TESTING TOOLS

MANUAL

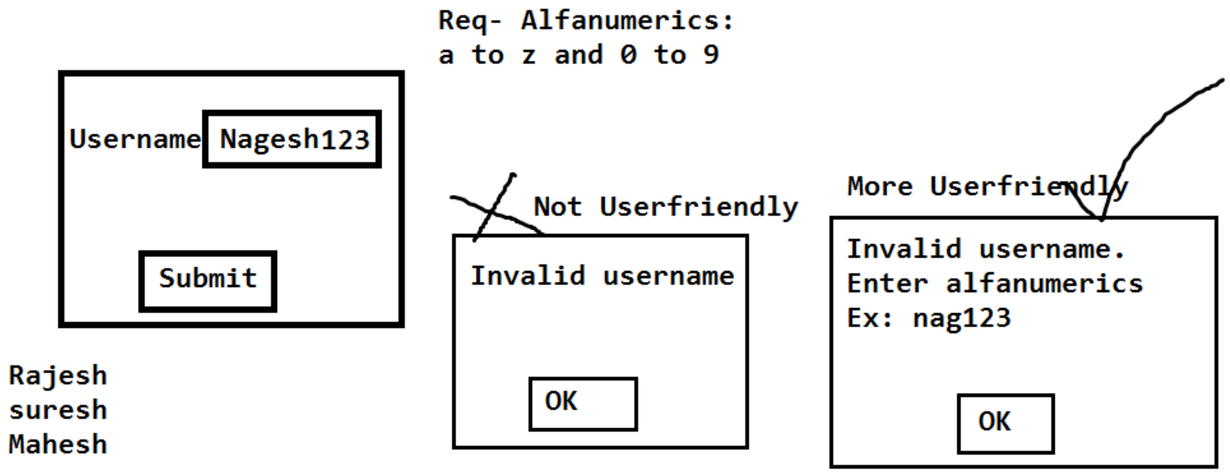
Testing:

- The process of identifying the bugs in a software (project/product) is known as *"Testing".*
- The Test Engineer has to cross check (validate) whether the developed software is as per the clients requirements or not. He is responsible to deliver quality software to the client.

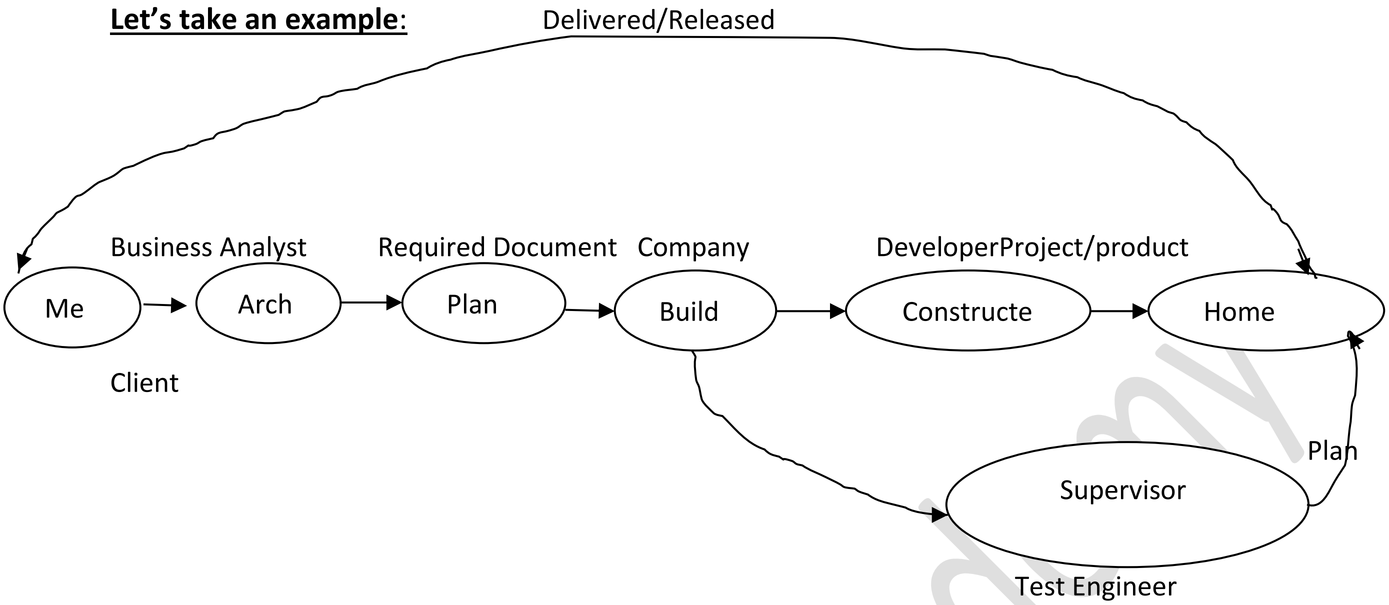
**Note: The main responsibility of the test engineer is, he has to check(validate) whether the developed application (software) is useful for end user or not.

Bug: The deviation from the requirement is known as the bug or defect.

Quality: Justification of all the client requirements with user friendliness and security is known as *quality*.



Let's take an example:



Project:

It will be developed based on the client's requirements; once it is developed it will be delivered to the client. Based on the clients need the team members (End users) will use it.

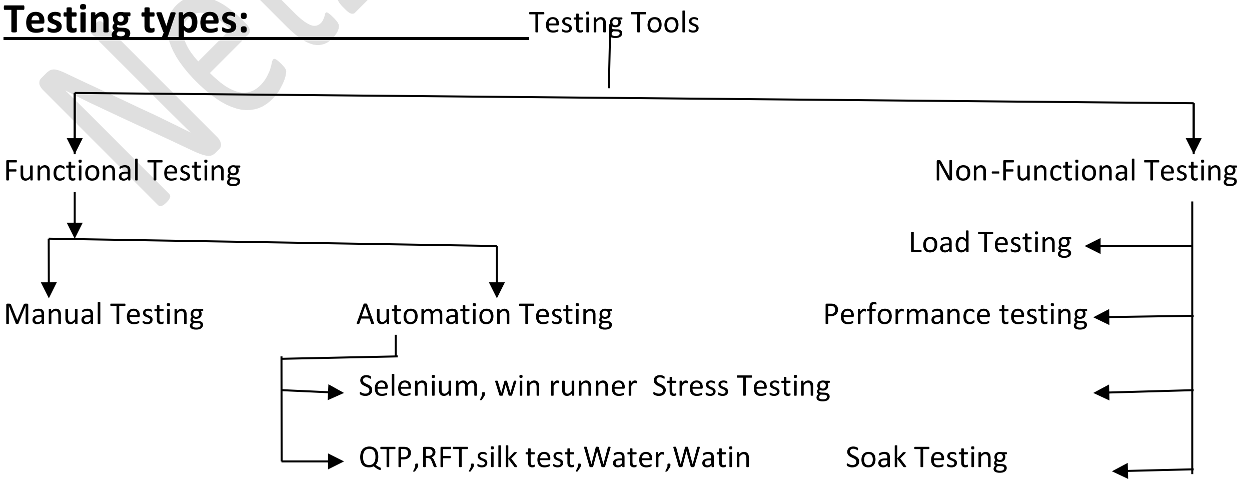
Ex: Spicejet.com, selenium4testing.com, Constructing own house with our requirements.

Product:

It will be developed based on the company's requirements. Once it is developed, it will be released in the market based on the customer needs they will choose the product.

Ex:Mobile App, Calculator, Facebook, yahoo, MS Office, Mac operating system, windows operating system, Gmail etc…

Testing types:



Internet based Applications:

These Applications can be accessed worldwide with unlimited number of users.

Ex: Gmail, Facebook, selenium 4testing

Intranet based Applications:

These applications also can be accessed worldwide, But with limited number of users.

Ex: Applications which are limited to a specific organization.

Bidding of the project:

Roles Involved:

1. Marketing Business Analyst(BA)
2. Engagement Manager(EM)

- Marketing Business Analyst(BA):

Marketing BA will meet the client and convince the client with the proposal. Once the client is satisfied with the proposal then the client and company will sign off the project.

Sign Off:

The official agreement between the client and company about the project delivery is known as '*sign off*'.

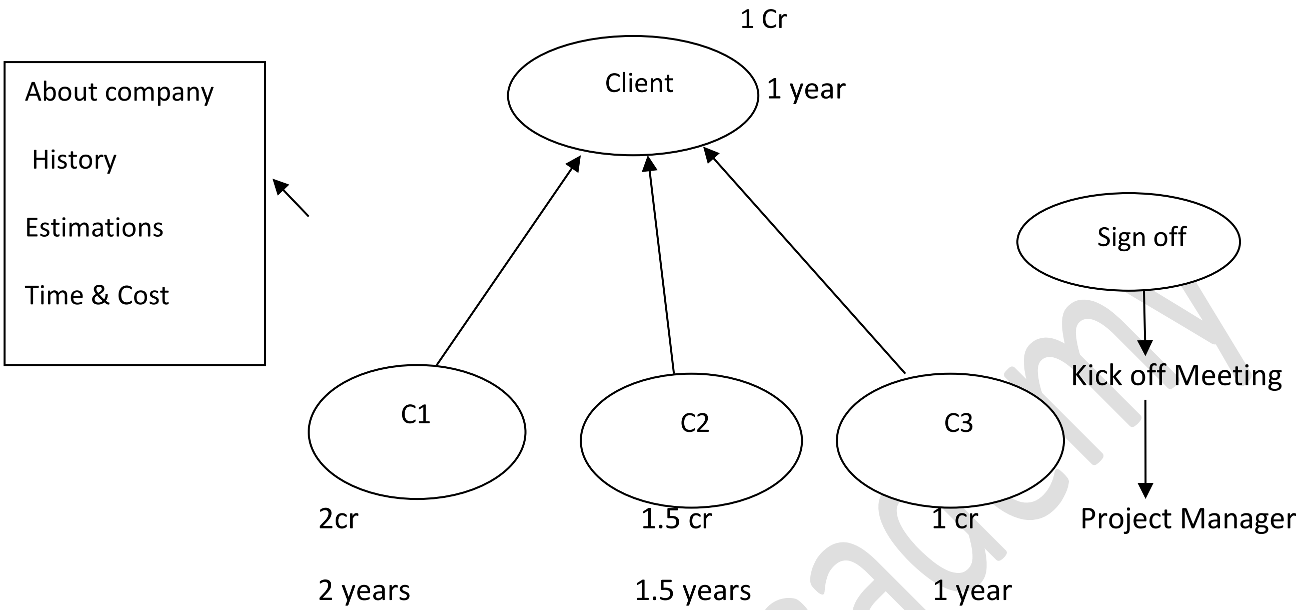
KickOff Meeting:

Once the project is signed off company will go for a '*kickoff'* meeting. It's a quick meeting where all the high level management will participate and they will announce the project, client and they will choose the project manager for the project. The project manager is responsible for SDLC.

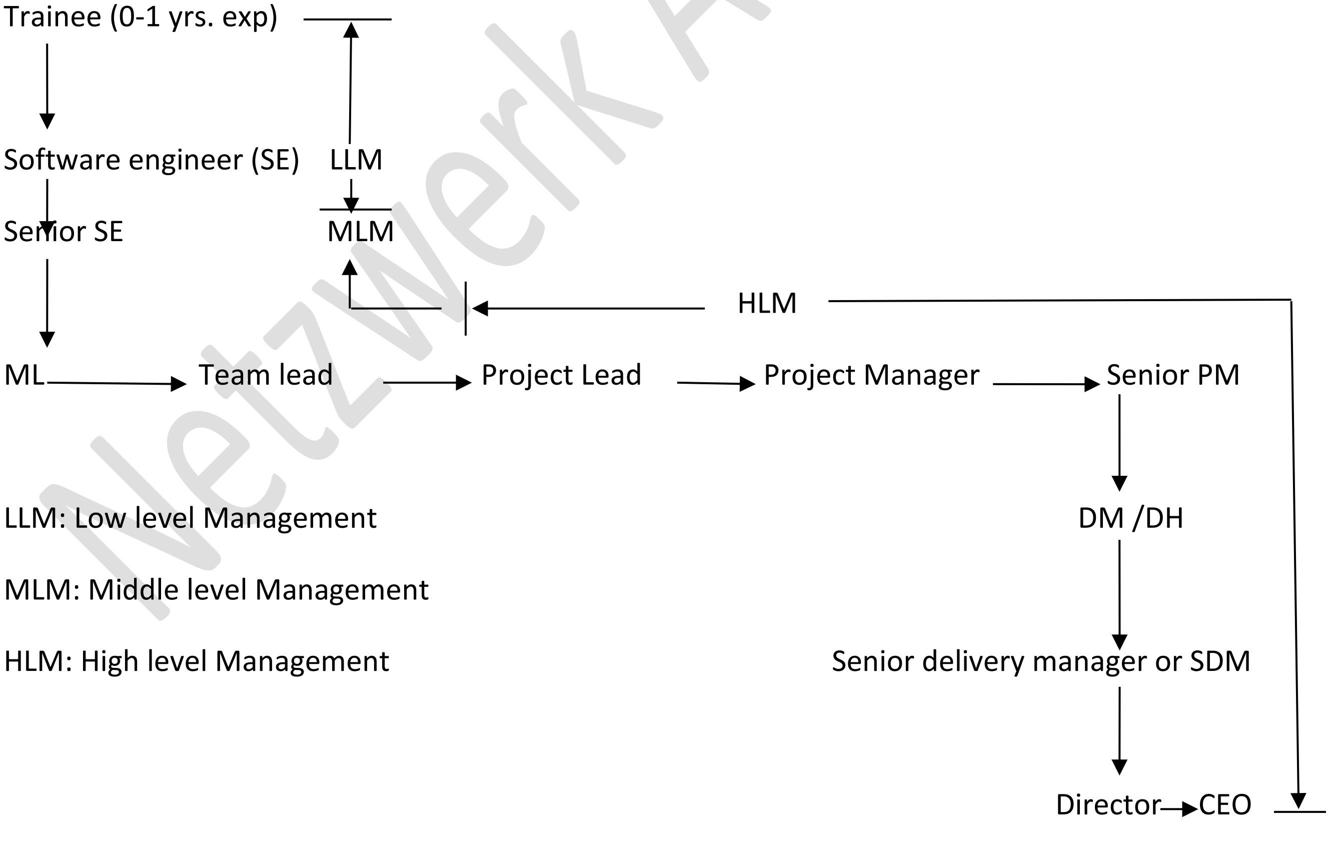
- Engagement Manager(EM):

Engagement Manager is responsible to maintain the relationship between the client and company. He acts as a bridge between the Client and the Company.

Bidding of the project



Project hierarchy or Designation hierarchy or Role hierarchy



LLM: Low level Management

MLM: Middle level Management

HLM: High level Management

SOFTWARE DEVELOPMENT LIFE CYCLE(SDLC):

It consists of the below phases,

1. Requirement Phase
2. Analysis Phase
3. Design phase
4. Coding Phase
5. Testing phase
6. Delivery and maintenance Phase.

PIN: Project initiation/intimation note:

It's an email, will be prepared by the project manager where it contains the start date and end date of the project. The mail will be sent to the client and the high level management.
PIN indicates the start of the project.

1) Requirement phase:

Roles: Business Analyst
Engagement Manager
Project Manager

- BA is responsible to collect all the requirements in a Requirement template document (RTD).
- Once all the requirements are collected in the requirement template then they will sign off the requirements,
- The signed off document is known as *SRS (Software/system requirement specification)* or *BRS (Business requirement specification)* or *FRS (Functional requirement specification)* or *BDD (Business design document)* or *BD (Business document).*
- Once the SRS document is base lined, the BA is responsible for POC (Proof of concept).
- During POC the BA is responsible to develop the prototype and it will be presented to the client.

  Prototype: It's rough and rapidly developed sample Application; it doesn't contain any of the actual functionalities. It simply describes how the project is going to be and how it is going to display. The main purpose of the prototype is to collect all the requirements properly and also to understand all the requirements properly.

- Engagement Manager is responsible to maintain Rapport or Relation between the client and company. He will also concentrate on the extra requirements, extra cost and extra time of the project.

- Project Manager is responsible to monitor the phases and he will help both BA and EM to complete their activities properly.
  Note: The outcome of requirement phase is SRS and Prototype

2) **Analysis Phase:** Analyzing the requirements.

Roles: High level Management
Middle level Management
Project Manager
BA

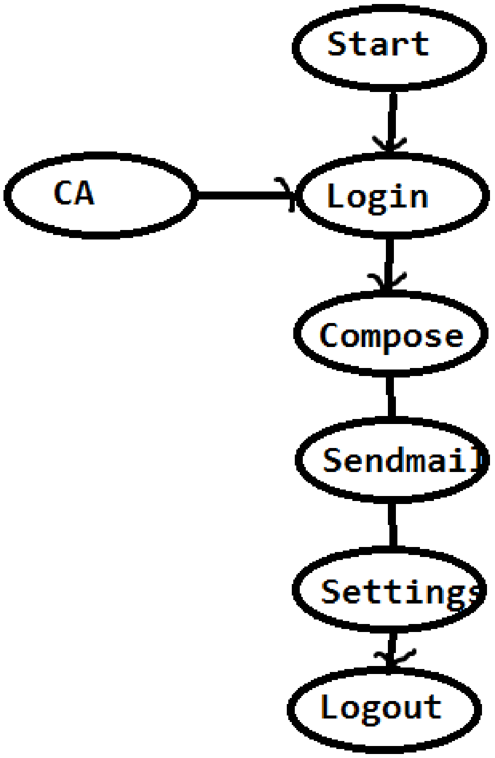
- All the above roles will assemble for a meeting and they will perform the below activities
  a. Feasibility Study
  b. Technology selection
  c. Resource plan
  d. H & S plan
  a. **Feasibility study**: Feasible means possible or not. The above roles will take each and every requirement in SRS document, Requirements will be analyzed and they will identify whether it is possible to develop or not, if it is possible to develop then they will identify how much time is needed for development, testing and delivering it to the client. If any requirement is not feasible to develop then they will inform it to the client.
  b. **Technology selection:** The list of technologies like Java, .net, MSSQL,Oracle, selenium etc. are required for the project development, testing and delivery to the client will be described here. Based on the technologies they will hire the resources.
  c. **Resource Plan:** The number of resources like developers, test engineers, database engineers are required for the project development and testing will be described here.
  d. **Hardware and software plan:**The number of hardware’s like desktops, laptops, mobiles, printers etc...With related softwares are required for the project will be described here.

- All the above will be documented in document called project plan. It will be sent to the client, for review.

3) Design Phase:

Roles: Architect/chief architect
Business Analyst (BA)
Project Manager

- Architect will review all the requirements of the SRS Document, while reviewing if any clarifications are needed on the requirements then BA is responsible to clear all the uncleared requirements.
- Once the Architect is clear on all the requirements then he will divide the requirements into multiple modules and sub modules. Group of related requirements is known as '*Module*'.
- Once all the modules are divided then he will provide the architectural diagram (flow diagram) of the entire project with the help of UML (unified modeling language)
- All the above will be documented in a document called Design document or SRS.



4) Coding phase:

Roles: Developing team
Testing team
BA
Project Manager

- Once the modules are divided by architect they will be assigned to the developers as well as testing team.
- The developers are responsible to develop the source code for the modules. Once the source code is stable then they will *check in* the source code into the *central repository*.
- The development lead will *checkout* the source code to his local system, then the development lead will build the source code and the build will be released to the testing team for testing.

Central Repository:

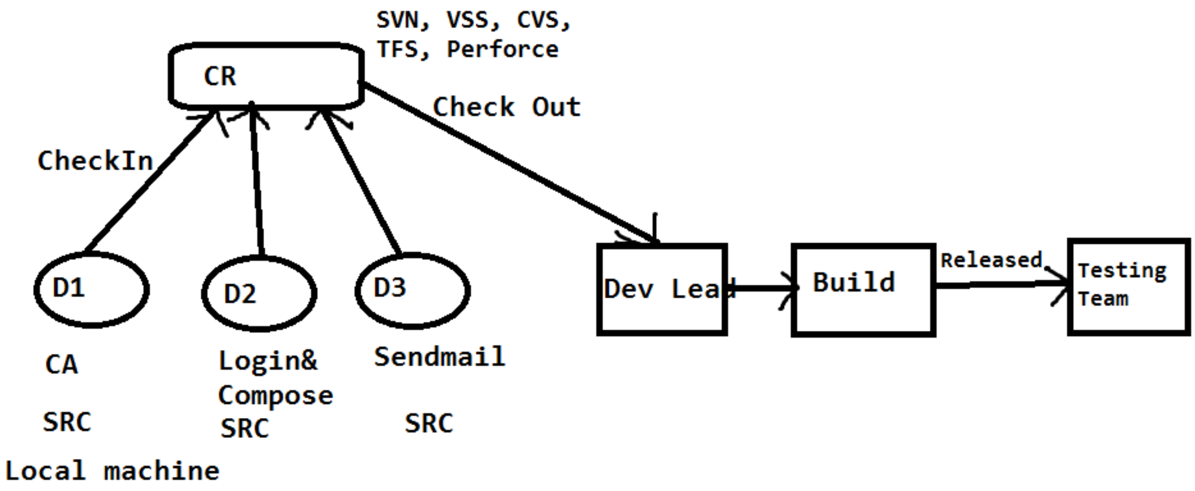
Repository means a storage folder. Central Repository means the folder which is commonly accessible to all the resources in the organization. It contains all the secure files.

Ex: SVN- Sub Version
VSS- Visual source safe
TFS- Team Foundation Server
CVS- Concurrent Version system
Perforce

Check in: The process of uploading the files from the local system to central repository is known as *Check in* or *Commit*.

Check out: The process of downloading the files from the central repository to the local system is known as *Check out*.

Build: The process of converting the source code to executable code is known as *Build*.

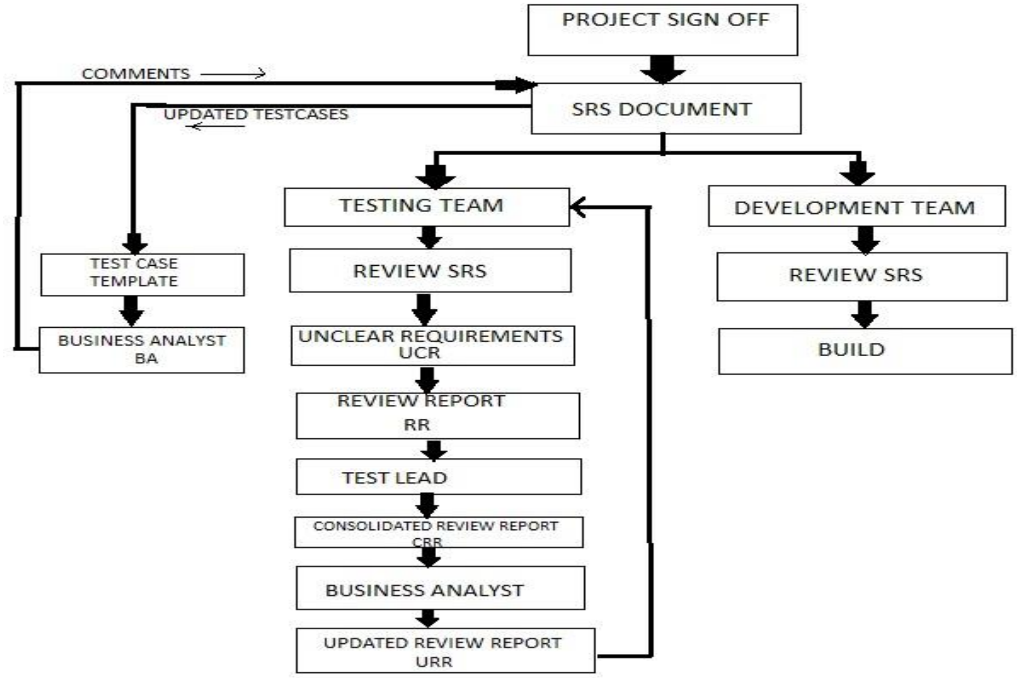


5) Testing Phase:

Roles:Test Engineers
Dev team
Business Analyst (BA)
Project Manager

- Once the SRS Document is base lined (Completed), it will be sent to both development team as well as testing team.
- Development Team is responsible to review the SRS document, understand it and develop the build.
- Testing team is also responsible to review the SRS Document. While reviewing, if any *unclear requirements* (Doubts) are identified those will be updated into the document called “*Review Report(RR)*”.
- The review report will be sent to the team lead where he will consolidate (Make one document) all the review reports to a single document called “*Consolidated review report*(CRR)”and it will be sent to the BA.
- BA is responsible to Review all the unclear requirements and he will provide the clarifications, then the “*Updated Review report*(URR)” will be sent to the testing team
- The testing team will again Review the SRS document with clarifications.
- Once the testing team is very clear on all the requirements then they will take the test case template and write the test cases for all the requirements.
- The test cases documents will be sent to the BA. Where he will review it and he will provide the comments whether any new test cases are needed to be added.
- Based on BA comments the testing team will update the test cases.
- Once test cases are base lined (completed) it will be sent to the client for final review.

- Once the build is released to the testing team, they will execute all the test cases on the Build.
- While testing the build, if any Bugs are identified it will be reported (send) to the development team. Developer will fix it and send it back to the testing team for testing.
- Test Engineer will test that whether the bug is really fixed or not and at the same time he will check for other bugs.
- If identified it will be reported to the developer.
- The same process will be continued until the build is stable (Bug free or No Bugs).
- The stable build will be delivered to the client.



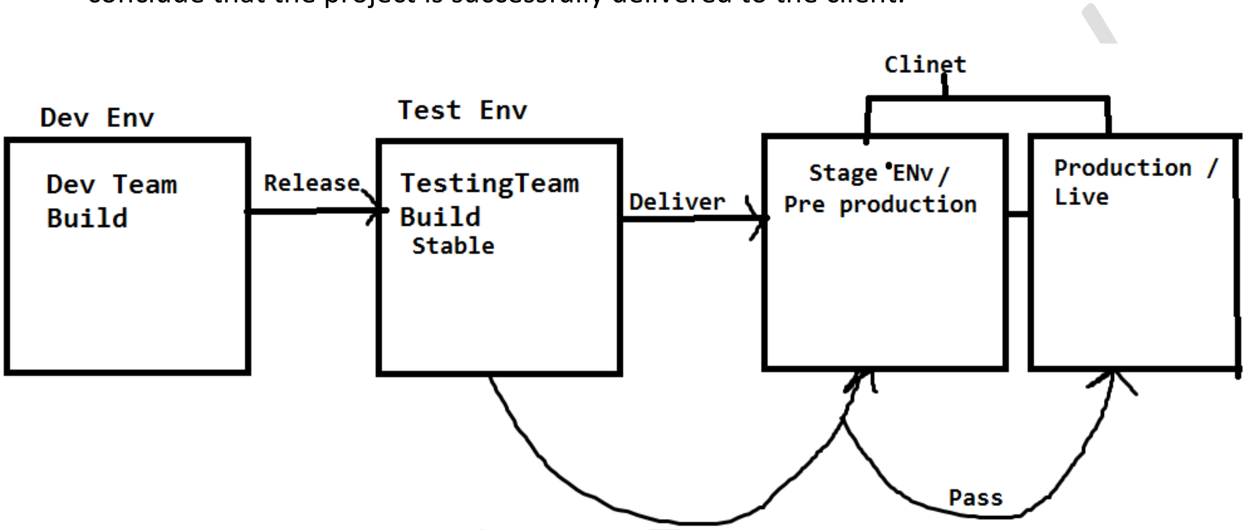
6) Delivery and Maintenance:

Roles:Test Engineers
Developer team
Business Analyst (BA)
Project Manager
Client

Delivery: Once the build is stable in test environment the testing team (TL) will send email to the project manager saying that the build is stable then the project manager will deliver the build to the client.

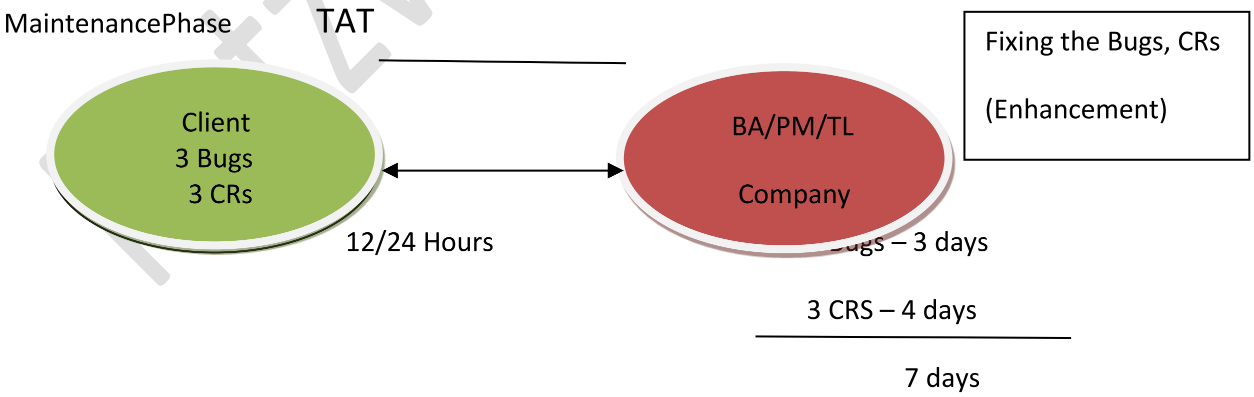
- The client will deploy the build in stage environment and perform testing.

- *Stage environmen*t is the common environment where both testing team and client's team will test the application before it go live. It is also known as *pre-production environment.*
- Once the Build is stable in stage environment then the client will deploy the build in *live or production environment.*
- Once the build is successfully deployed in production/live environment then we can conclude that the project is successfully delivered to the client.



Maintenance:

'Live' means when client or end user are using the application. Maintenance will be continued as long as application is on live.



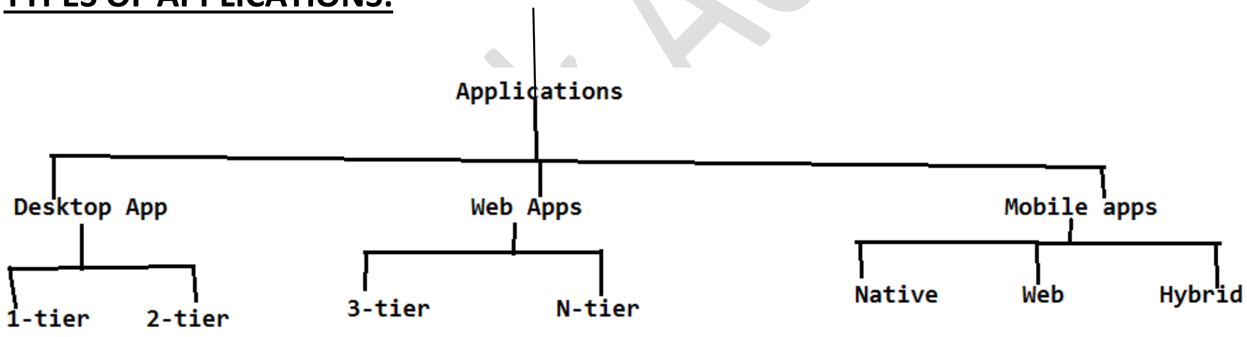
- Once the project is successfully delivered to the client and it is successfully deployed in live/production environment then the maintenance of project will be started.
- During the maintenance the company is responsible for two activities.

a) Fixing the Bugs.
b) Updating the enhancements/Change requests CRs.

- As long as project is on live the maintenance of the project will be continued.
- As per the signoff (Agreement) initially the company will be providing free maintenance up to 3 to 5 years.
- Once the free maintenance period is completed the client will renew the maintenance agreement.
- Whenever the client is sending any bugs and CRS then from the company someone (Developer, BA, Tester) has to send the acknowledgement email to the client within the TAT (Turnaround time) agreed as per the agreement, it could be 12/24 hours.
- The mail contains the number of hours which we are going to take to fix and deliver the new build to the client.
- As long as project is on live the maintenance of the project will be continued

Q. What kind of applications you have tested?

TYPES OF APPLICATIONS:



There are three types of applications which can be tested;

1) Web Applications
2) Desktop Applications
3) Mobile Applications

1) WEB APPLICATIONS:

These applications are accessed with the help of some browser.
It is of two types

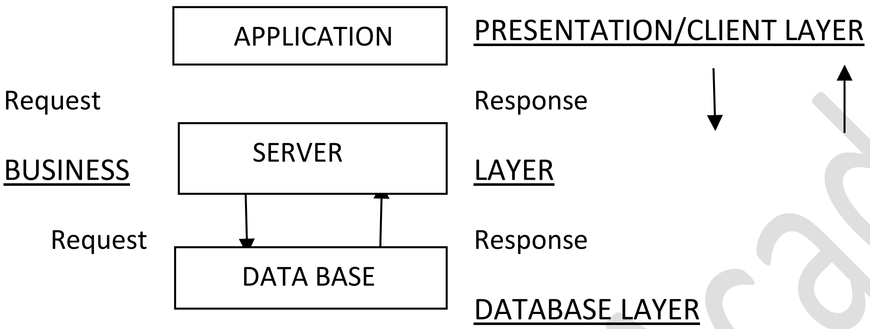
a) 3-Tier architecture applications.
b) N-Tier architecture applications.

Environment/System:

All the applications are combination of the environment.
Environment contains:

1. Presentation layer.
2. Business layer.
3. Database layer.

ENVIRONMENT



1. Presentation layer:

The Front end which the end user is going to access is known as presentation layer/client.

2. Business layer:

It is the server which is responsible to serve the request. It means it will take the request from the application, send it to the database, takes the response from the data base and send it back to the application. The whole process is known as serving the request.

Ex: Tomacat, JBoss, Weblogic, WebSphere, IIS

3. Database layer:

Database layer is responsible to store the data in the form of tables.
Ex: MS SQL, My SQL, ORACLE

a) 3-Tier architecture applications:

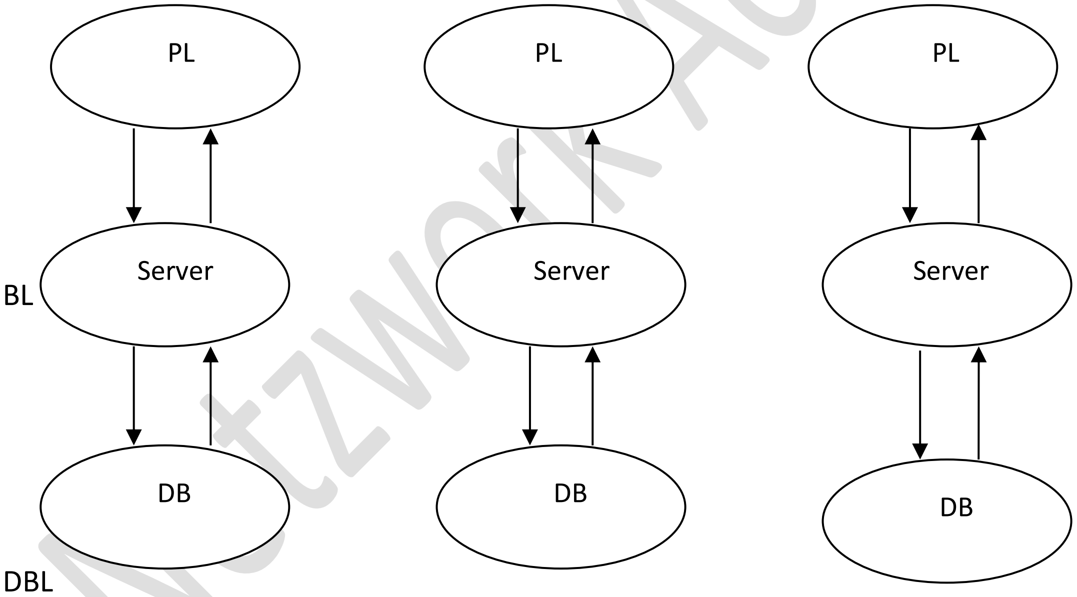
In 3-tier architecture applications, the above 3 Layers are available in three different machines (Layers) which we will call it as tiers. Hence they are called as 3 Tier architecture applications.

Ex: PL - Browser
SErver - Tomacat
DataBase - Oracle

b) N-Tier architecture applications:

It is same as like 3-tier architecture applications based on the number of users (load), we will be having more number of servers and databases.

Based on the load request the business logic will be distributed among the servers and DB's. Hence we will call it as Distributed environment applications.



2) DESKTOP APPLICATIONS:

These applications can be accessed without any browser.

Ex: Skype, calculator, MS Office, vlc player etc.

It is of two types:

- 1-Tier

- 2-Tier

- **1-Tier:**

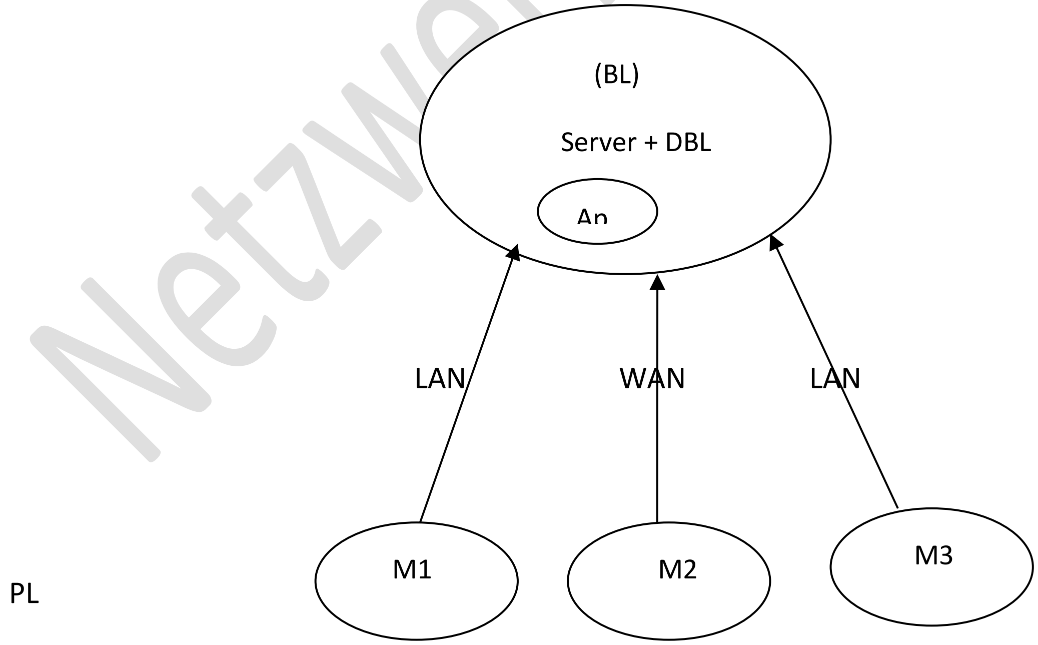
  These applications are limited to a specific system (PC). All the 3 layers PL, BL and DBL will be available in that specific system only.
  Ex: VLC player , Calc

- **2-Tier:**

  The presentation layer will be available in one system the business layer and Data base layer will be available in another system, with the help of LAN/WAN we can able to access these applications from multiple systems. Hence we will call it as 2- Tier applications it is also known as client- server architecture applications.
  Ex: Skype

NOTE: For Desktop Applications we need to install application at user/client side then only we arcane able to access it. To perform testing for desktop applications we need to perform it in both client as well as server.

For Web application, we need to test it in client only.



3) MOBILE APPLICATIONS:

The applications which can be accessed on the mobile platform are known as Mobile Applications.
Ex: Android, IOS, Blackberry, Windows etc.

They are of three types

a. Native Applications
b. Web Applications
c. Hybrid Applications

a. Native Applications:

These applications can be accessed without any browser.
Ex: Viber, call functionality, msg, clock etc.

b. Web Applications:

These applications can be accessed with the help of the browser in the mobile.
Ex: selenium4testing.com

c. Hybrid Applications:

These Applications can be accessed both without browser and with browsers.
Ex: Gmail/Gmail app, Facebook/Facebook app, Banking Websites/App etc.

NOTE: For Web Applications not necessary to install application at user/client side as we are able to access with the help of browser. To perform testing for web applications we need to perform it only in client.

TESTING METHODOLOGIES:

Q: Who is responsible for testing. At what level the Test Engg.. will involve in testing

They are of three types

a. Black box testing
b. White box testing
c. Grey box testing

a. Black box testing:

If the resource is performing testing on the functional part of the application then he will be treated as "*Black box tester*".

Functional part means whether the developed application is as per the client's requirements or not. Testers will perform black box testing in test environment and stage env (Pre production env)

b. White box testing:

If the resource is testing the structural part (programming) of the application, then he will be treated as *"white box tester*". Developers are responsible for white box testing in development environment.

c. Grey box testing:

If the resource is having the experience on both testings (white box testing and black box testing). Then he will be treated as "*Grey box tester*".

LEVELS OF TESTING:

If one project has to go from the signoff stage to live(production), it has to undergo the below levels of testing.

1) **Unit level of testing**
2) **Module level testing**
3) **Integration level of testing**
4) **UAT(User acceptance testing)**
5) **System testing**

1) **Unit level of testing:** *Unit* means the smallest flow or scenario in the application.
   - Developer is responsible for Unit level testing.
   - He will divide the assigned module to multiple units and develops the code for all the units.
   - He is responsible to check whether each and every unit is working as expected or not.

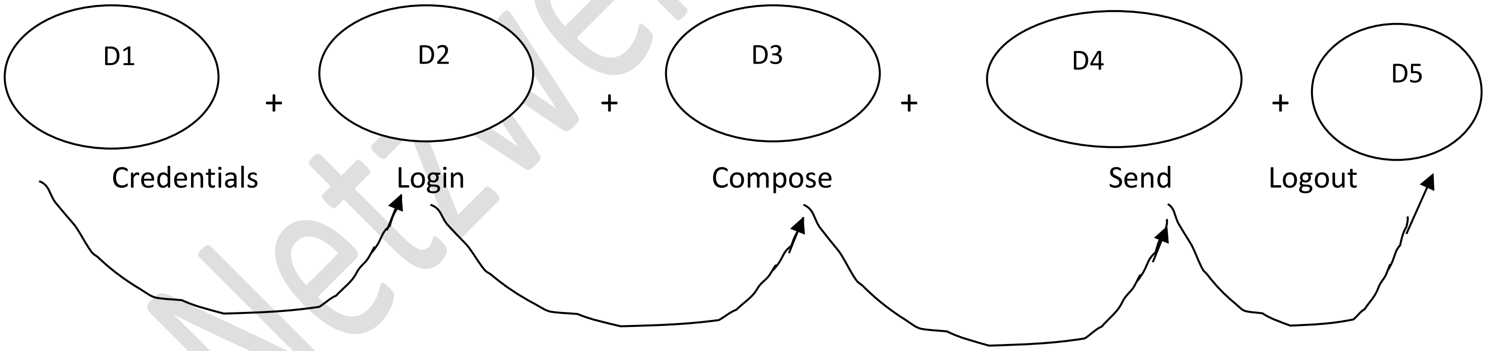
2) **Module level testing:**
   - From Module level testing, both testing team and development team is responsible.

- The developer will combine all the related units to form a module.
- Once the module is developed, the developer is responsible for white box testing in development environment.
- Once the module is released to the testing team, they are responsible for Black box testing in testing environment.

3) **Integration level testing**:

- The process of combining all the developed modules is known as *integration*.
- Check whether the data flow is navigating from one module to other is known as *integration level testing*.
- Both development team and testing team is responsible for integration level testing.

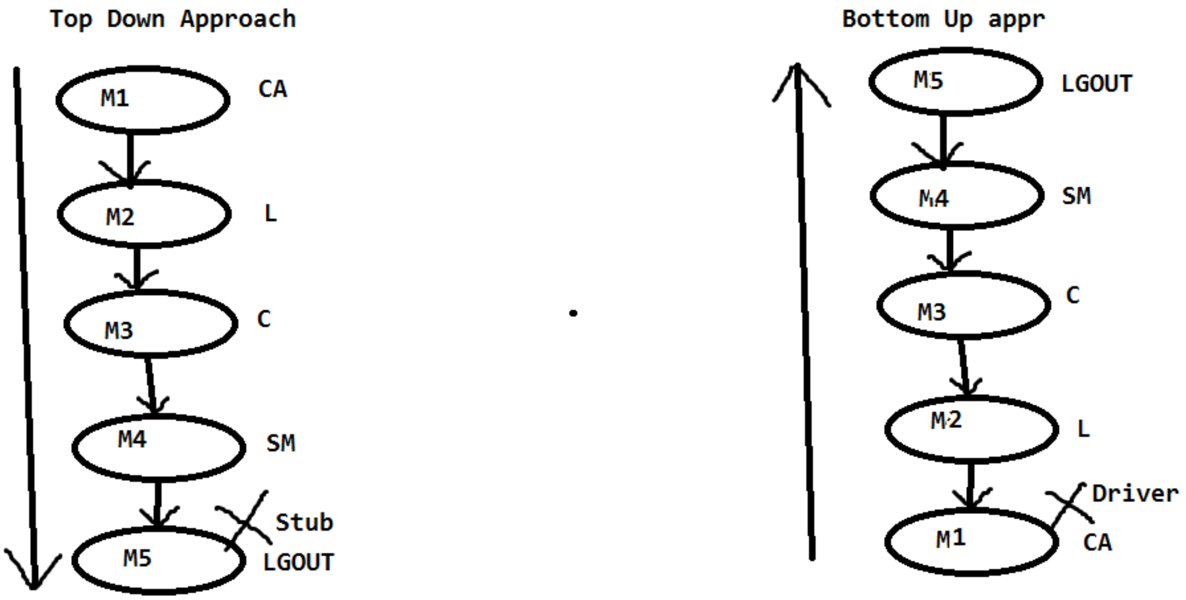
  Ex: Create one account in Gmail, check that whether you are able to login into the application with created account. Then compose mail and send it, check that whether it is properly delivered or not.
- While integration if any mandatory module is missing then the development Lead will replace the mandatory module with some dummy code is known as stub or driver.



Stub/Driver: Both are nothing but a dummy code, it does not contain any functionalities.

- If the development lead is using top down approach to integrate the modules, while integration if any mandatory module is missing then it will be replaced by *Stub*.

- If the development lead is using bottom up approach to integrate the modules, while integration if any mandatory module is missing then he will replace with *Driver.*

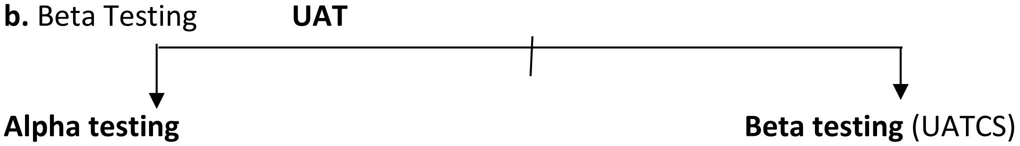


4) <u>User Acceptance Testing</u>:

- It is known as *user/client acceptance testing*. Once the build is stable in test environment then we will planto deliver the build to the client. Before delivering the build to the client, the client will send User acceptance test cases to the testing team for execution.
- The testing team will execute all the UA test cases in test environment; if all are passed then the project manager will deliver the build to the client.
- Client again will execute all the UA test cases in the client's environment (stage environment). If all are passed then the client will deploy the build in the Live or Production environment.
- UAT is of two types:

a.Alpha Testing

b. Beta Testing



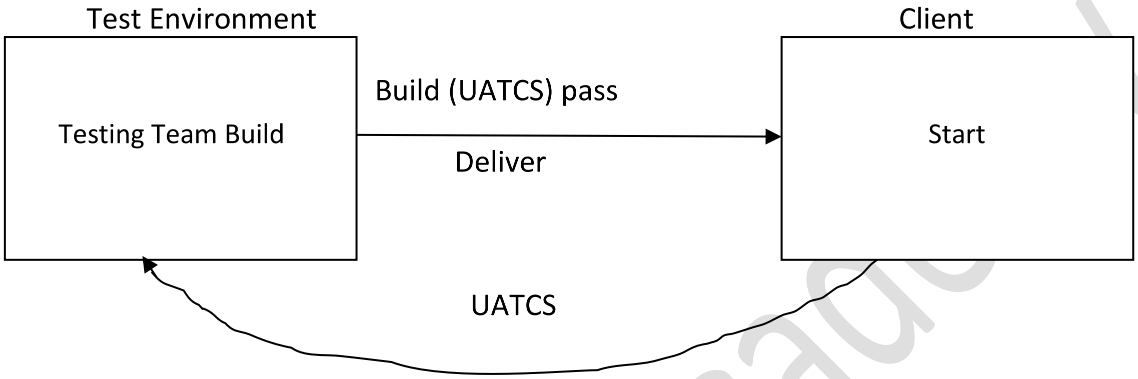
Alpha testing **Beta testing** (UATCS)

(UATCs) Test environmentStage Environment

a. Alpha Testing: Executing all the UA test cases in a test environment by the testing team is known as '*Alpha testing'*.

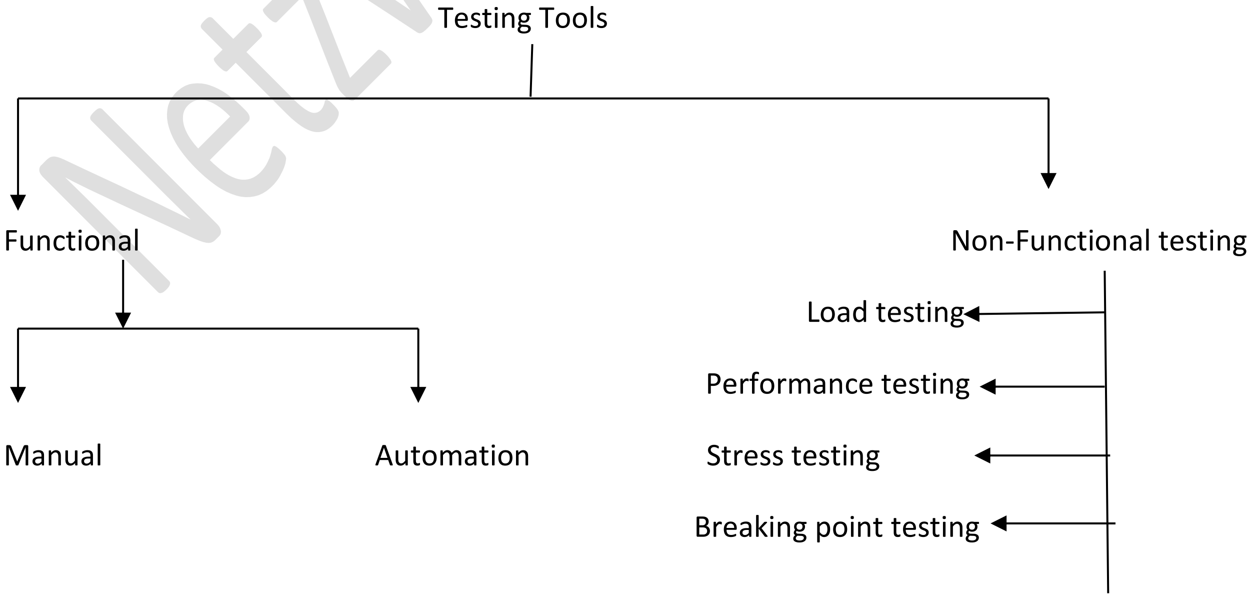
b. Beta Testing: Executing all the UA test cases in the clients Stage environment by the client's team or the testing team is known as '*Beta testing'*.

Once the Beta testing is passed then the client will go to the live environment.



5) System Testing:

- It is also known as *non-functional testing*. Once the application is stable, then we can go for non-functional testing.
- In non-functional testing performance (response time) of the application will be identified.
- The time taken between the request and response is known as *response time*. It will be identified with the help of multiple non-functional testing types like Load testing, Performance testing, Stress testing, Breaking point testing.



SOFTWARE DEVELOPMENT MODELS:

Q. What process you have used to develop your project

The models are as follows.

1) **Waterfall model**
2) **Spiral model**
3) **V-model**
4) **Fish Model**
5) **Agile process**

1) Waterfall Model:

It was the initial process or model introduced for software development (old process). The sequential execution of all the phases in SDLC is known as water fall model. Once the phase is completed, high level management will analyze that phase.

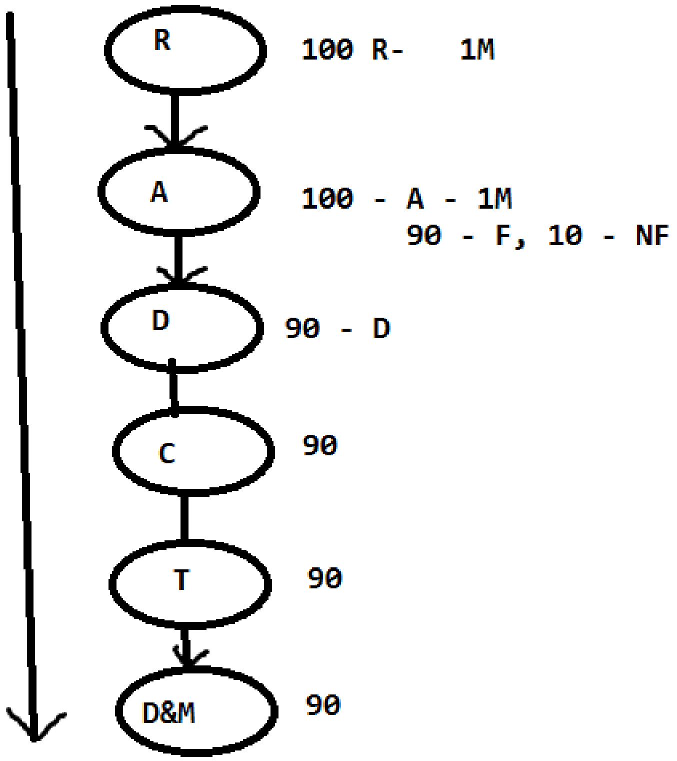
> **NOTE:** How waterfalls from one level to other, in the same way the phases of SDLC will be implemented.

Advantages:

- It is very easy to implement the project because it is Sequential Execution.

Disadvantages:

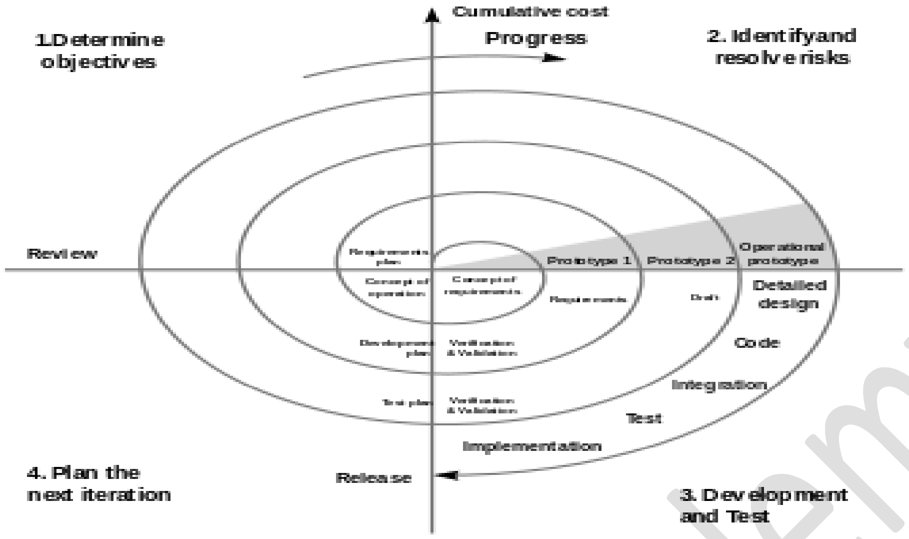
- The risk can't be identified at the early stage of the life cycle so it can't be prevented.
- It is time consuming process as well as costly process.
- We can't accept the requirements change in the middle of the project. If still needs to be accepted then we will accept the requirement change in the form of CRs-change requests. Change requests are done at the end of the project and CRs charged by the company.



month

2) Spiral Model:

- Spiral model is a combination of waterfall model and prototype.



- Instead of collecting all the requirements once, the BA collects few requirements, it will be analyzed and designed with the help of the prototype. Then it will be given to the development.
- Once the developer develops the build then it will be released to the testing team. The same process will be continued for all the requirements.
- Once all the requirements are completed and the build is stable, then the build will be delivered to the client.

Advantages:

- We can save the time and cost, because we are executing all the phases in parallel.
- The risk can be identified at the early stage of the SDLC and it can be prevented at the early stage of the life cycle.
- The requirement change can be accepted at the middle of the process.

Disadvantages:

- It is having the huge delivery risk, because of the aggressive time lines(less time).
- Cannot accept requirement change at the end stage of the project to avoid delivery risk.

3) V-Model(Verification and Validation model):

Validation:

It is also known as "*QC" (Quality control*). The testing team is responsible for validation. Testing team will check whether the developed software is as per the client's requirement or not.

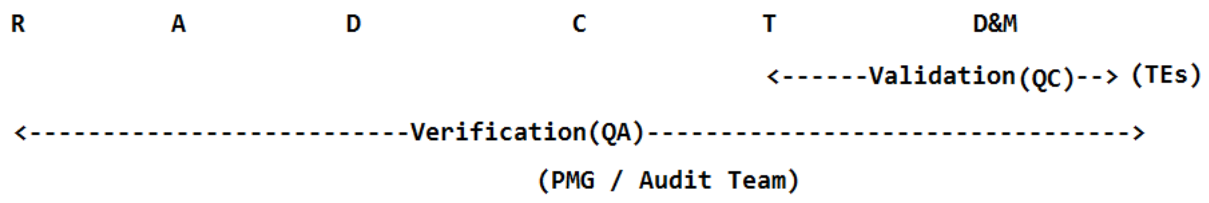
Test engineers are validators.

Verification:

Def1: Check whether each and every phase outcome document is as per the company and client guidelines or not.

Def2 **:** Check whether each and every role in the organization is working as per the Companies and clients guide lines or not. Verification is also known as *QA (Quality assurance).*

The project/Process management group (PMG) or audit group are responsible for verification.



```
R          A           D              C           T            D&M
                                                  <------Validation(QC)--> (TEs)
<-------------------------Verification(QA)------------------------------->
                                        (PMG / Audit Team)
```

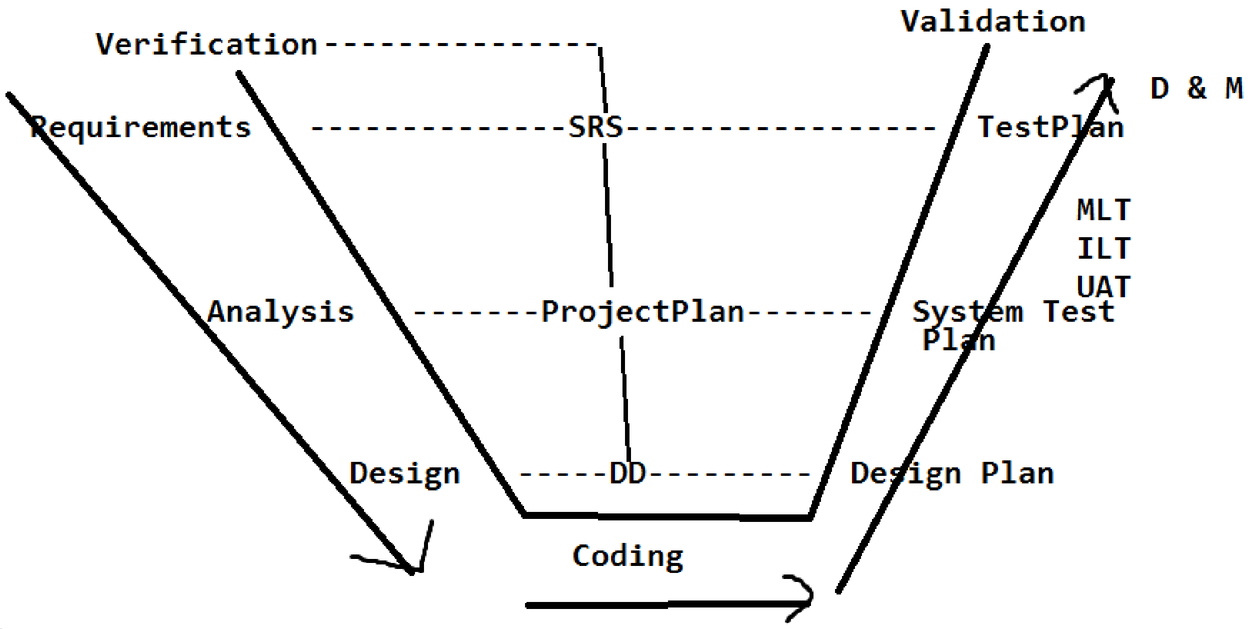
- From V-model onwards even the testing team will participate in collecting requirements.
- BA is responsible to collect the requirements, parallel the testing team will be analyzing all the requirements to check whether it is possible to test or not.
- Once the SRS is baselined, validation team is responsible for test plan
- Based on the analysis and design phases the validation team is preparing the system test plan and design plan.
- Once the code is developed they will release the build to the testing team where they will perform all the levels of testing. Once the build is stable it will be delivered to the client.

Advantages:

- The testing activities are started from the requirement phase onwards so that we can ensure for quality.
- For each and every phase the verification team and validation team will check the phases so that we can ensure for quality.
- The risk can be identified at the early stage because we have a parallel testing activity, so it can be prevented.
- We can accept the requirement change at the middle of the phase.

Disadvantages:

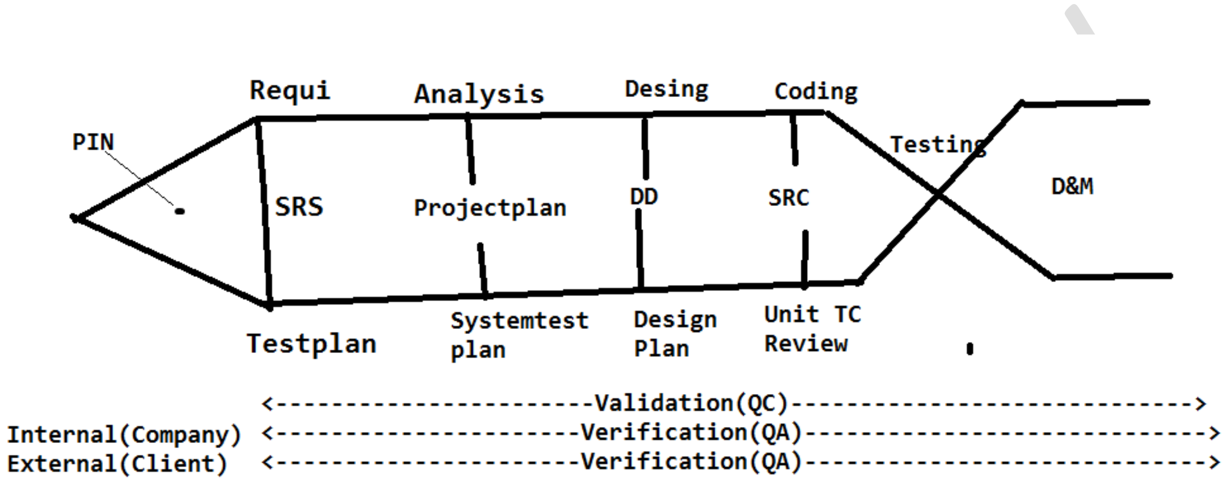
- It's a time consuming and costly process but we can ensure for quality.



4) Fish model:

- It is same like v-model.
- In fish model we will have multiple Verification teams from the client and company to check the process and to provide more quality and security.
- It is more expensive than v-model.
- It provides more security and is applied in high level security projects like NASA, Air force, Navy, Army etc.

Note: Validation team will review unit test cases results when the build is under development


Requi
Analysis
Desing
Coding
PIN
Testing
SRS
Projectplan
DD
SRC
D&M
Testplan
Systemtest plan
Design Plan
Unit TC Review
<----------------------Validation(QC)---------------------------->
Internal(Company) <---------------------Verification(QA)---------------------------->
External(Client) <---------------------Verification(QA)---------------------------->

5) <u>Agile process:</u>

- It is having multiple sub models like adaptive, Scrum, iterative model etc…The model which we are going to use is scrum model.
- It is an iterative and incremental model. Scrum model is having the below activities.
  - **a.** Scrum master
  - **b.** User stories
  - **c.** Scrum meeting/scrum calls/DSM
  - **d.** Sprint plan
  - **e.** Sprint meeting
  - **f.** Backlogs

a. Scrum master:

The scrum master is, who is going to lead the project. The project manager or the client will acts as a scrum master. Scrum master is responsible for scrum meetings and sprint meetings.

b. User stories:

The requirements will be captured in the form of end user used flows (end user used ways). Hence we will call it as *User stories*. BA is responsible to collect

c. Scrum meeting:

- On daily basis all the team members will participate in a quick meeting where they will describe what activities were performed yesterday and what tasks are planned to perform today and is there any challenges.
- All the team members including the scrum master and client have to describe.
- The main purpose of the scrum meeting is to track the resources and also to maintain the transparency.

d. Sprint plan:

- Sprint is fixed time period can be one week/two weeks/three weeks etc. It will be decided by the scrum master.
- Sprint plan is, to collect user stories, analyze, develop, test and deliver to the client.
- During the sprint if we are unable to complete any of the requirements the sprint won't be extended. And the pending requirements should be carried to the next sprint. Sprint is a fixed time period

e. Sprint meeting:

Once the sprint is completed the next sprint plan will be decided under the sprint meeting. They will discuss, if the current sprint is delivered successfully or not, is there any challenges faced.

f. Backlogs:

During the sprint plan if any user stories are unable to accomplish, those will be taken as Backlogs. These backlogs have to be completed in the next sprint.

It is of two types,

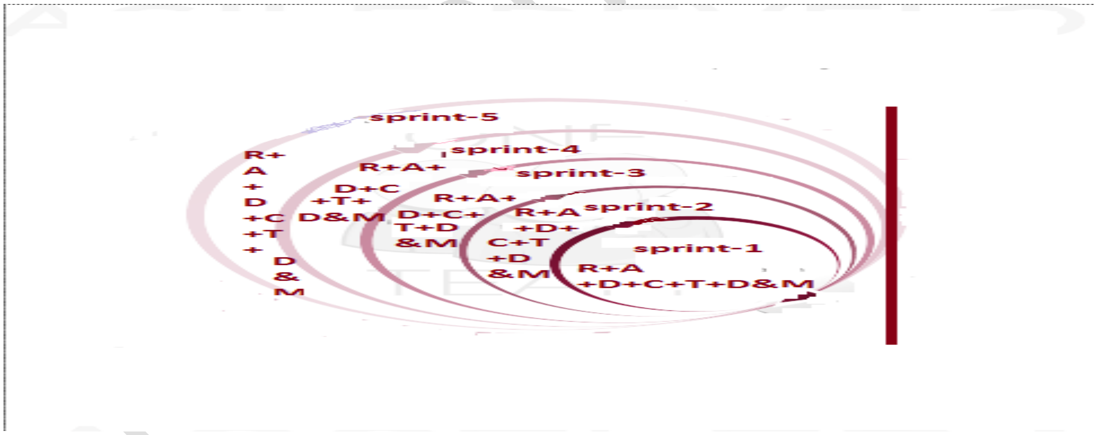
(i) Product Backlog
(ii) Sprint backlog

Product Backlog: The Requirements (user stories) which we are going to collect, develop, test and deliver it to the client as a part of sprint plan is known as *product backlogs.*

Sprint Backlog: The Requirements which are not completed as part of the sprint plan will be treated as sprint backlog.

Advantages:

- Each and every sprint will be tested multiple times by the testing team and client, so we can ensure for quality.
- All the phases in SDLC are performed parallel y so we can save time and cost.
- The requirement change can be accepted at any stage of the project (even after delivery of sprint).
- Risk can be identified at the early stage and it can be prevented
- We can maintain transparency of the project.
- The client will also participate in scrum meetings, so we can get the information very quickly.
- Each and every sprint is delivered to the client so we doesn't have delivery risk.
- We can gain the customer satisfaction by delivering all the sprints to the client.
- Sprint is also known as iterative. Its and iterative and incremental model



Disadvantages:

Maintaining all the sprint related information is a very challenging task, but we can overcome with the help of test management tools like Scrum wise, Quality center, JIRA and test link etc.

Functional testing types (or) Black box testing types:

1) Smoke testing:

- Once the build is developed and deployed it in any environment then the initial testing will be performed, that is known as smoke testing. Initially the development team will deploy the build in development environment, and perform smoke test. They will check each and every module related field is properly navigating their pages or not and checks the main functionality of the application. The objective of smoke test is to check whether the build is ready for further testing or not. The developer will concentrate on white box testing
- If all these fields are properly navigating to the related pages then they will conclude that smoke test is passed.

2) Sanity testing:

- Once the build is deployed in the test environment, the testing team will perform the smoke test in test environment. It is known as sanity testing.
- In sanity test the testing team will perform at least one round of the main flow functionality and check whether its properly working or not.
- If sanity test is passed then the testing team will execute all the test cases if it fails they will reject the build to the development team.

Ex for Main flow: Create an account in Gmail and login into that account and compose email and send it to one valid email and check that whether it is properly delivered or not.

Note: Once the sanity test is performed the testing team (test lead) has to send an email with sanity test results to the development team.

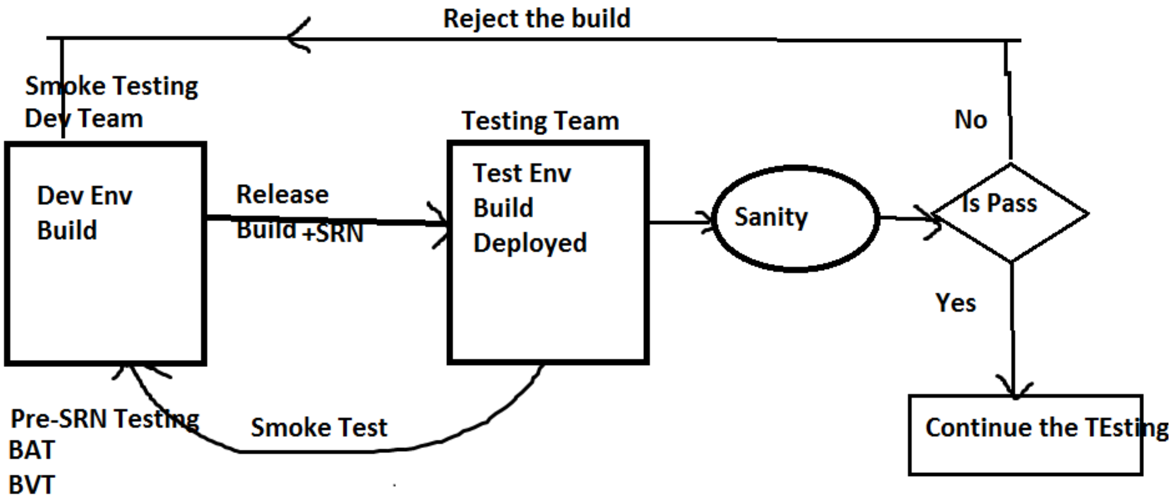
3) Pre SRN testing: SRN - Software Release Notes

- It contains the build status like, number of modules available in the build for testing.
- Number of modules which are under development.

- Number of stubs and drivers are available in the build.
- Number of bugs which are fixed and available in the build.
- Number of bugs which are under development
- Deployment guidelines etc.. the SRN document contains

  - Before releasing the SRN document along with the build to the testing team, the testing team will perform the smoke test in development environment.
  - It is known as *pre SRN testing*, it is also known as *Build acceptance testing (BAT) or Build verification testing (BVT).*

Note: Once the build is released to the testing team, the test engg's will review the SRN doc to know the build status (what build contains). Then the testing will perform sanity test.

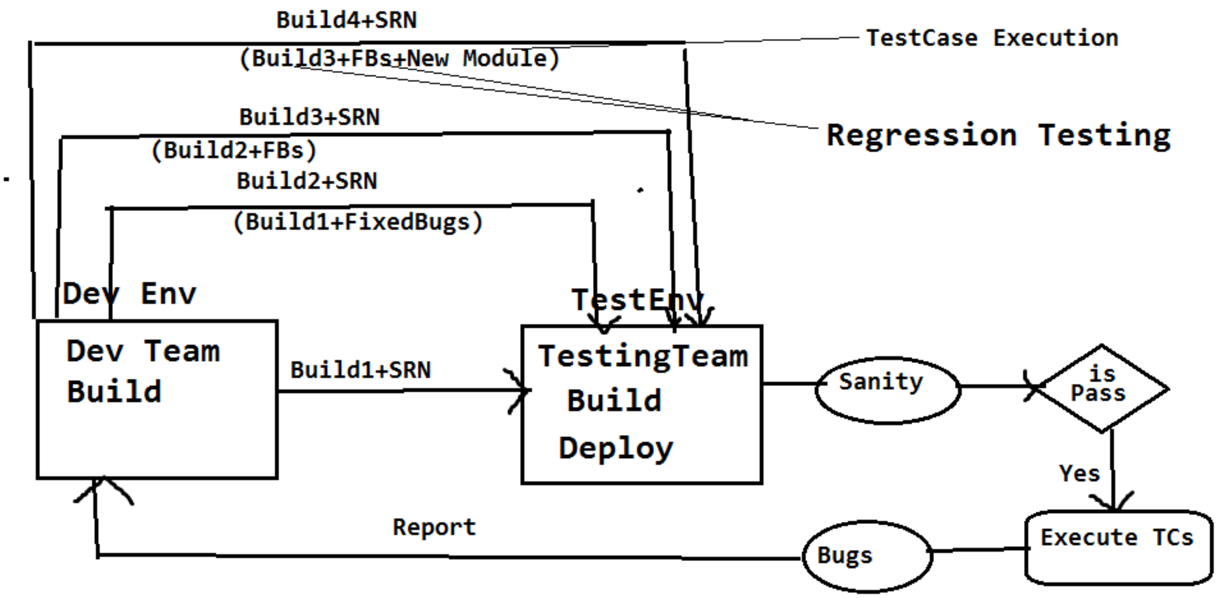


4) GUI/UI Testing:

Graphical user interface/user interface testing.The below five activities will be tested in GUI.

- Check the spellings of all the fields.
- Check the font of all the fields where it should maintain the consistency.
- Check the color and alignments of all the fields it should maintain the consistency.
- Check the overall look and feel of the page

5) Regression testing:



Performing testing on already tested functionalities on the iterative and incremental builds is known as '*Regression Testing'*.

It will be performed in two ways:

- Whenever any bug is identified it will be reported to the developer, the developer will fix it then he will releasing the fixed bug in the form of new build(Build2) to the testing team.

The test engineer will test again, to check whether the bug is really fixed or not.

- The test cases which are passed on the old build will be executed again on the new build and check that whether these are working same as previous or not.

Testing already tested functionalities is *regression testing.* Testing the new functionalities is not the regression testing. It comes under test case execution.

Note: If any code update is there, then that new code may affect the existing code, so we are performing the regression testing.

6) Retesting:

- Perform testing on the same functionalities again and again with multiple sets of different test data on the same build is known as '*Retesting*'.
- Executing the failed test cases on the iterative and incremental builds is also known as "Re testing".

Test data: The data which the testing team is using for testing is known as "*Test data*".

Ex: 1. Test the login functionality of Gmail with multiple sets of different credentials.

2. Test the spicejet one way search with the multiple sets of different origins and different passengers.

Q: What is the difference between Regression Testing and Retesting

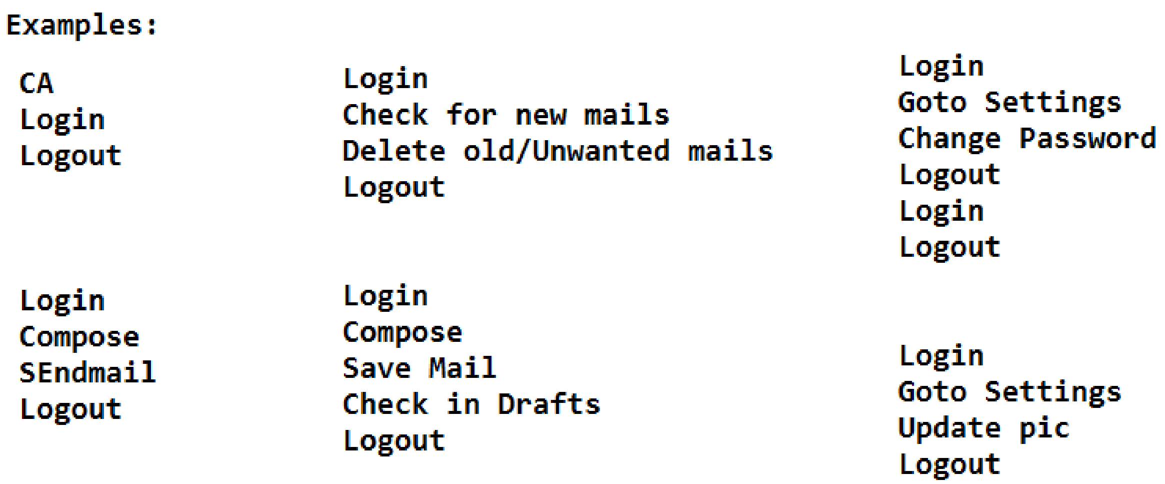
Q: What is the difference between Regression Testing and integration Level Testing

7) End to End Testing:

The test engineer has to identify all the end user used scenarios of the application, and check that whether the End to End Scenarios are properly working or not

By performing End To End testing we can achieve Integration level testing

Ex: The end to end scenario for Gmail.



```
Examples:

CA
Login
Logout

Login
Compose
SEndmail
Logout

Login
Check for new mails
Delete old/Unwanted mails
Logout

Login
Compose
Save Mail
Check in Drafts
Logout

Login
Goto Settings
Change Password
Logout
Login
Logout

Login
Goto Settings
Update pic
Logout
```

8) <u>Compatibility Testing</u>:

- Test whether the application is working same as expected in all the targeted environments or not is known as '*compatibility testing*'. Environment is combination of OS, Browser, Server, DB etc.
- Compatibility testing is two types '*cross browser testing*' and '*cross platform testing*'.
- Test whether the web application is working as expected in all the targeted browsers like firefox, safari, google chrome, IE etc. is known as '*cross browser testing'.*
- Test whether the desktop application is working as expected in different platforms or operating systems like windows, LINUX, MAC, Solaris etc. is known as '*cross platform testing'.*

<u>**Ex** for Cross browser testing:</u> Test whether the spicejet is working in the below environments or not.

Windows – Internet explorer, Firefox, Google chrome, Safari, Opera

Linux -   Internet explorer, Firefox, Google chrome, Safari, Opera

MAC -   Firefox, Google chrome, Safari, Opera

<u>**Ex** for Cross platform testing:</u> Test whether the skype is working in the below platforms or environments

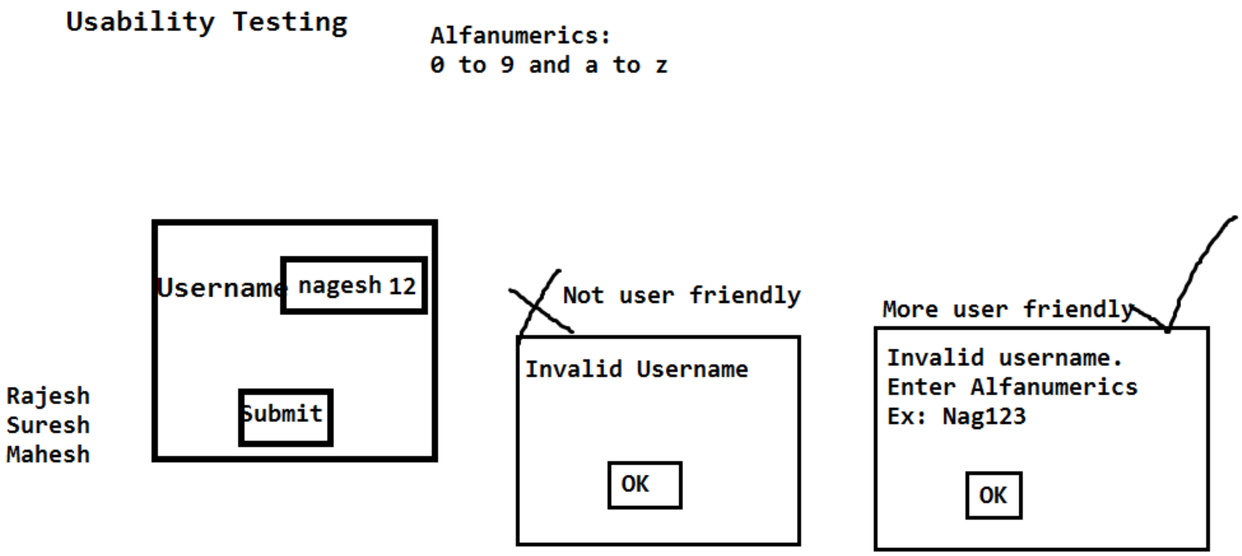
Windows
Linux
MAC and Mobile

<u>**Note:**</u> Whenever we are performing compatibility testing we need to concentrate more on GUI of the application

9) <u>Usability testing:</u>

- *Usability* means user friendliness. The test engineer has to provide usability to the application for the end user satisfaction.
- Depends on the application we have to provide the usability.

<u>**Ex:**</u> For Banking (secured) application we have to provide more security whereas for social sites (Face book, twitter)**,** we need to provide more user friendliness.



10) Adhoc Testing:

- Adhoc means in our own way.
- *Adhoc testing* means test the application in your own way, After understanding all the requirements and at least one round of manual testing is completed on the application
- The main purpose of adhoc testing is to provide usability to the application.

11) Exploratory Testing:

- *Exploratory* means identifying the new requirement / new Feature. Once the build is stable the domain experts will test the application as per their domain knowledge, while testing they will explore if the existing requirements are sufficient, if not they will provide the new requirements.
- The main purpose of exploratory testing is to provide usability and security to the application.

12) Monkey testing/Gorilla testing:

- Once the application is stable then we can go for monkey testing.
- Perform testing on the application by doing some abnormal actions is known as *Monkey/Gorilla testing*.
- Abnormal actions means continuously click on some field for long period of time and check that whether the application is crashing or not.

- Test the application with invalid data like HTML tags (<html>) and check that whether the application is crashing or not.
  Note: The objective of monkey testing is to check whether the application is crashing or not (Means will get server not found exception)

13) Static testing:

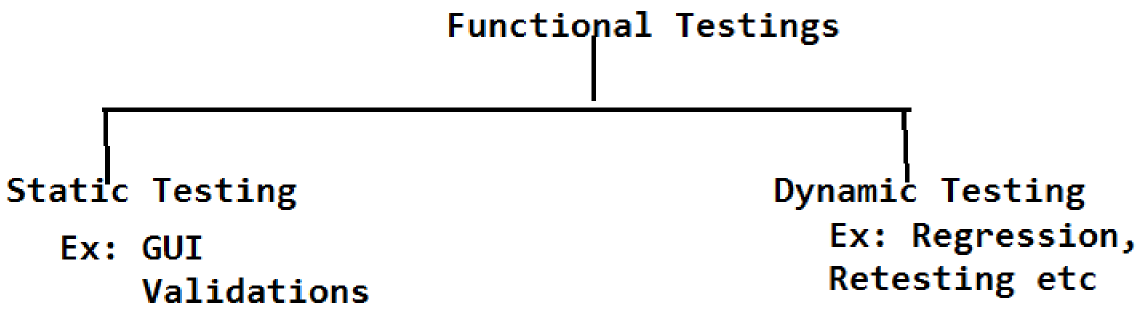
- Testing the application without performing any action is known as *static testing*.

  Ex: GUI testing and checking the availability of the fields in the page (Validations) comes under static testing.

14) Dynamic Testing:

- Test the application by performing some action is known as *dynamic testing*.

Ex: Regression testing, Retesting, Adhoc testing etc…



15) Authentication testing:

- Authentication means check whether the secured credentials/data is available in the database or not.
- *Authentication testing* means test the application with multiple sets of valid and invalid data, for valid data it should display the homepage, whereas for invalid it should display the proper authentication message (error message).

Ex: Test the login functionality of HMS with multiple sets of valid and invalid credentials. It has to authenticate the application properly.

16) **Direct URL testing:**

- Login into a secure page and take the URL of the secured page and access that URL in a new browser. Where it should not be accessible if it is accessible then the application is not secured.

Ex: Login into Gmail.com, copy the URL of the home page. Open in new browser and access the URL directly, where it should not be accessible.

17) **Firewall Leakage Testing**:

- Login into the application as one level of user and try to access the data beyond your role limitation. If it is accessible then we conclude that the application is not working as per the role (It is having the firewall leakage).

  Ex: Login into the HMS application as Jr. Doctor and try to access the data of Sr. Doctor, where it should not be accessible

18) **Database Testing:**

- Test whether the data is properly inserting into the database of all the tables or not is known as *database testing*. With the help of SQL queries we can perform DB testing.

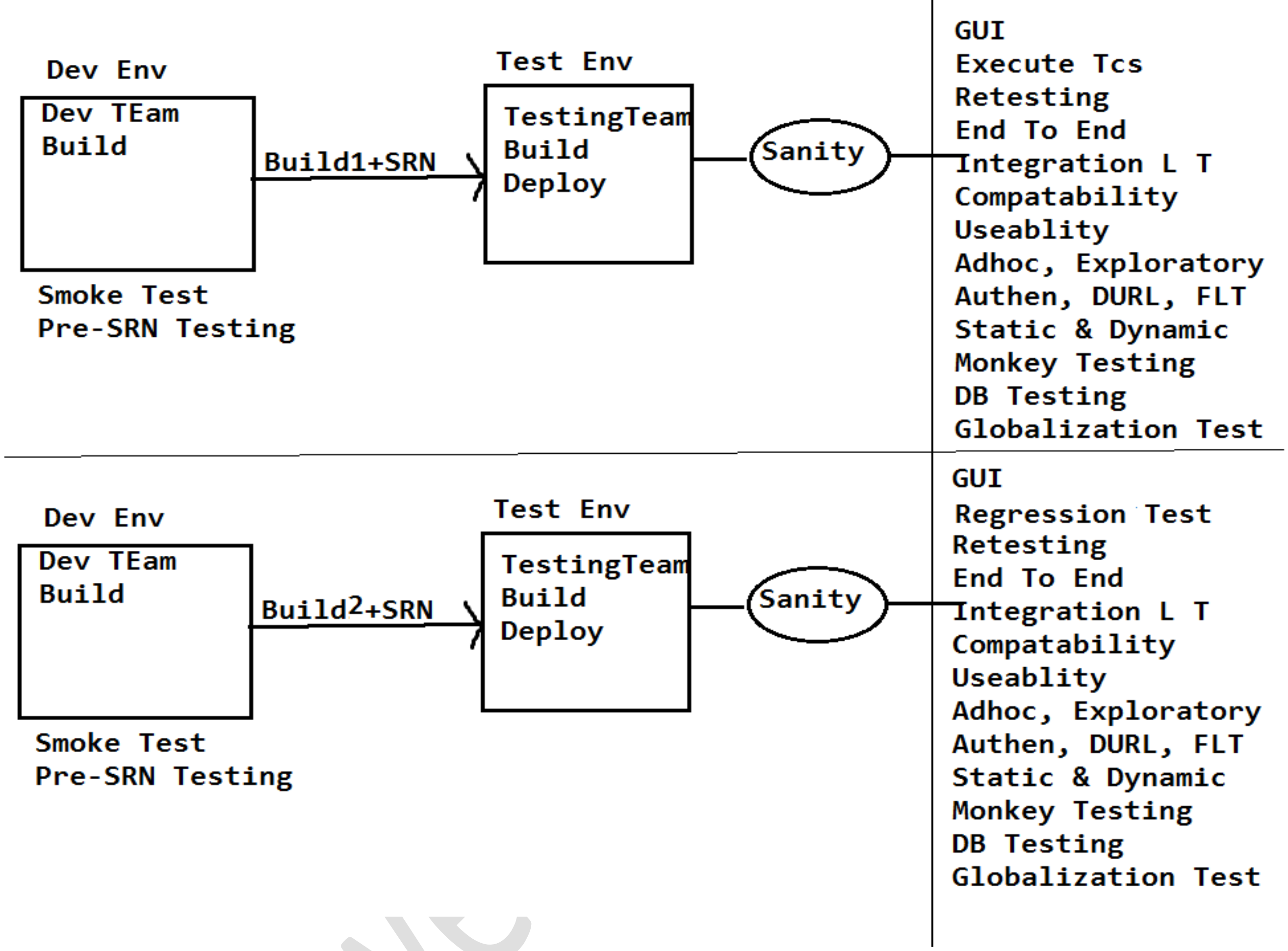
  Ex: Whenever we are creating the permanent registration in HMS, all the patient details will be stored in HMS database, as a test engineer we have to login into the database and check that whether the data is properly inserted in all the tables or not.

  Deployment Testing/ Installation Testing : The deployment team or Test lead will deploy the build in multiple environments like dev, testing, stage1,stage2, production etc and check that whether its properly deploying or not

Q: Once the build is released, how you will test the build

A: Initially we will perform sanity testing, if it is pass then we will execute all the test cases then will perform all the functional testing types as below. By performing all the below testing then we can ensure for quality for the application

Functional Testing types – Function testing's execution flow on the Build, once its released to testing team



Note: For any web application all the above testing's will be performed to ensure quality. If it is a desktop top application Direct URL Testing and cross browser testing is not possible to perform.

Review Report Template:

Review the SRS document of spice jet and provide the review report in the below template.

Requirement ID	**Requirement Description**	**Comments by TE**	**Comments**
7.	Adult, child and infant drop Downs should be available.	1. What is the difference between Child, adult and infant	

2. What values the adult, child, infant fields?

19) Globalization testing:

It is two of types

a. Localization testing.

b. Internationalization testing.

a. Localization testing:

- Test the application in all the local languages which are selective to our country like Hindi, Bengali, Telugu, etc. is known as *localization testing*.
- It supports maximum of 10 languages for single integration. Hence we will call it as ‘*L10N’ testing*.

Ex: 1. Test Google.co.in in all the local languages like Hindi, Bengali, Telugu etc…

2. Test the ATM machine in local languages likes Hindi, Telugu and English.

b. Internationalization testing:

- Test the application in all the international languages like Japanese, Chinese and Spanish etc. is known as *internationalization testing*. It supports maximum of 18 languages for a single integration. Hence we will call it as ‘*I18N’ testing*.

Ex: Test Gmail.com in all the international languages like, Japanese, Spanish and Chinese etc…

SOFTWARE TESTING LIFE CYCLE:

It contains the below phases:

1) Software test plan.
2) Software test design.
3) Test Execution.
4) Result analysis.
5) Reporting & BLC.
6) Delivery and maintenance.
7) Test summary report/ Build postmortem report.

1. Software test plan:

- Plan is a strategic document which describes how to perform a task in an effective and efficient way.
- Software test plan is also a strategic document which describes how to perform testing in an effective and efficient way. The test plan will be prepared by the test lead; once it is prepared it will be send to the testing team for review.
- Based on the test plan we are responsible to perform testing.
- It contains below activities or Index.

Test plan Index

1. Objective

    1.1 Scope of testing

2. Reference documents

3. Test Items

    3.1 Features to be tested

    3.2 Features not to be tested

4. Test strategy

    4.1 Testing types

        4.1.1 Functional testing types

5. Test environment

6. Test pass/Fail criteria

7. Defect analysis and closure

8. Test Deliverables

9. Automation testing

10. Risks and contingencies

11. Hardware and software requirements

12. Resource plan

13. Test summary report/ Build postmortem report.

1. Objective:

The main purpose of the test plan will be described here. It contains scope of testing.

1.1 Scope of testing:

What kinds of testing's the testing team is responsible to test on the application is known as scope of testing.

Ex: Testing team is responsible for manual testing and automation for the project.

2. Reference Documents:

The list of documents which the test lead used to prepare the test plan will be described here.

Test lead will use SRS documents to prepare the test plan.

3. Test Items:

3.1 Features to be tested:

The list of functionalities or modules which the team is responsible for, will be described here and also the list of testing's which the testing team is performing on the modules will be described here.

Ex: Testing team is responsible for Book a Flight, Book a hotel and manage my booking.

For the above modules they are responsible for manual testing and automation.

3.2 Features not to be tested:

The list of modules and testing's which the testing team is not responsible for will be described here.

Ex: Testing team is not responsible for payment modules and also they are not responsible for performance testing, Load testing, Stress testing.

4. Test strategy:

Strategy means the list of steps which we are going to take to accomplish the plan.

- The test strategy means the list of functional testing types. Which the testing team is going to take to perform testing is known as test strategy.

- We will perform all the functional testing types like regression, re testing, etc… on the application
- In short, plan means what to do. Strategy means how to achieve the plan.

5. Test Environment:

Environment means the system which we are going to use to deploy the build and to test the application is known as the test environment.

Ex:	Machine type	:	Windows server enterprise
	OS	:	Windows
	Processor	:	Intel Xeon CPU
	Memory	:	4GB/2.13 GHZ
	Hard disk	:	150GB
	Data base	:	Microsoft SQL server 2008 standard edition
	Web server	:	IIS 7.0
	Client	:	Microsoft internet explorer, Firefox, Google chrome

6. Test pass fail/criteria:

If any test case is deviating from the expected result then it will be treated as failure or bug.

Every bug is having the criteria or bug type.

It is of five types

a. Blocker
b. Very High
c. High
d. Medium
e. Low

7.Defect Analysis Closure:

At the time of delivering the build if any bugs/defects are available it will be analyzed by the testing team with project manager. If any bug is not necessary to be fixed then it will be closed.

8. Test Deliverables:

The list of modules which we are going to deliver to the client known as *deliverable*s.

All the modules will be divided into multiple phases and also the lead will be providing the targeted deadline (delivery date).

Phase No	Modules	Dead Lines (Date of Delivery)
1	1. BookaFlight 2. ManageMy Booking 3. PNR Status	30th Jun
2	4. Flight Schedules	31st July
3	5. Corporate Benefit 6. Spice connect	30th Sept

9. Automation testing:

The number of modules which the testing team is going to automate will be described here and also the automation tool and strategy which the test engineers are going to follow will be described here.

10. Risks and contingencies:

The list of risks which the team is going to face while executing the project and also with the related solution will be described here.

Risks	Contingencies
Resource shortfall	Maintain buffer resources
Continuous Requirement Changes	Analyze the requirements
Lack of peer reviews	Monitor Peer reviews

11. Hardware & Software requirements:

The number of machines like laptops, mobiles, printers etc… required for the testing with related software will be described here.

12. Resource Plan:

The numbers of resources required for manual testing, automation testing, database testing will be described here.

13. Test summary report /B uild postmortem report:

Once the testing is completed the test lead has to prepare the test summary report, it contains the summary of the testing.

2) Software Test Design:

The process of writing the test cases on the test case template after understanding all the requirements is known ‘*software test design*’.

- Every organization will be having their own template based on that template; the test engineer is responsible to write the test cases.
- We are having the below templates to write the test cases. It contains CoverSheet, Test cases, Testdata, Traceability matrix and Test Report

Require- Test ment Types ID	priority	TC ID	Test scenario	Pre-production	Test types	Actual Result	Expected Result	Result	Comments

Cover sheet:

Module name :

Total no. of test cases :

No. of P1 test cases :

No. of P2 test cases :

No. of P3 test cases :

No. of P4 test cases :

Requirement ID: The requirement number for which we are writing the test cases will be described here.

Test types: The test case type is known as test type. It is of five types.

- GUI
- Validation
- Positive test case (or) Functional positive test case.
- Negative test case (or) Functional negative test case
- Database test case

Positive test case: Test the application with all the valid data is known as positive test case.

Negative test case: Test the application with at least one invalid data is known as Negative test case.

Priority: It describes how important the test case is. It is below types: P1, P2,P3 and P4.

P1: If the test case is describing the main functionality of the application/module then it will be treated as P1.

The main functionality means if the test case failed we can't continue the testing further, so that priority is 'P1'.

P2: If the test case is describing the field level functionality then the priority is 'p2'.

Field level test case means, if it is failed we can continue the testing but it is important to be there in the application as per client requirement.

P3: All GUI test cases are comes under P3 priority.

P4: Test engineer is having the option to give the suggestion to the application. Those suggestions will be captured in the form of test cases and then the priority is 'P4'.

Test case ID: The serial number of the test case will be described here.

Test Scenario: Scenario means a flow or the end user used way. The requirement will be divided into all the end user used flows or scenarios and those will be described here. The test engg has to identify the maximum possible flows(Scenrios) for the requirement or user story

Pre-condition: The condition which is required to test the scenario will be described here.

Test Steps: The list of steps which are required to execute the scenario will be described here. Based on the test steps the test engg will execute on the application or build

Expected Result: At the time of writing the test cases we won't be having the application with us. So we will be expecting the result for the scenario. That expected result will be updated in the expected result column.

Actual Result: It will be updated at the time of executing the test cases. The test engineer will observe the actual behavior of the application for the scenario and it will be updated here.

Result: Once the test execution is completed the test engineer will compare the actual result with expected result, if both are matching then he will update the result as pass, if not he will update it as fail.

Comments: The BA or client will provide the comments here.

Refer Gmail login TCs Document

Test design techniques:

To perform testing in an effective and efficient way, we need to follow the below test design techniques.

1. Boundary value analysis (BVA)

2. Equivalence class partition (ECP)

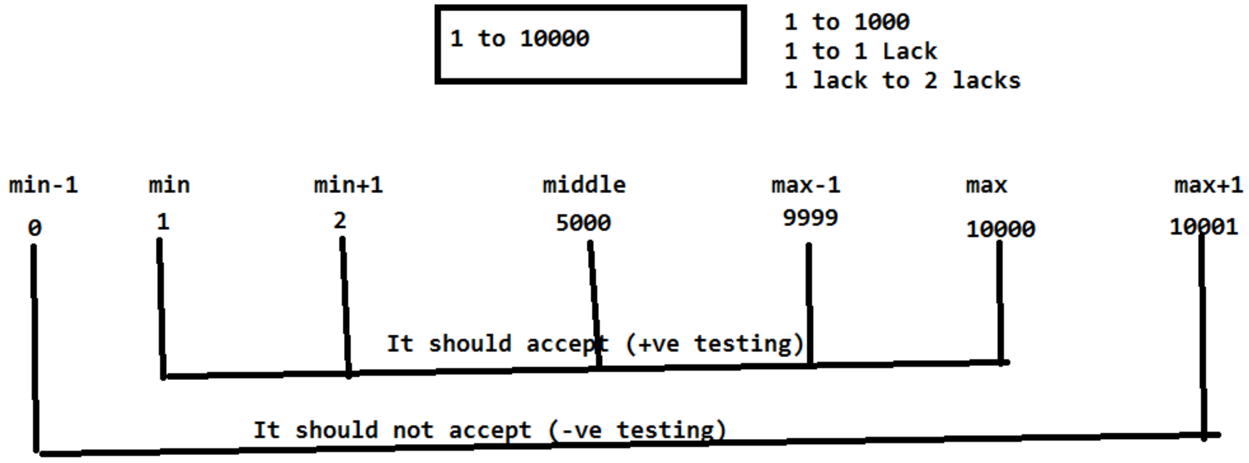
3. Error Guessing

1. Boundary value analysis (BVA):

Whenever we have the requirement to test the range like 1 to 100 or 1 to 1000 or 1 to 1 lack or 1 lack to 2 lacks then it is not possible to perform the exhaustive testing (complete testing). So we need to apply the BVA technique.

- Divide the range to multiple boundaries like min-1, min, min+1, middle, max-1, max and max+1.

- To perform the positive testing, test the field with min, min+1, middle, max-1, and max. Where it should accept. (Its +ve Test case)
- To perform negative testing, test the field with min-1 and max+1. Where it should not accept. (Its -ve Test case)
- If it is working as per the above then we can conclude that it is accepting only the range.



+Ve Test Scenario: Verify the field with the boundaries like min, min+1, middle, max-1, and max

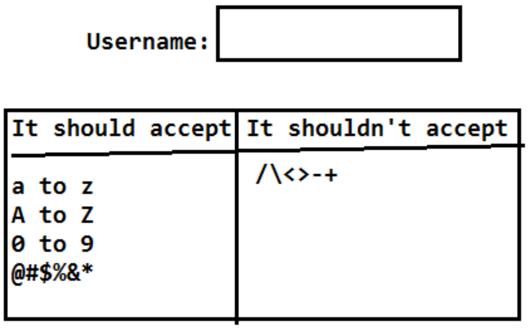
-Ve Test Scenario: **Verify the field with the boundaries like** min-1 and max+1

2. Equivalence class partition(ECP):

Whenever we are having the special requirement like check whether the field (user name or password) is accepting the characters like a to z, A to Z, 0 to 9 and #%@$&*. At the same time the field should not accept the special characters like <>-+/\.

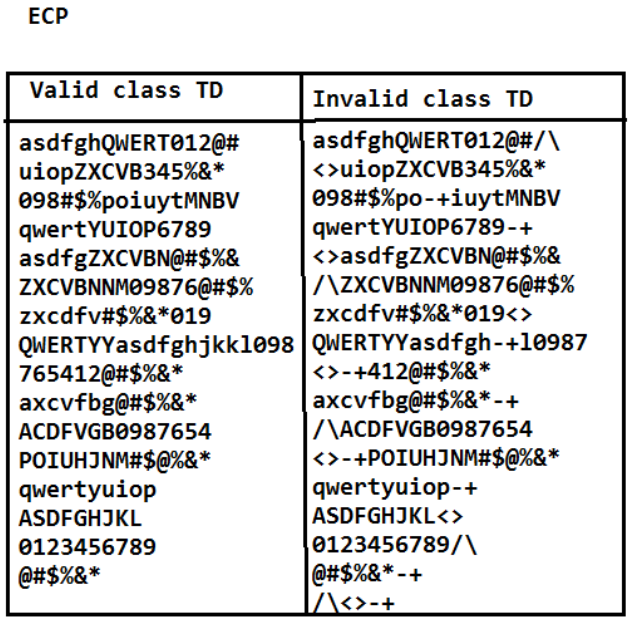
- In this scenario it's not possible to perform the exhaustive testing with all the characters. So we need to follow the ECP technique.
- Divide equally the test data into two classes.
  a. Valid test data class   b. Invalid test data class
- Prepare the test data with all the possible ways.
- To perform positive testing, tests the field with valid test data. Where it has to accept. (Its +ve Test case)
- To perform negative testing, tests the field with invalid test data. Where it should not accept. (Its -Ve Test case)

- If it is working as expected in the above we can conclude that it working as per the requirement.



Username:

It should accept	It shouldn't accept
a to z	/\<>-+
A to Z	
0 to 9	
@#$%&*	



ECP

Valid class TD	Invalid class TD
asdfghQWERT012@#	asdfghQWERT012@#/\
uiopZXCVB345%&*	<>uiopZXCVB345%&*
098#$%poiuytMNBV	098#$%po-+iuytMNBV
qwertYUIOP6789	qwertYUIOP6789-+
asdfgZXCVBN@#$%&	<>asdfgZXCVBN@#$%&
ZXCVBNNM09876@#$%	/\ZXCVBNNM09876@#$%
zxcdfv#$%&*019	zxcdfv#$%&*019<>
QWERTYYasdfghjkkl098	QWERTYYasdfgh-+l0987
765412@#$%&*	<>-+412@#$%&*
axcvfbg@#$%&*	axcvfbg@#$%&*-+
ACDFVGB0987654	/\ACDFVGB0987654
POIUHJNM#$@%&*	<>-+POIUHJNM#$@%&*
qwertyuiop	qwertyuiop-+
ASDFGHJKL	ASDFGHJKL<>
0123456789	0123456789/\
@#$%&*	@#$%&*-+
	/\<>-+

3) Error guessing:

Whenever any bug is identified by the test engineer then it should be reported to the developer where he will fix it and send it back to the testing team. The test engineer will check if the bug is really fixed or not. At the same time he has to guess the errors or bugs in the related functionality. He has to perform the testing in the related functionality also. It is known as ‘Error guessing’.

Ex: In PR page alert msg is not displaying, it was fixed by developer and tested by the tester. Where the alert msg is properly working in PR page. Now the test engg.. has to go the related functionalities like Admission advice and admission then look (guess) for the similar kind of bug.

Traceability Matrix:

It is to track back whether the test engineer has covered all the test cases for the entire requirements or not.

Based on the traceability matrix the lead or the client will track whether the test engineer has covered all the test cases or not.

Req ID	No of TCs	Test case id	Comments
1	1	1	
2	1	2	
3	1	3	
4	1	4	
5	1	5	
6	1	6	
7	1	7	
8 & 9	5	8 to 12	
10		Not yet Implemented	Requirement is not clear. Waiting for BA comments

4) Test Execution:

- The process of executing the test cases on the build in test environment is known as test execution. Whenever the build is released to the testing team the test engineer has to review the SRN document to know the build status.
- Based on the SRN document the test lead will deploy the build and the testing team will perform sanity test.
- Once the sanity test is completed, the sanity test results are mailed to the developer.
- If sanity test is pass the testing team will continue to execute the test cases, if sanity test is failed, the testing team will reject the build back to the development team.
- While executing the test cases the test engineer will observe the actual behavior of the application for the scenario and it will be updated under the actual result field. The same will be continued for all the test cases.

5) Result analysis:

- While executing the test cases the test engineer will update the actual result field then he will compare the actual result with expected result, if both are matching then he will provide the result as pass else he will update as fail.
- For pass we will give the green color, whereas for the fail we will provide the red color. Test execution and result analysis, both are parallel process.

Note: Once the test cases execution is completed we are responsible to execute all the types of functional testing's on the application to identify the bugs.

*** How many test cases can we write in a day?**

It all depends on all the requirements and test engineer but an average we can write around 40-50 test cases in a day. It means we are taking approximately 8-10 minutes for one test case to analyze the requirement and to prepare the test case on the test case template with the test data.

*** How many test cases we can execute in a day?**

It also depends on the test cases and the application but on an average we can execute 50-60 test cases in a day because to review the test case and execute it on the application.

We are taking around 5-8 minutes to execute one test case on an average.

6) Reporting:

- The process of reporting/sending the bugs (failed test cases) to the developer is known as Reporting.
- It is two types.

1. Reporting the Bugs by using XL files.
2. Reporting the bugs by using reporting tools.

1. Report the bugs by using XL file:

It was the old process we used to have the below template to update the bug and send it to the developer.

Bug ID	Bug title/ Summary	Status	Severity	Priority	Bug description	Screenshot	Build Version	Reported By	Assigned to	Comments

1.					

Bug ID:

The serial number of the bug will be described here.

Bug title/Summary:

The actual result of the bug will be described here.

Status:

Based on the bug the test engineers as well as the developer are responsible to give the status. It is below of types.

- **New:**
  Whenever the test engineer identifies any bug. Initially the status of the bug is New. The new bug will be reported to the developer.
- **Open:**
  The developer will check that whether the new bug is really a bug or not. If yes then we will update the status from new to open.
- **Fixed/Verified:**
  Developer will take some time to fix the open bug once it is fixed he will update the status from open to fixed. Fixed bug will be sent to the test engineer.
- **Closed:**
  Test engineer will check whether the fixed bug is really working as expecting or not. If it is working then we will update the status from fixed to closed. Closed state is the end of the Bug.
- **Re open:**
  The fixed bug will be tested by the test engineer; if it is not working as expected then he will update the status from fixed to reopen and the reopen bug will be sent back to the developer.

The developer will check that whether it is really a bug or not, if yes he opens it, fix the bug and send it back to the testing team.

- **Rejected/Not a Bug/Hold/Differed:**
  When the test engineer identified any bug it will be reported to the developer with new status. If developer is not accepting a bug then he will update a status from new to Rejected/Not a bug and it will be sent back to the testing team.

Severity:

It describes how seriously the bug impacted the application on testing. Severity means seriousness of the bug. It is below types.

- **Blocker:**
  If the bug is blocking the entire testing of the module then the severity or type of a bug is Blocker.
- **Very high:**
  If the bug is blocking partially the testing of the module then the severity of the bug is very high.
- **High:**
  If the bug is blocking only the specific scenario of the module then the severity is high.
- **Medium:**
  All GUI bugs severity is medium.
- **Low:**
  Test engineer is having the option to give the suggestion also. So the suggestion will be reported in the form of bug, where the severity is low.

Priority:

Priority describes in which order the bug has to be fixed by the developer. Based on the severity the test engineer will provide priority to the bug as below

Severity	Priority
Blocker/Urgent/critical	P1
Very high	P2
High	P3
Medium	P4
Low	P5

Description:

The detailed steps to produce/get the bug will be described here. Based on the steps developer will check that whether it is really a bug or not.

Screenshot:

The test engineer will capture the screenshot of the bug and it will be uploaded in the bug template. It is to prove the reported bug is valid and also to understand about the bug.

Build version:

The build number on which the test engineer identified the bug will be described here.

Reported by:

The test engineer who identified the bug will describe here.

Assign to: The developer name or the developer lead name, who is going to fix the bug will be described here.

Comments:

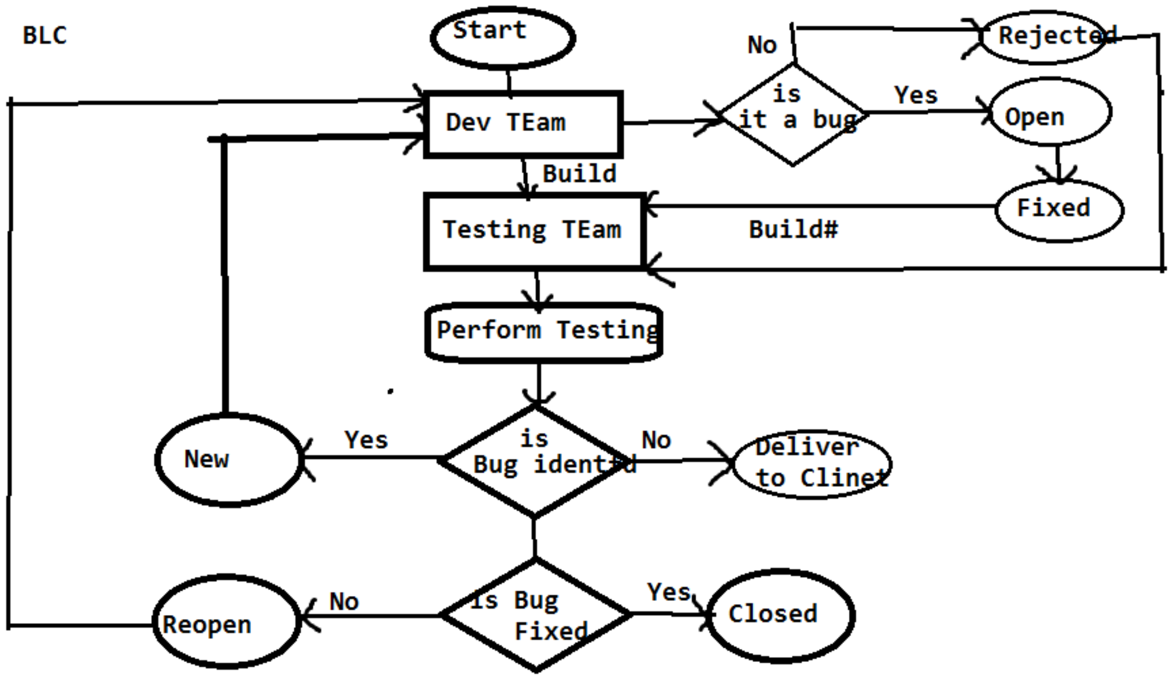
Both test engineers, developer can ask the questions in the form of comments.

Note: The XL file reporting will consume lots of time so our plan is to use the reporting tools.

Ex:

Phases	BUG ID	Bug Title / Summary	Status	Severity	Priority	Bug Description	Screenshot
I	1	Application is not displaying both date pickers	New	Blocker	P1	1. Open Spicejet.com 2. Click on Roundtrip radio button 3. Application is not displaying both the date pickers	D:\Nagesh\SPicejet_
II	2	Spicejet name is displaying as spacejet	New	Medium	**P4**	1. Open Spicejet.com 2. Observe the Spicejet logo 3. Its displaying as spacejet	D:\Nagesh\SPicejet_
III	3	Oneway radio button is not displaying	New	VeryHigh	P2	1. Open Spicejet.com 2. Oneway radio button is not available	Path
I	4	Student check box is not available	New	High	P3	1. Open Spicejet.com 2. Student check box is not available	Path
II	5	Cheange the color of spicejet home page to blue	New	Low	P5	1. Open Spicejet.com	Path
III	6	Spicejet club link is not navigating to Spicejet club page	New	Blocker	**P1**	1. Open Spicejet.com 2. Click on Spiceje Connect link 3. Spiceje Connect link is not navigating to MySpicetrip page	Path
I	7	Application is not maintaing GUI	New	Medium	P4	1. Open http://selenium4testing.com/hms 2. Login into the application 3. Click on Search Registration 4. Application is not maintaining GUI	Spicejet_GUI.png

II	8	Admission worklist GUI is not maintaining properly	New	Medium	p4	1. Open http://selenium4testing.com/hms 2. Login into hms with user1/user1 3. Click on ADT 4. Click on Admission worklist 5. Observe the GUI, its not maintaining properly	D:\Nagesh\hms_AD
III	9	Adult field is not displaying on the page	New	Blocker	P1	1. Open http://spicejet.com 2. Observe all the fields 3. Adult dropdown is not available	D:\Nagesh\Spicejet\
I	1	Application is not displaying Hyderabad and Bangalore	New	VeryHigh	P2	1. Open http://spicejet.com 2. Click on LeavingFrom field 3. Application is not displaying Hyderabad and Bangalore	D:\Nagesh\Spicejet\



Q: what is the difference between severity and Priority

Q: what is the difference between Priority in test cases and Priority in bug template

Q. If the developer is not accepting your bug then how you will prove that yours is valid bug?

A: Based on the bug description, SRS document, screenshot we will try to prove that the bug is valid if is not accepting it then I will take the sever log to prove the bug is valid, if still not accepting it and then I will send it to a BA, project manager and finally client.

Q. Explain the scenario where the bug is having high severity with low priority and low security with high priority?

A:	**Severity**		**Priority**	
High security	Blocker	-	P1	High priority
	Very high	-	P2	
	High	-	P3	
Low security	Medium	-	P4	Low priority
	Low	-	P5	

We have two bugs one is Blocker another one is medium. The blocker will be having high priority and medium will be having low priority.

Based on the severity the test engg.. will provide priority. Based on the priority the dev team is responsible to fix

But the development lead is having the option to change the priority, depends on the situation.

- The bugs which are related to the current phase delivery will be converted to high priority irrespective of the severity.
- The bugs which are not part of the current delivery will be converted to low priority irrespective of the severity.

Phase	Bug Id	Bug title/Summary	Status	Severity	Priority
I.	**1.**	Spice jet name is displaying As space jet	New	Medium	P4---P1
2.	2.	Spice jet connect link is not Navigating spice jet connect page	New	Blocker	P1----P4

Test Report/Build status report:

Once the test case execution is completed on the build then test engineer is responsible to send a test report to the lead as well as client. It below format

Build status Report/Test Report

Build Status Report / Test Report	
Test Engg Name:	
Build No	1
Login Credentials	
Browser	FF, IE GoogleChrome
Test Matrics	
Total no of testcases	200
No of Test cases executed	150
No of Test cases pending	50
No of Test cases Pass	100
No of Test cases Fail	50
No of Test cases Skipped	10
No of Bugs Reported	3

Test Metrics:

Metrics means measurement of the task. Test metrics means measurement of the testing.

Pending:

If the developer has not giving functionality at all then those test cases can't be executed. It's comes under pending.

Skipped:

Developer has given the functionality, but we are unable to test the functionalities, because of the dependent functionalities failed.

Ex: if Login failed, we can't execute compose.

Compose test cases comes under skipped.

- The reporting will be continued until the build is stable, stable means majority of the test cases are pass and no blocker bugs are available in the reporting tool.
- The stable build will be delivered to the client.

Q: Explain me the reporting structure in your organization

Report the Bugs By using Reporting Tools:

- Any reporting tool having two kinds of users: One is admin user and another one is End user.
- Admin user: The admin user is responsible to create the project, create users like test engineers, developers…etc. He will assign the user to the project
- End user: He is responsible to use (report) or receive the bugs ex: test engineers, developers…etc.

Ex: QC, JIRA, Bugzilla, Redmine, Test manager etc…

BugZilla:

- Access the Bugzilla by using selenium4testing.com
- Then click on Bugzilla.
- Login into the Bugzilla as a test engineer (jan30selenium@gmail.com & password : selenium)
- By using Bugzilla we can perform below activities.
  a. Reporting a Bug.
  b. Search for the bugs.
  c. We can take the report.
  d. Preference.

Introduction to Bugzilla

What is Bugzilla?

Bugzilla is an open-source issue/bug tracking system that allows developers effectively to keep track of outstanding problems with their product. It is written in Perl and uses MYSQL database.

Bugzilla is a defect tracking tool, however it can be used as a test management tool as such it can be easily linked with other test case management tools like Quality Center, Testlink etc.

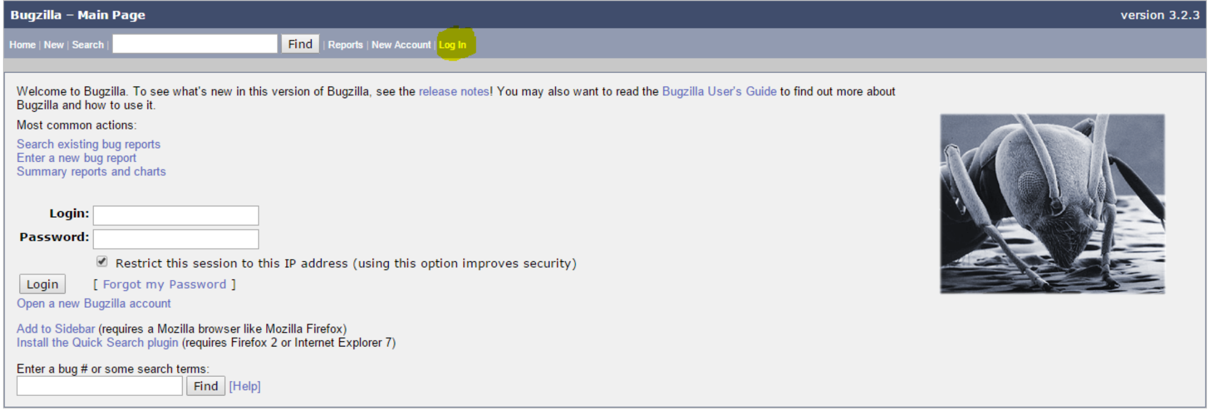
This open bug-tracker enables users to stay connected with their clients or employees, to communicate about problems effectively throughout the data-management chain.

Key features of Bugzilla includes

- Advanced search capabilities
- E-mail Notifications
- Modify/file Bugs by e-mail
- Time tracking
- Strong security
- Customization

How to log-in to Bugzilla?

Step 1) To create an account in Bugzilla or to login into the existing account go to **New Account or Log in** option in the main menu.



Welcome to Bugzilla. To see what's new in this version of Bugzilla, see the release notes! You may also want to read the Bugzilla User's Guide to find out more about Bugzilla and how to use it.

Most common actions:

Search existing bug reports
Enter a new bug report
Summary reports and charts

Login:
Password:

☑ Restrict this session to this IP address (using this option improves security)

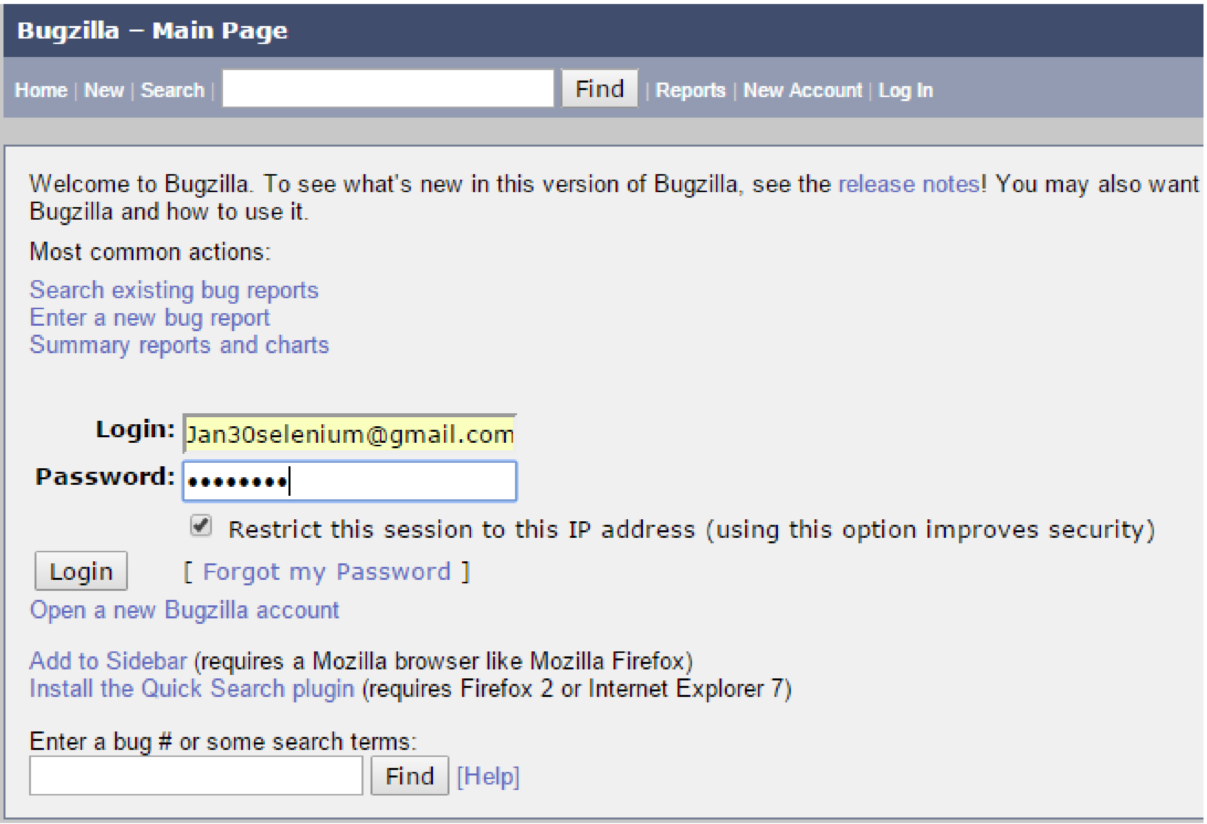
Login [Forgot my Password]
Open a new Bugzilla account

Add to Sidebar (requires a Mozilla browser like Mozilla Firefox)
Install the Quick Search plugin (requires Firefox 2 or Internet Explorer 7)

Enter a bug # or some search terms:
Find [Help]

Step 2) Now, enter your personal details to log into Bugzilla

1. User ID: jan30selenium@gmail.com
2. Password: selenium
3. And then click on **"Log in"**



Welcome to Bugzilla. To see what's new in this version of Bugzilla, see the release notes! You may also want Bugzilla and how to use it.

Most common actions:

Search existing bug reports
Enter a new bug report
Summary reports and charts

Login: Jan30selenium@gmail.com
Password: ••••••••

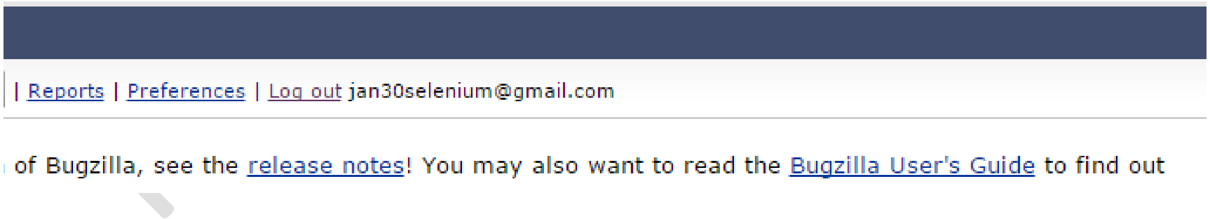
☑ Restrict this session to this IP address (using this option improves security)

Open a new Bugzilla account

Add to Sidebar (requires a Mozilla browser like Mozilla Firefox)
Install the Quick Search plugin (requires Firefox 2 or Internet Explorer 7)

Enter a bug # or some search terms:

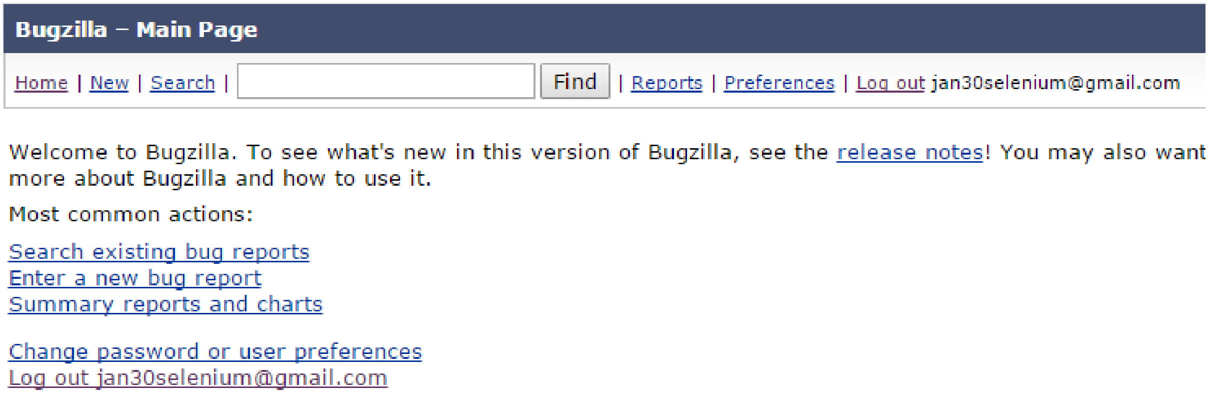
Step 3) You are successfully logged into Bugzilla system



of Bugzilla, see the release notes! You may also want to read the Bugzilla User's Guide to find out

Creating a Bug-report in Bugzilla

Step 1) To create a new bug in Bugzilla, visit the home-page of Bugzilla and click on **NEW** tab from the main menu



Bugzilla – Main Page

Home | New | Search | Find | Reports | Preferences | Log out jan30selenium@gmail.com

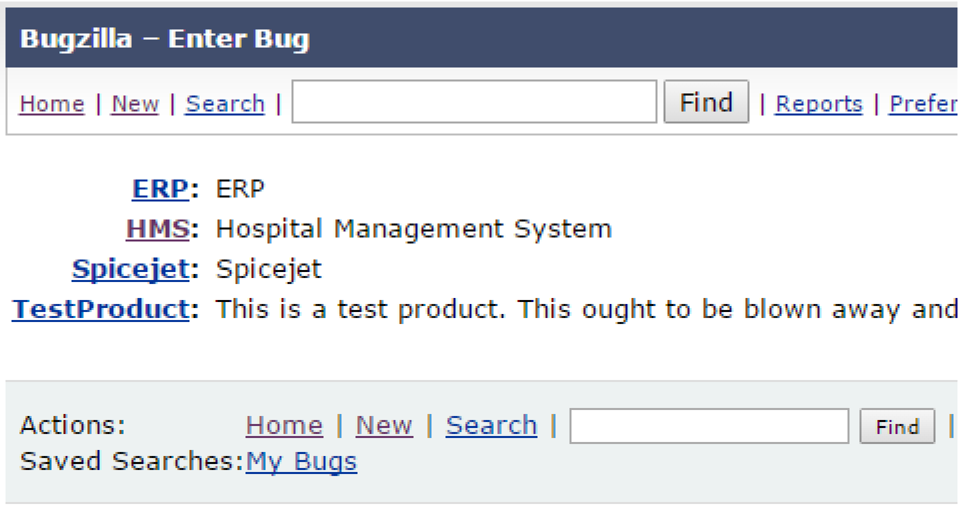
Welcome to Bugzilla. To see what's new in this version of Bugzilla, see the release notes! You may also want more about Bugzilla and how to use it.

Most common actions:

Search existing bug reports
Enter a new bug report
Summary reports and charts

Change password or user preferences
Log out jan30selenium@gmail.com

Click on the link New then the application opens the next window as below.



Bugzilla – Enter Bug

Home | New | Search | Find | Reports | Prefer

ERP:	ERP
HMS:	Hospital Management System
Spicejet:	Spicejet
TestProduct:	This is a test product. This ought to be blown away and

Actions: Home | New | Search | Find |
Saved Searches: My Bugs

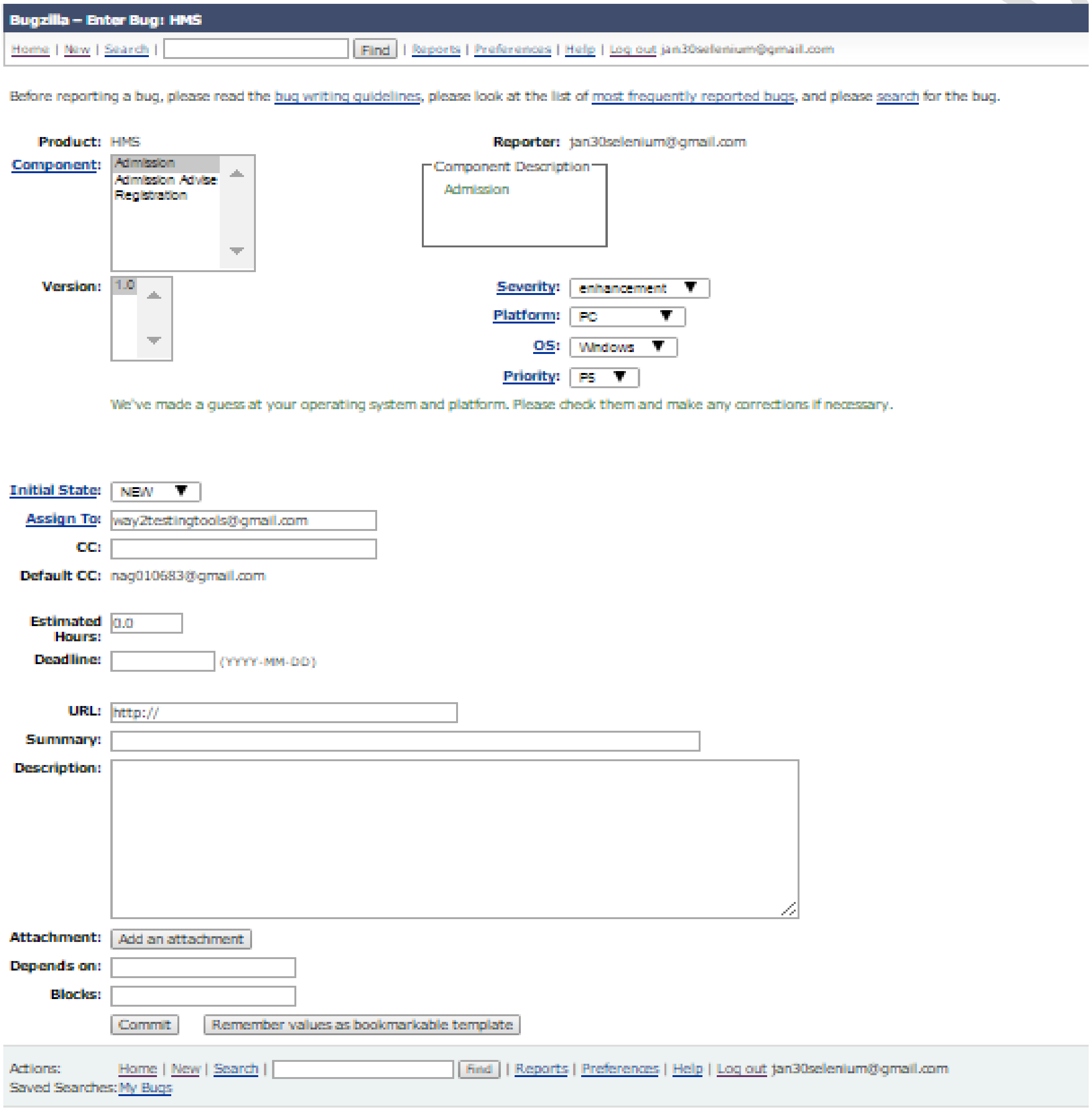
Step 2) In the next window

Click on the project name like HMS then the application opens a new window with following options

1. Enter Product
2. Enter Component
3. Give Component description
4. Select version,
5. Select severity
6. Select Hardware

7. Select OS
8. Enter Summary
9. Enter Description
10. Attach Attachment
11. Submit

NOTE: The above fields will vary as per your customization of Bugzilla



Bugzilla – Enter Bug: HMS

Home | New | Search | Find | Reports | Preferences | Help | Log out jan30selenium@gmail.com

Before reporting a bug, please read the bug writing guidelines, please look at the list of most frequently reported bugs, and please search for the bug.

Product: HMS

Reporter: jan30selenium@gmail.com

Component: Admission
Admission Advise
Registration

Component Description
Admission

Version: 1.0

Severity: enhancement

Platform: PC

OS: Windows

Priority: P5

We've made a guess at your operating system and platform. Please check them and make any corrections if necessary.

Initial State: NEW

Assign To: way2testingtools@gmail.com

CC:

Default CC: nag010683@gmail.com

Estimated Hours: 0.0

Deadline: (YYYY-MM-DD)

URL: http://

Summary:

Description:

Attachment: Add an attachment

Depends on:

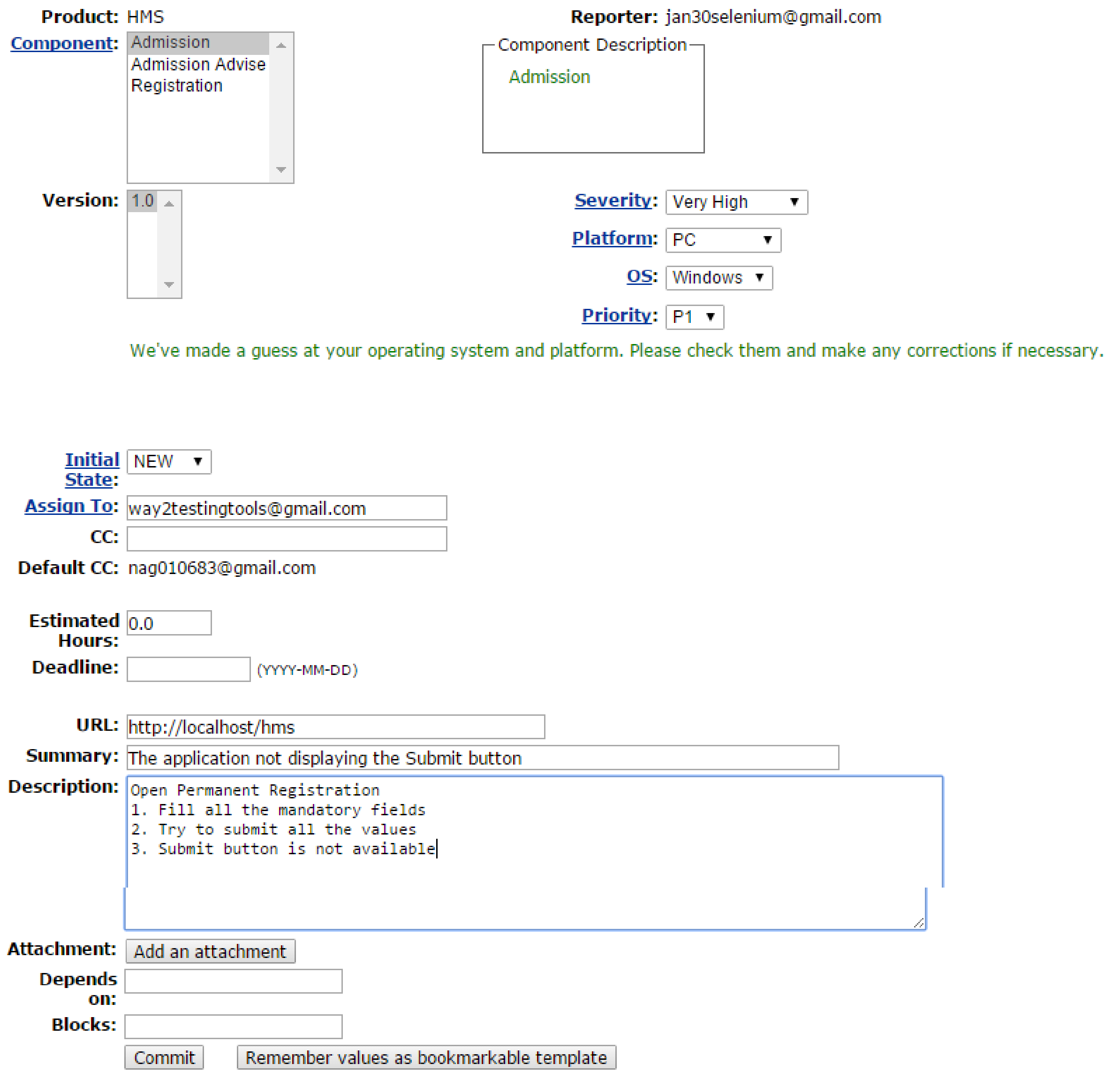
Blocks:

Commit | Remember values as bookmarkable template

Actions: Home | New | Search | Find | Reports | Preferences | Help | Log out jan30selenium@gmail.com

Saved Searches: My Bugs

Enter all the necessary fields regards your bug as below,



Product: HMS

Reporter: jan30selenium@gmail.com

Component: Admission
Admission Advise
Registration

Component Description

Admission

Version: 1.0

Severity: Very High ▼

Platform: PC ▼

OS: Windows ▼

Priority: P1 ▼

We've made a guess at your operating system and platform. Please check them and make any corrections if necessary.

Initial State: NEW ▼

Assign To: way2testingtools@gmail.com

CC:

Default CC: nag010683@gmail.com

Estimated Hours: 0.0

Deadline: (YYYY-MM-DD)

URL: http://localhost/hms

Summary: The application not displaying the Submit button

Description:

```
Open Permanent Registration
1. Fill all the mandatory fields
2. Try to submit all the values
3. Submit button is not available
```

Attachment: Add an attachment

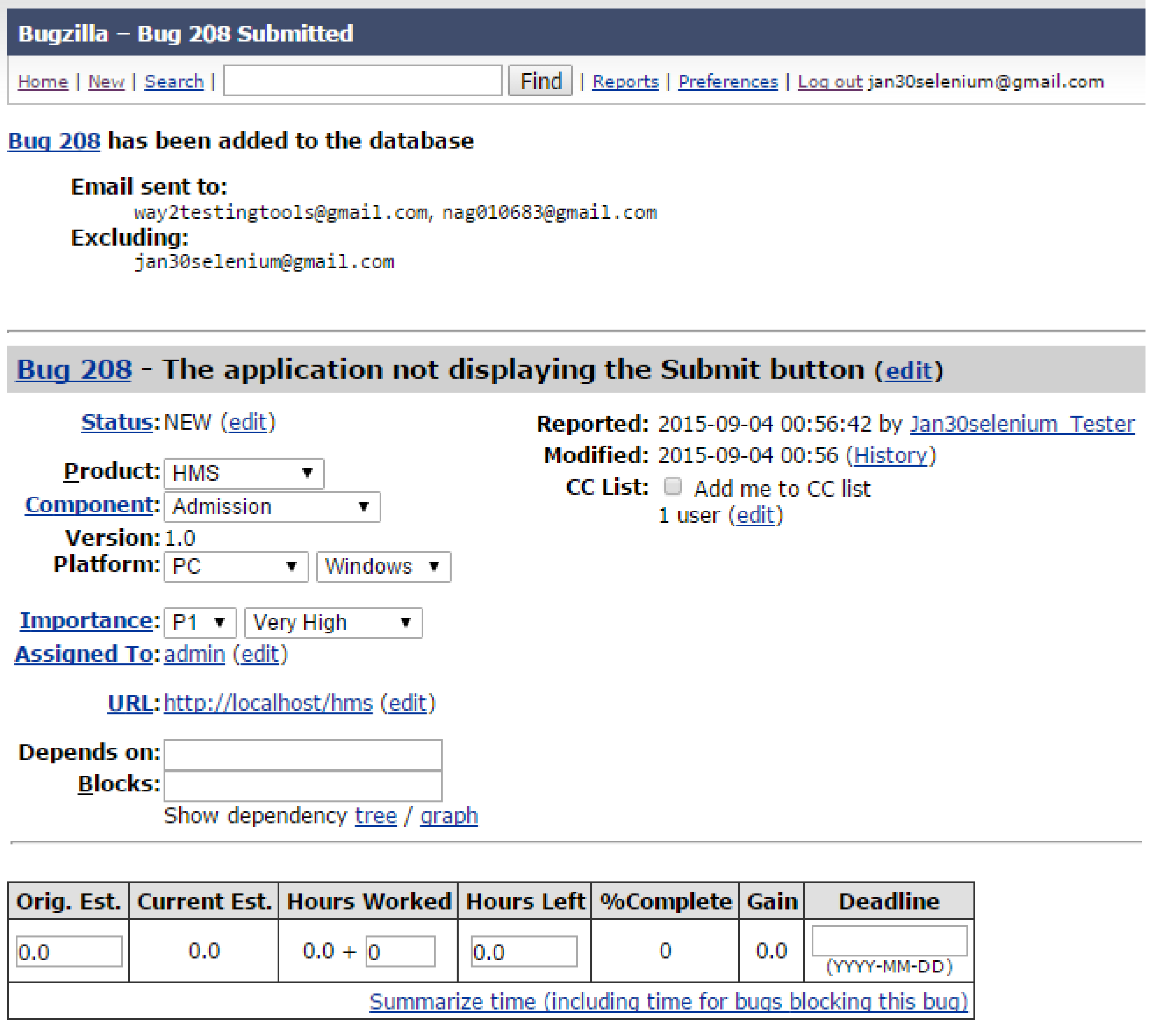
Depends on:

Blocks:

If you have any attachments regards to your reporting bug, you can attach that by clicking “Add an attachment” button and Click on the button Commit at the end of the page to report your bug.

Step 4) Bug is created ID# 208 is assigned to our Bug. You can also add additional information to the assigned bug like URL, keywords, whiteboard, tags, etc. This extra-information is helpful to give more detail about the Bug you have created.

1. Large text box
2. URL
3. Whiteboard
4. Keywords
5. Tags
6. Depends on
7. Blocks
8. Attachments


Bugzilla – Bug 208 Submitted
Home | New | Search | Find | Reports | Preferences | Log out jan30selenium@gmail.com

Bug 208 has been added to the database

Email sent to:
way2testingtools@gmail.com, nag010683@gmail.com
Excluding:
jan30selenium@gmail.com

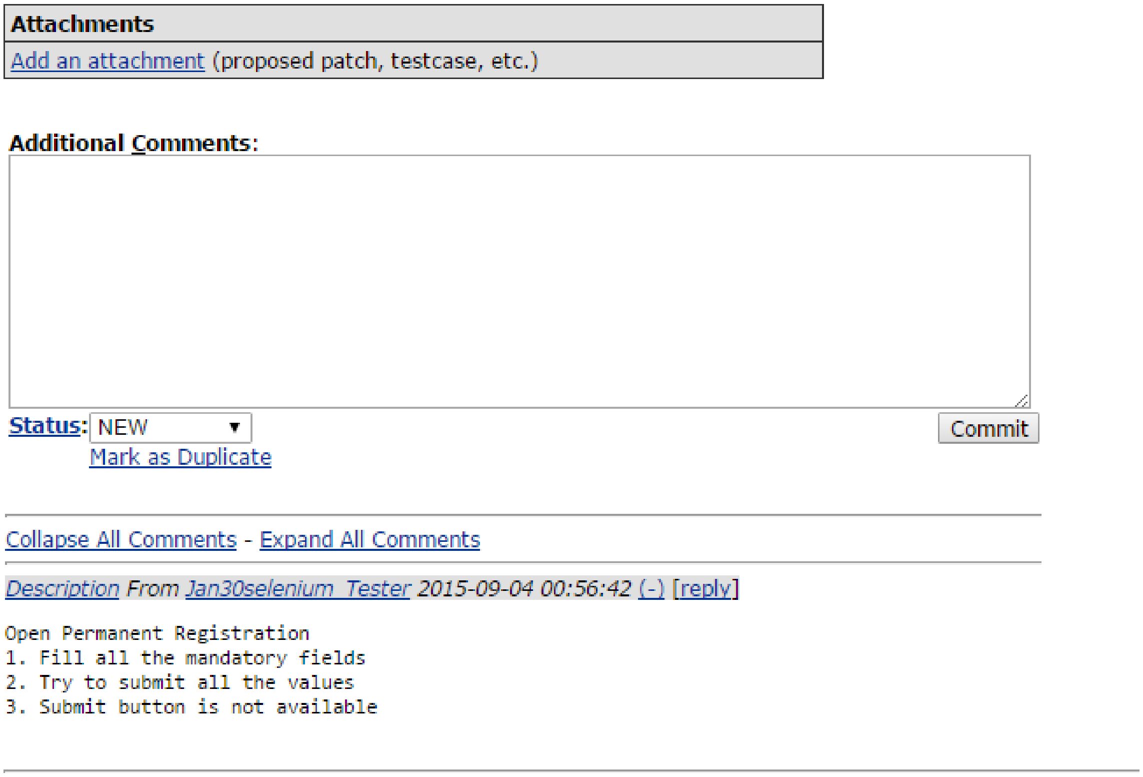
Bug 208 - The application not displaying the Submit button (edit)

Status: NEW (edit)
Product: HMS
Component: Admission
Version: 1.0
Platform: PC Windows
Importance: P1 Very High
Assigned To: admin (edit)
URL: http://localhost/hms (edit)
Depends on:
Blocks:
Show dependency tree / graph

Reported: 2015-09-04 00:56:42 by Jan30selenium Tester
Modified: 2015-09-04 00:56 (History)
CC List: □ Add me to CC list
1 user (edit)

Orig. Est.	Current Est.	Hours Worked	Hours Left	%Complete	Gain	Deadline
0.0	0.0	0.0 + 0	0.0	0	0.0	(YYYY-MM-DD)

Summarize time (including time for bugs blocking this bug)

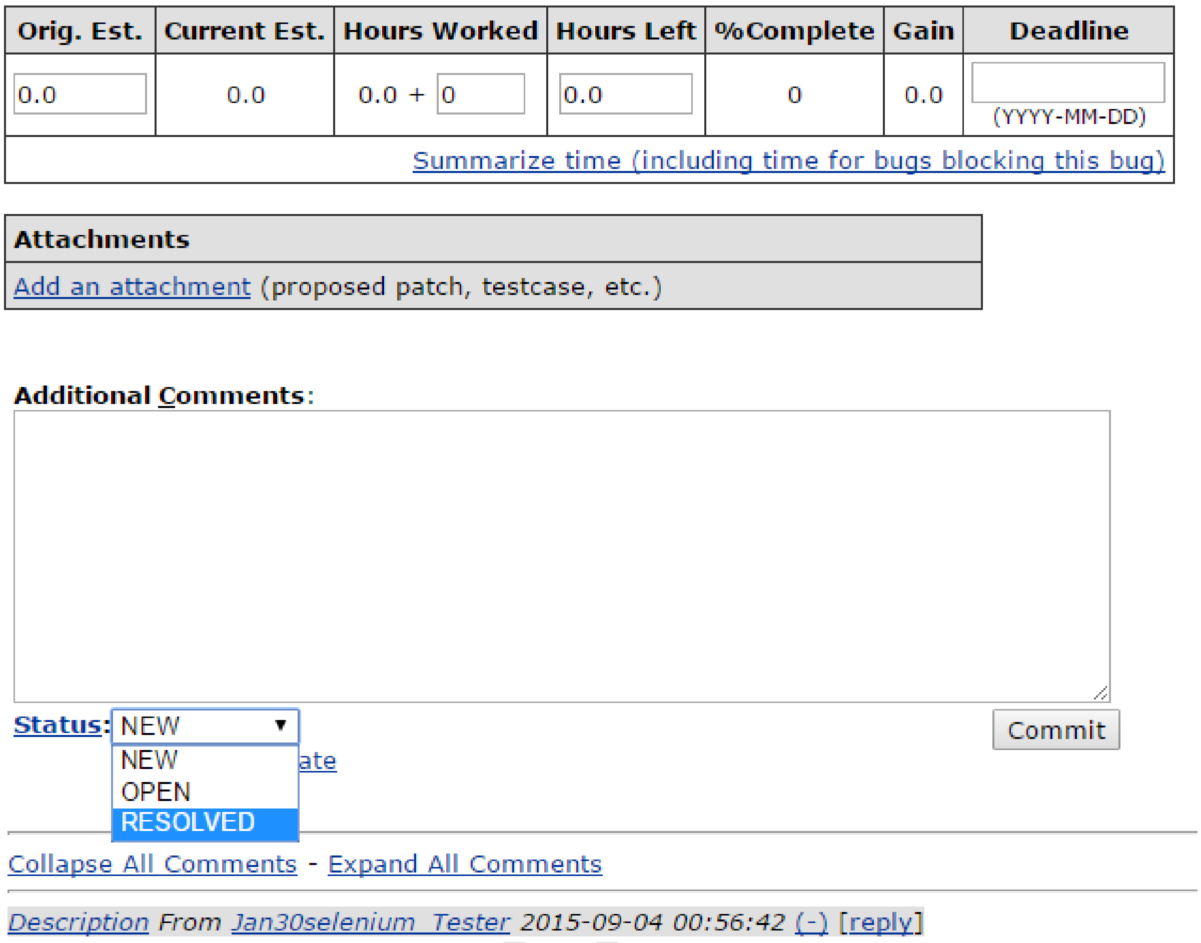


Attachments
Add an attachment (proposed patch, testcase, etc.)
Additional Comments:
Status: NEW
Mark as Duplicate
Commit
Collapse All Comments - Expand All Comments

Description From Jan30selenium_Tester 2015-09-04 00:56:42 (-) [reply]

```
Open Permanent Registration
1. Fill all the mandatory fields
2. Try to submit all the values
3. Submit button is not available
```

Step 5) In the same window if you scroll down further. You can select deadline date and also status of the bug.**Deadline in Bugzilla usually gives the time-limit to resolve the bug in given time frame.**



Orig. Est.	Current Est.	Hours Worked	Hours Left	%Complete	Gain	Deadline
0.0	0.0	0.0 + 0	0.0	0	0.0	(YYYY-MM-DD)
Summarize time (including time for bugs blocking this bug)						

Attachments
Add an attachment (proposed patch, testcase, etc.)

Additional Comments:

Status: NEW

Collapse All Comments - Expand All Comments

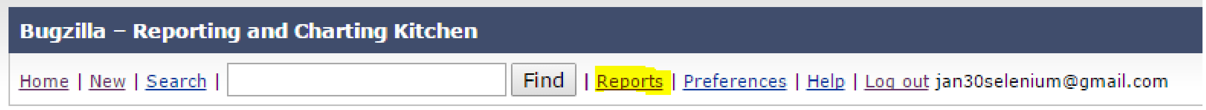
Description From Jan30selenium Tester 2015-09-04 00:56:42 (-) [reply]

Create Graphical Reports

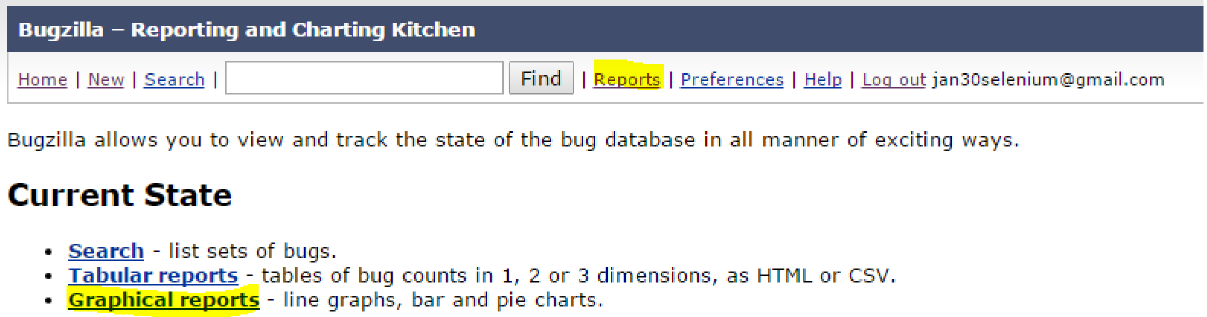
Graphical reports are one way to view the current state of the bug database. You can run reports either through an HTML table or graphical line/pie/bar-chart-based one. The idea behind graphical report in Bugzilla is to define a set of bugs using the standard search interface and then choosing some aspect of that set to plot on the horizontal and vertical axes. You can also get a 3-dimensional report by choosing the option of "Multiple Pages".

Reports are helpful in many ways, for instance if you want to know which component has the largest number of bad bugs reported against it. In order to represent that in the graph, you can select severity on X-axis and component on Y-axis, and then click on generate report. It will generate a report with crucial information.

1. Click on reports at the top of the window as follow


Bugzilla – Reporting and Charting Kitchen
Home | New | Search |
Find
| Reports | Preferences | Help | Log out jan30selenium@gmail.com

2. Click on Graphical reports

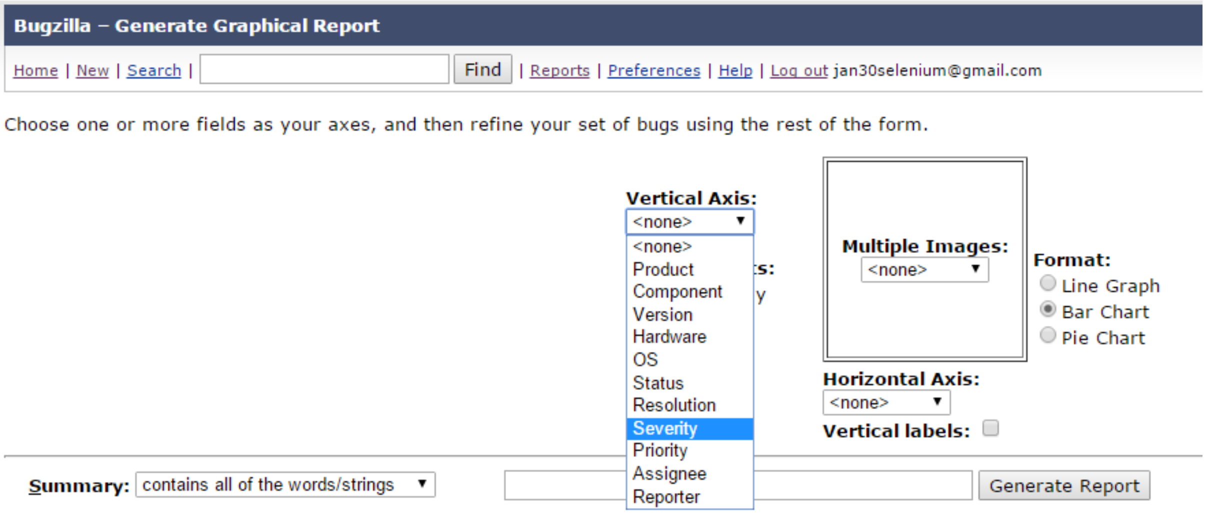

Bugzilla – Reporting and Charting Kitchen
Home | New | Search |
Find
| Reports | Preferences | Help | Log out jan30selenium@gmail.com

Bugzilla allows you to view and track the state of the bug database in all manner of exciting ways.

Current State

- Search - list sets of bugs.
- Tabular reports - tables of bug counts in 1, 2 or 3 dimensions, as HTML or CSV.
- Graphical reports - line graphs, bar and pie charts.

Graphical reports page will be displays as below,



Bugzilla – Generate Graphical Report

Home | New | Search | [ ] Find | Reports | Preferences | Help | Log out jan30selenium@gmail.com

Choose one or more fields as your axes, and then refine your set of bugs using the rest of the form.

Vertical Axis: <none>

Multiple Images: <none>

Format: ○ Line Graph ● Bar Chart ○ Pie Chart

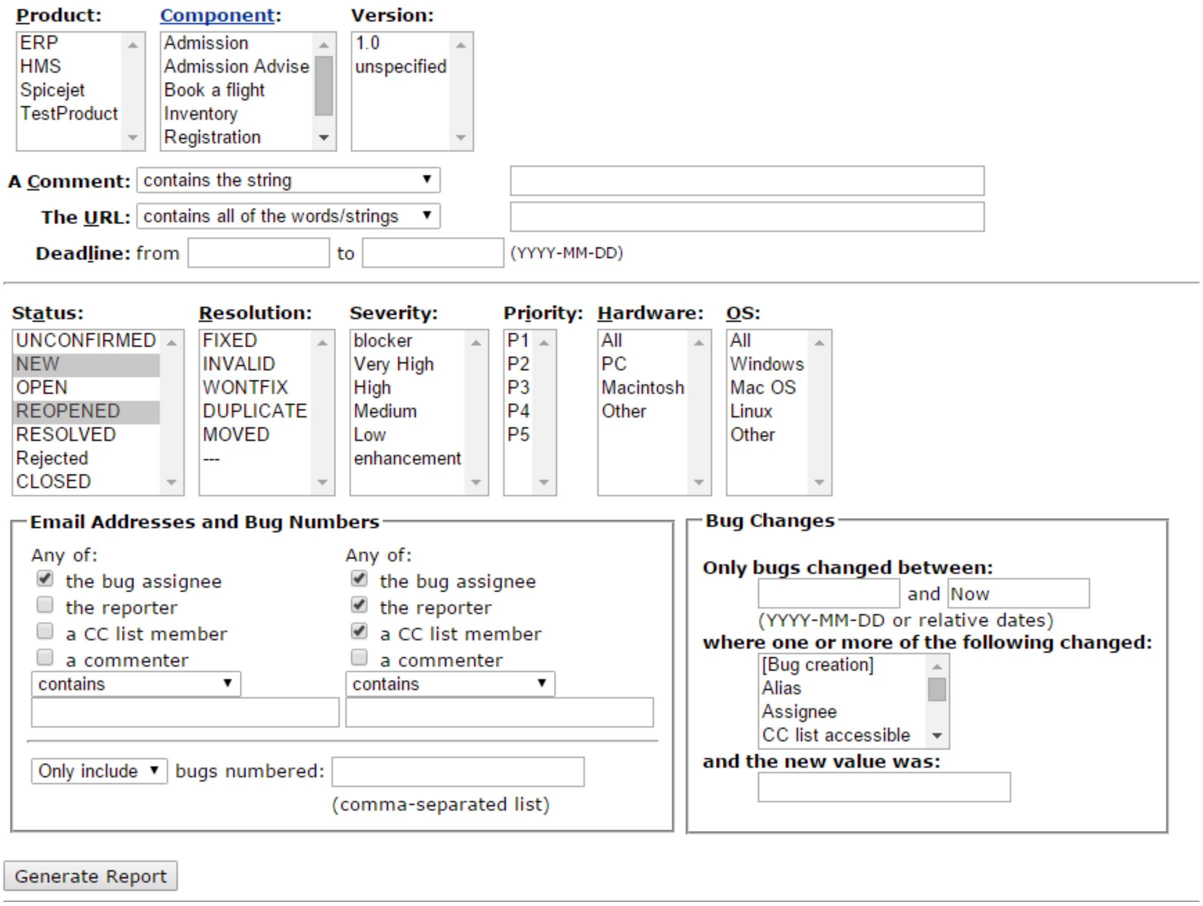
Horizontal Axis: <none>

Vertical labels: □

Summary: contains all of the words/strings [ ] Generate Report

Select Vertical Axis as Severity and Horizontal Axis as Priority or whatever you want you can select it and you will get corresponding Graphical report.

➔ Generate report with Advanced features by entering more details regards the bug as below



Product: ERP, HMS, Spicejet, TestProduct

Component: Admission, Admission Advise, Book a flight, Inventory, Registration

Version: 1.0, unspecified

A Comment: contains the string

The URL: contains all of the words/strings

Deadline: from [ ] to [ ] (YYYY-MM-DD)

Status: UNCONFIRMED, NEW, OPEN, REOPENED, RESOLVED, Rejected, CLOSED

Resolution: FIXED, INVALID, WONTFIX, DUPLICATE, MOVED, ---

Severity: blocker, Very High, High, Medium, Low, enhancement

Priority: P1, P2, P3, P4, P5

Hardware: All, PC, Macintosh, Other

OS: All, Windows, Mac OS, Linux, Other

Email Addresses and Bug Numbers

Any of:
☑ the bug assignee
□ the reporter
□ a CC list member
□ a commenter
contains

Any of:
☑ the bug assignee
☑ the reporter
☑ a CC list member
□ a commenter
contains

Only include bugs numbered: [ ] (comma-separated list)

Bug Changes

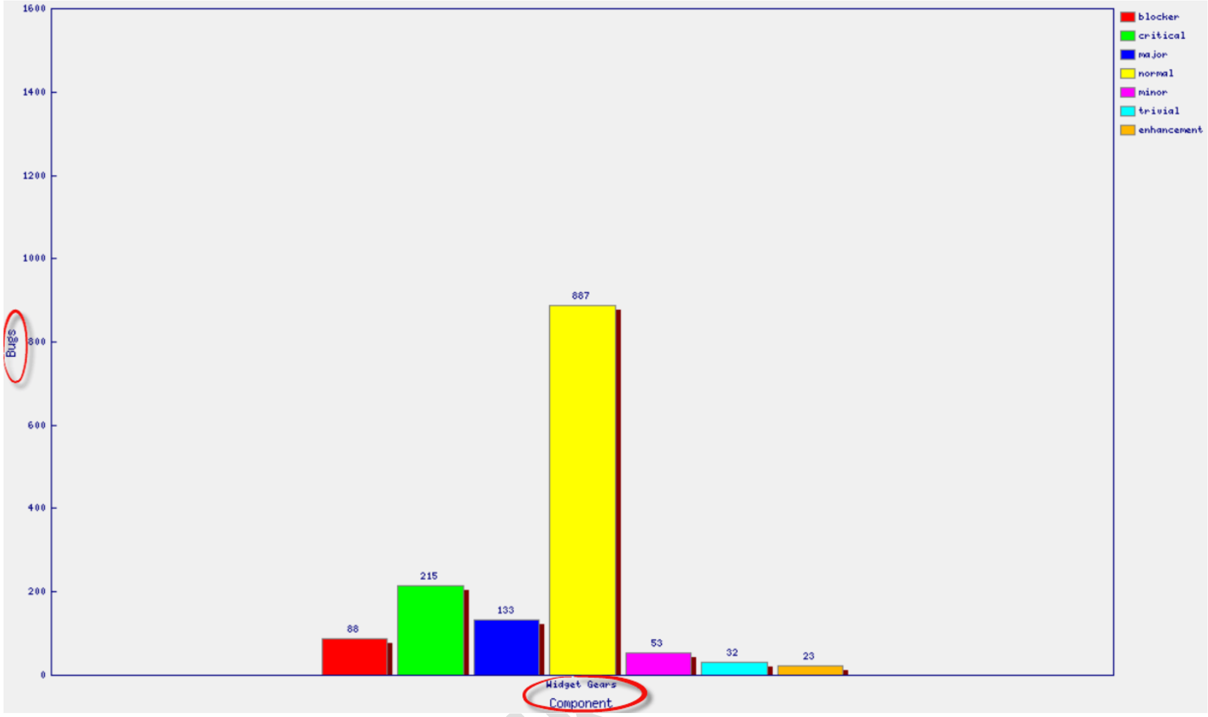
Only bugs changed between: [ ] and Now (YYYY-MM-DD or relative dates)

where one or more of the following changed: [Bug creation], Alias, Assignee, CC list accessible

and the new value was:

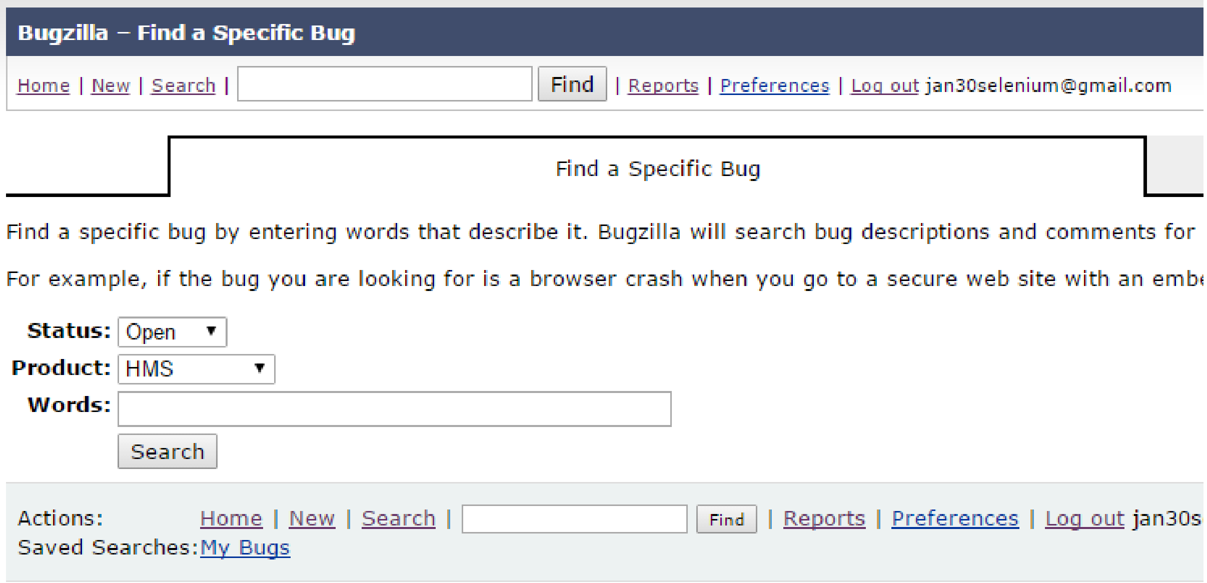
Generate Report

The graph below shows the Bar chart representation for the Bugs severity in component **"Widget Gears".** In the graph below, the most severe bug or blockers in components are 88 while bugs with normal severity is at top with 667 number.



Browse Function

Step 1) To locate your bug we use browse function, click on **Search** button from the main menu.



Bugzilla – Find a Specific Bug

Home | New | Search | Find | Reports | Preferences | Log out jan30selenium@gmail.com

Find a Specific Bug

Find a specific bug by entering words that describe it. Bugzilla will search bug descriptions and comments for

For example, if the bug you are looking for is a browser crash when you go to a secure web site with an embe

Status: Open ▾

Product: HMS ▾

Words:

Search
Actions: Home | New | Search | Find | Reports | Preferences | Log out jan30s
Saved Searches: My Bugs

The reporting will be continued until the build is stable. Once its stable it will be delivered to the client then Refer delivery and maintenance phase

Once the build is successfully delivered to the client, the test lead will prepare test summary report and it will be updated in test plan and the test plan send to the client at the time of delivering the build.

TEST SUMMARY REPORT / BUILD POST MARTUM REPORT

- No of Builds released by Dev Team - 50
- No of Builds accepted by testing team - 25
- No of Builds rejected by testing team - 25
- No of test cases prepared by testing team - 1000
  - **P1 - 500, P2- 350, P3-100, P4-50**
- No of bugs identified - 400
  - Blocker - 100
  - Very High - 150
  - High - 100
  - Medium - 40
  - Low - 10
- No of bugs identified by client - 100
  - Blocker - 10
  - Very High - 50
  - High - 10
  - Medium - 10
  - Low - 20
- Success Stories
- Challenges

Q : What is the entry(start) criteria of testing and exit(end) criteria of testing

A: Test plan and SRS document are the entry criteria of testing.

There is no end for testing. It will be continued as long as the build is on live. But the testing activities will change after the build delivered to client. During maintenance we are not going to perform all the functional testing types. We will perform the regression testing.

Terminology	
Peer Review	Peer means the team mate with equal designation. All the peers will participate in a meeting and discuss on the project to get clarity on all the modules. The objective of peer review is to get functional knowledge on all the modules by each test engg..
Peer Review Report	While peer review the senior peer will prepare the document on peer review is known as Peer review report
Walk through	If the same peer review is conducted infront of the lead or project manager then its known as Walk through. The lead or PM will observe if the team members are understanding the project properly or not
Walk through Report	While Walk through the lead will prepare the document on Walk through is known as Walk through report
Test Suite	Combination of multiple test cases is known as Test suite
Test Bed	Combination of Test suite and test environment is known as Test Bed
DSR	Daily status Report. On daily basis we need to send our work status to the lead in a template
MOM	Minutes of meeting. When ever we participate in a meeting, the meeting discussion will be taken into a rough note. Later it will be updated in a mail and the mail will be send to all the team members. The purpose of MOM is to maintain the transparency on the meeting among the team members
Inspection	The process of checking whether the company following the guidelines or not. Inspection will be done with out intimation
Audit	It is also the process of checking whether the company following the guidelines or not. Audit will be performed with prior intimation
Stable	Stable means no further updations are needed. Stable build means majority of the test cases are pass and no blocker bugs identified in the build
AUT	Application under test
Patch Build	When ever the build is rejected the developer will analyze the failure. If he is releasing the same build to testing team again with out adding any new functionalities is know as PatchBuild. If the developer is releasing the build with some new functionalities is known as new build
Publish	Make it available for the targetted users. Once the Testcases or SRS document is base lined it will be check in in the central Repository for targetted users. It is know as publish
Bug/Defect/Error	Design related faults are know as defect. Ex: GUI defects Functional related faults are bugs (Programmer mistake). Ex: All functional bugs Error: Exceptions in the script is known as error
Use cases	Use case is a list of steps, typically defining interactions between a role (known as an "actor") and a system, to achieve a goal. The actor can be a Test engineer or End user The requirements will be converted to list of steps by the BA
NCR	Non-conformance Reports or Non-conformance change Request. The Requirement which is under discussion

CAPA	Corrective actions are implemented in response to customer complaints Preventive actions are implemented in response to the identification of potential sources
SCM	In software engineering, software configuration management (SCM or SWCM) is the task of tracking and controlling changes in the software. SCM practices include revision control and the establishment of baselines.
SDN	Software Delivery Note
Slippage	Subsiding. Moving away from the task The no of days you have slipped from the task
Defective product	The product with defects
Defect age	Defect Age (in Time) is the difference in time between the date a defect is reported and the current date (if the defect is still open) or the date the defect was fixed (if the defect is already fixed).
Latent Defect	Hidden Defect
PPM	Product Portfolio Management,is is the centralized management of the processes, methods, and technologies used by project managers
PPR	Program Performance Review Reports
MRM	Marketing Resource Management