severity

>To change the status of defect:

Click on Defects → Double click on defect → change the status:

Status: New

open

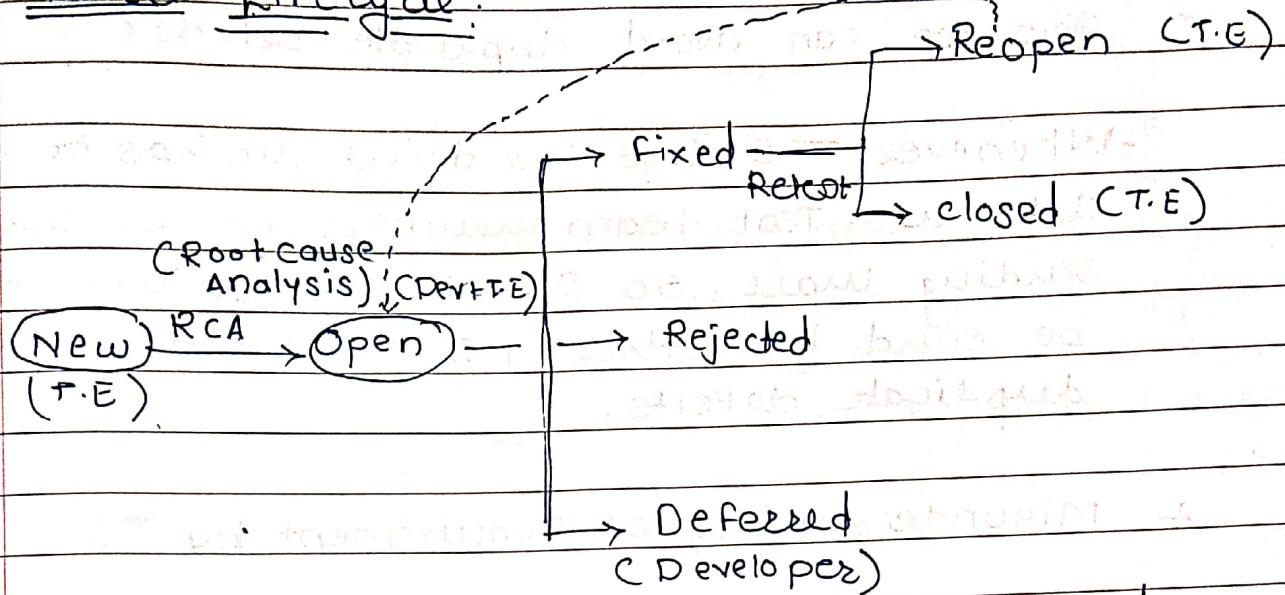
Fixed

Close

Rejected

Deferred
Reopen

* * Defect Life Cycle



New: Defect is found by Test engineer and T.E is now logging this defect for the first time. By default all defects have status "New".

Fixed: Defect is accepted by developer and he'll fix it and will provide modified build to T.E then T.E has to check whether defect is fixed or occurring again & choose status "Reopen" or "Closed" accordingly.

Rejected:

* What are the reasons developer rejects your defect? Why developer rejects your defect?

1 Environment issue : Runtime error, Build issue.

2 Duplicate Defect : No same (100%) x Similar (99%)

L → Defect is similar or likewise of other already raised defect & assigned.

e.g. Pop-up window on every screen of the application.

3 Hardware & Networking issue : LAN speed slow,

* How we can avoid duplicate defects?

- Whenever T.E logs the defect, he has to keep all the Test team members in CC while sending mail. So similar defects will not be send by other T.E if we can avoid duplicate defects.

* Misunderstanding of Requirement by T.E.

* If the defect is valid & still developer is rejecting your defect, so as a testee what you are going to do?

- As a testee, we have to convince the developer with strong reasons and proofs.

Proofs :> 1) Screenshots of execution of test case
> 2) Test cases reviewed by same developer
> 3) Requirements.

- After this even developer is still rejecting your defect then we will approach to test lead. (If also we can discuss this in scrum meeting).

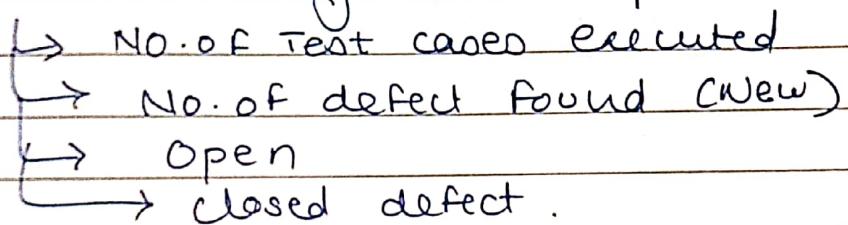
Deferred :- Developer is accepting defect, But due to lack of time developer is saying a defect will be fixed in next sprint.

- This can happen when defect is logged on last day of sprint & we don't have time for regression testing.

- This kind of defects will be informed to client also.
- In case of critical defected defect, client will pull out Jira, that means ~~the~~ new change will not be deployed in the sprint.
- This decision must be taken by client only.

* Open

DSR → Daily Status Report .



Open :- When new defects are raised if developer is not taking action for long time then status will be changed to open .

e.g. Defect found - Tuesday : Status : New.

No action taken by developer .

Wednesday : T.E can make status "Open".

- Its not good to have more defects open bcz there may be chances that some other test cases are dependant on same defect if testing might get stopped .