Manual Testing

1. What is a software? Types of software?  
   A software is a collection of computer programs that helps us to perform a task.
2. System software:  
   E.g., Device drivers, operating systems, servers etc.
3. Programming software:  
   E.g., Compilers, Debuggers, interpreters etc.
4. Application software:  
   E.g., Web applications, mobile apps, desktop applications etc.
5. What is the software testing?  
   X Bank (company) --- > IT Company ---- > Develop ----- > Test ----- > Deliver ----- > X Bank  
   It is the process of executing an application under controlled conditions with the intent of finding defects.
6. What is software quality?
7. Bug – free.
8. Delivered on - time.
9. With - in budget.
10. Meets requirements / or expectations.
11. Maintainable.

These are the parameters for the product to be as a quality product.

1. Project vs Product:

If the software application is developed for a specific customer based in the requirement, then it is called project.  
E.g., Bank application.

If the software application is developed for multiple customers based on the market requirement, then it is called product.  
E.g., WhatsApp.

Types of companies:

1. Service based companies.

Eg: TCS, Accenture.

1. Product based companies.

Eg: Google, Microsoft.

1. Why do we need testing?

To deliver the quality product to the customer.

1. Error, bug / defect, failure:
2. Error: Incorrect input from the user.
3. Bug / defect: unexpected outcome.
4. failure: In real time, if there are any bugs / defects that is the failure.
5. Why the software has bugs?
6. Miscommunication / no communication between testers and developers.
7. Software complexity.
8. Programming errors.
9. Changing requirements.
10. Lack of skilled testers.
11. SDLC: Software Development Life Cycle  
    SDLC is a process used by software industry to design, develop and test software’s.  
    Three p’s:
12. People.
13. Process.
14. Product.

a. Requirements analysis.  
b. Design.  
c. Development.  
d. Testing.  
e. Maintenance.

a. Requirement analysis:  
 Collect all the requirements for the product from the customer.

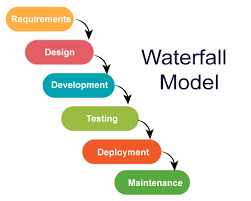
b. Design:  
 Designers will design the product type.

c. Development:  
 Developers write a code in multiple languages based on the customer requirement.

d. Testing:  
 Testers will test the product and deliver to the client.

e. Maintenance:  
 Deploying product in customer environment and using it.

9. Waterfall Model  
 It is the oldest and traditional model.



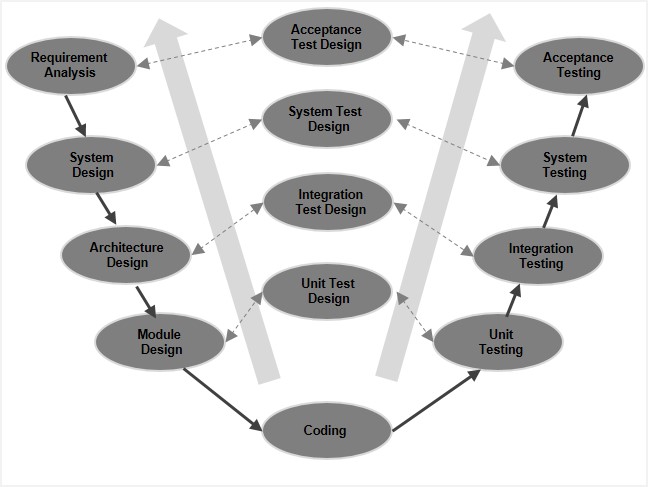
1. Advantages:  
   1. Quality of the product will be good.  
   2. Since requirement changes are not allowed, chances of finding bugs will be less.

3. Initial investment is less since the testers are hired at the later stages.  
4. Preferred for small projects when requirements are freezed.

1. Disadvantages:  
   1. Requirement changes are not allowed.

2. If there is defect in requirement phase, that will be continued in next phase.  
3. Testers will start only after coding.

10. V - Model  
 In every part of the development life cycle, testing is done.



BRS – Business Requirement Specification  
CRS – Customer Requirement Specification  
URS – User Requirement Specification

* These documents contain the requirements of a product and these documents are prepared by business team.
* These documents are tested based on user acceptance testing (UAT).
* These documents are not understandable by technical group.
* Based on these business documents another document called SRS (Software Requirement Specification) is prepared by project manager.
* Based on SRS document, HLD and LLDs are prepared by architects or designers.
* Developers write a code based on the documents.
* All the documents re tested based on static technique methods.
* Static testing techniques:  
   Used for testing documents.  
  1. Review.

2. Walkthrough.  
3. Inspection.

* Testing conducting on coding is known as white box testing done by the developers.
* Dynamic Testing:

Testing the actual software not documentation part.

* Verification:  
  1. Checks whether we are building a right product.  
  2. Focus on documentation.
* Validation:  
  1. Checks whether we are building the right product.  
  2. Takes place after verification is completed.  
  3. Focus on software testing.

Advantages:  
 Testing is involved in each and every phase.

Disadvantages:  
1. Documentation is more.  
2. Initial investment is more.

11. Static testing techniques:  
 a. Review:  
 1. Conducts on documents to enclosure correctness and completeness.  
 2. Requirement reviews.  
 Design reviews.  
 Code reviews.  
 Test plan reviews.  
 Test case reviews.  
 b. Walkthrough:  
 It is an informal review.

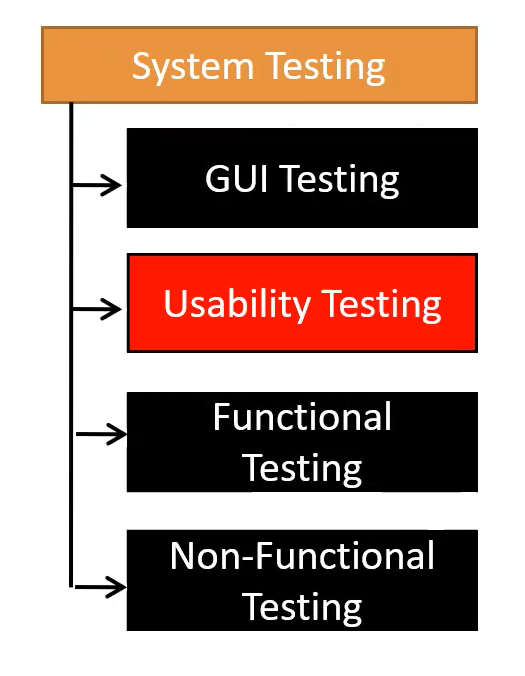
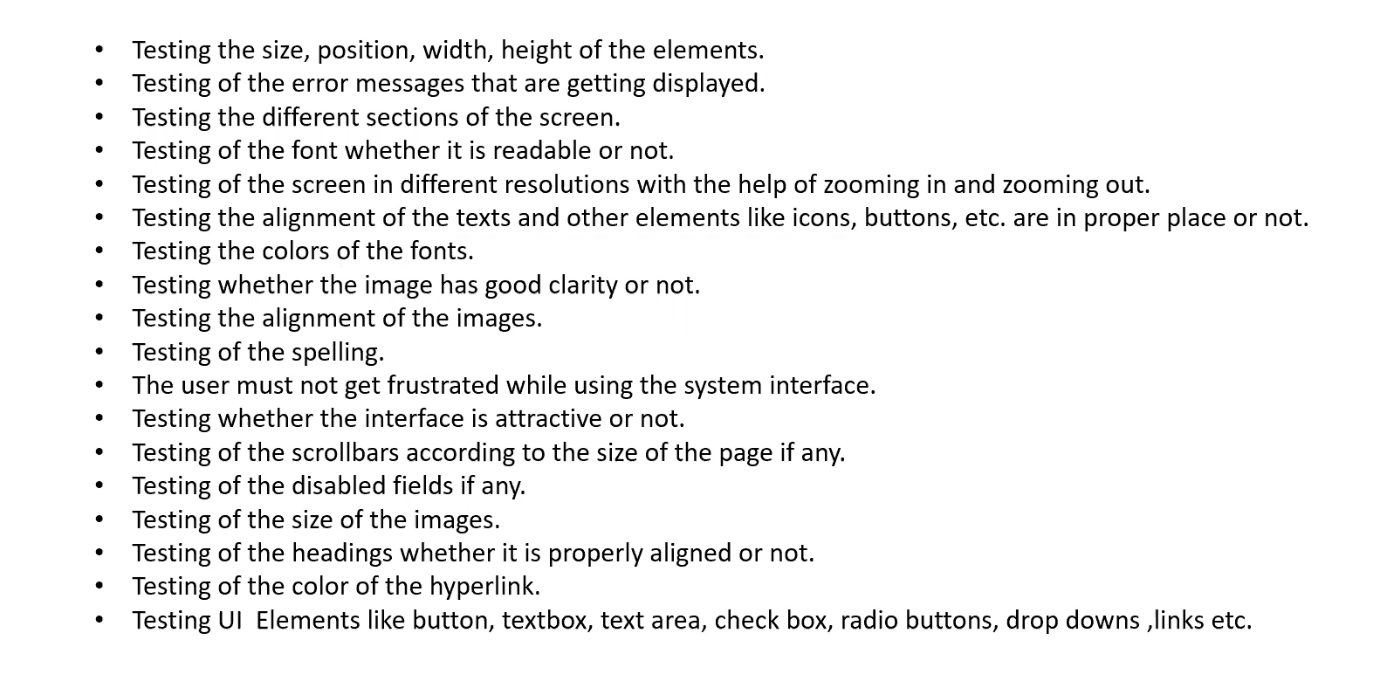
c. Inspection:  
 1. It is a formal review.  
 2. Inspection will have a proper schedule to conduct a meeting.

12. QA, QC  
 1. QA is a process related.  
 QC is actual testing of the software.  
 2. QA is focuses on building in quality.  
 QC focuses on testing for quality.  
 3. QA is preventing defects.  
 QC is detecting defects.  
 4. QA is process oriented.  
 QC is product oriented.

QE – Quality Engineering

13. Types of software testing:

1. Unit Testing:  
   1. It is a type of white box technique.  
   2. It is conducted by developers.  
   3. A unit is a single component or a module of a software.  
   4. Unit testing conducts on a single program or a single module.  
   5. Unit testing techniques:  
    a. Basic path testing.  
    b. Control structure testing.  
    c. Conditional coverage.  
    d. Loops coverage.  
    e. Mutation coverage.
2. Integration testing:  
   1. Integration testing is also a type of white box testing technique.  
   2.It is conducted by developers.  
   3.Integration testing is performed between two or more modules.  
   4. Integration testing focuses on checking data communication between  
    multiple modules.  
   5. Types: a. Incremental Integration testing.  
    b. Non – Incremental Integration testing.
3. System Testing:  
   1. It is a type of black box testing.  
   2. It is done by testing team.  
   3. Testing all over the functionality with respective client requirement of  
    that application.  
   4. Focuses on:  
    a. GUI testing.  
    b. Functional testing.  
    c. Non – functional testing.  
    d. Usability testing.
4. User acceptance testing:  
   After completion of system testing UAT team conducts acceptance testing in two levels.  
   a. Alpha testing.  
   b. Beta testing.

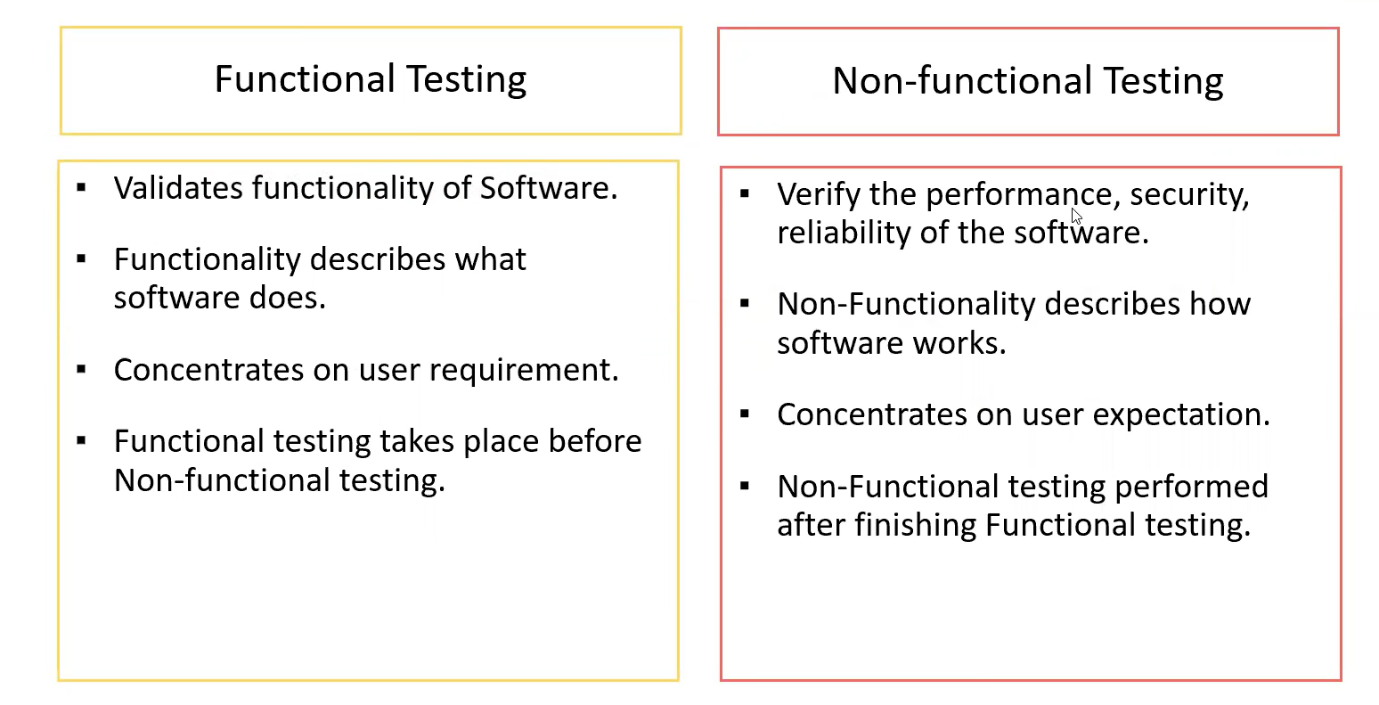
14.System Testing types:  
System testing: Testing software or an application.  
  
  
  
a. GUI Testing:  
Graphical User Interface (GUI) testing is a process of testing the user interface of an application.  
🡪A GUI includes all the elements such as menus, checkbox, buttons, colour, fonts, sizes, icons, content, and images.  
  
GUI Testing checklist:  


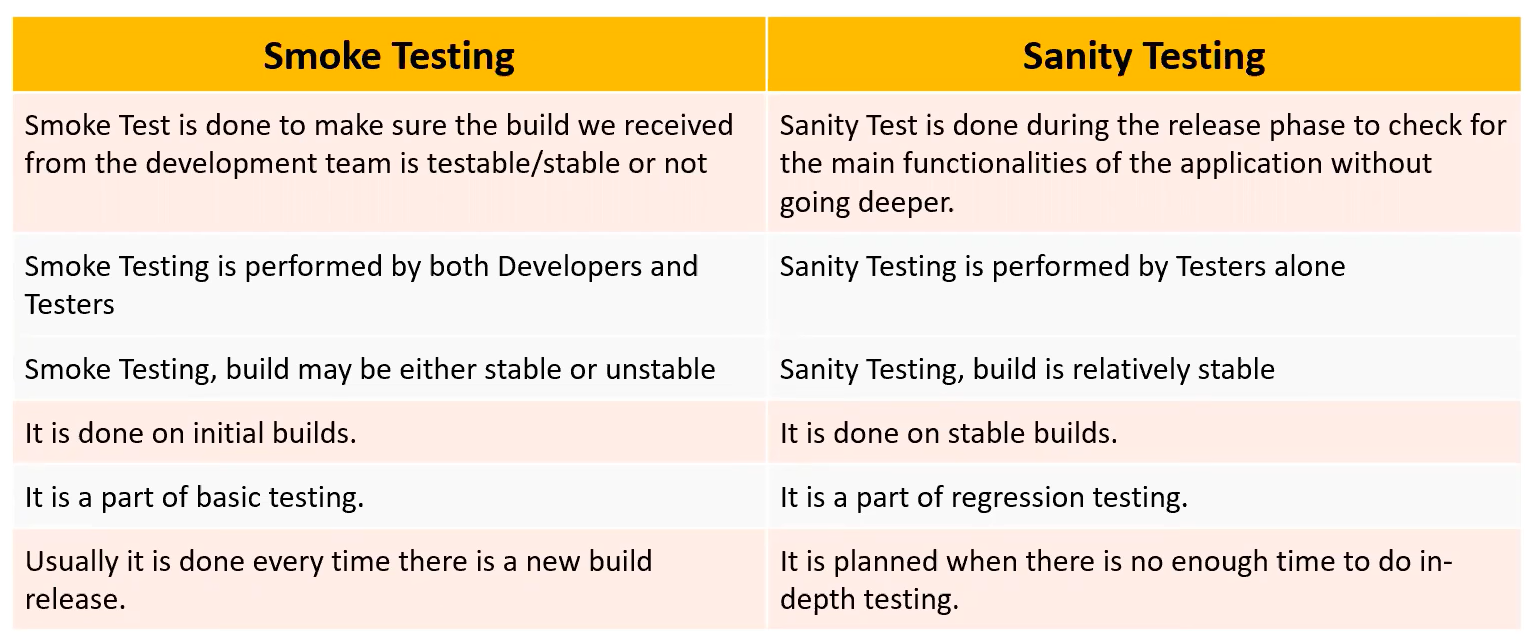
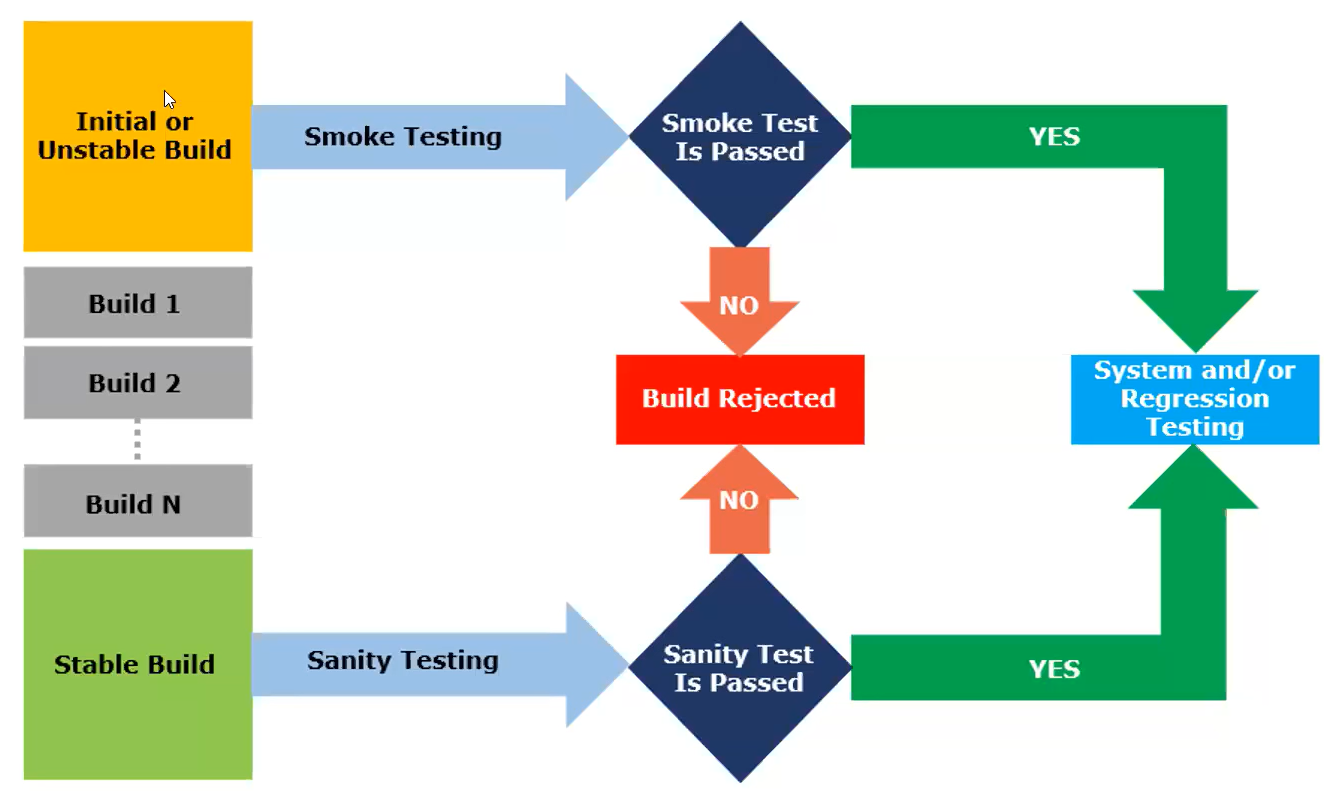
b.Usability Testing:  
Checks how easily the end users are able to understand and operate the application is called Usability testing.  
  
c.Functional testing  
🡪Functionality is nothing but behaviour of an application.  
🡪Functional testing talks about how your feature should work.  
  
🡪Object Properties Testing:  
Checks the properties of objects present on the application.  
e.g. enable, disable, visible, focus, etc.

🡪Database Testing:  
It is also known as Backend Testing.  
DML operations (Data Manipulation Language)  
insert, update, delete, select

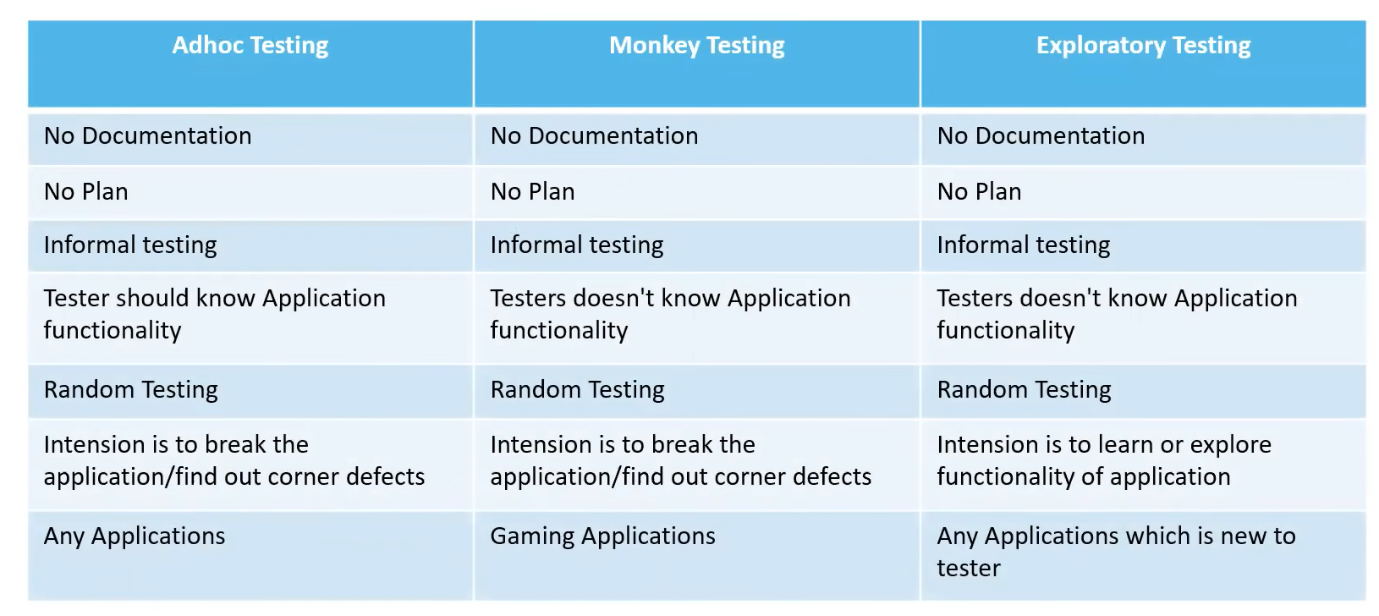
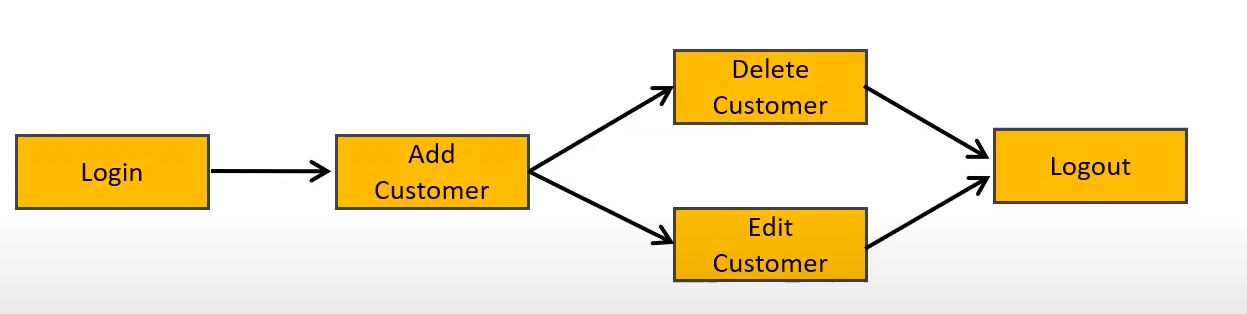
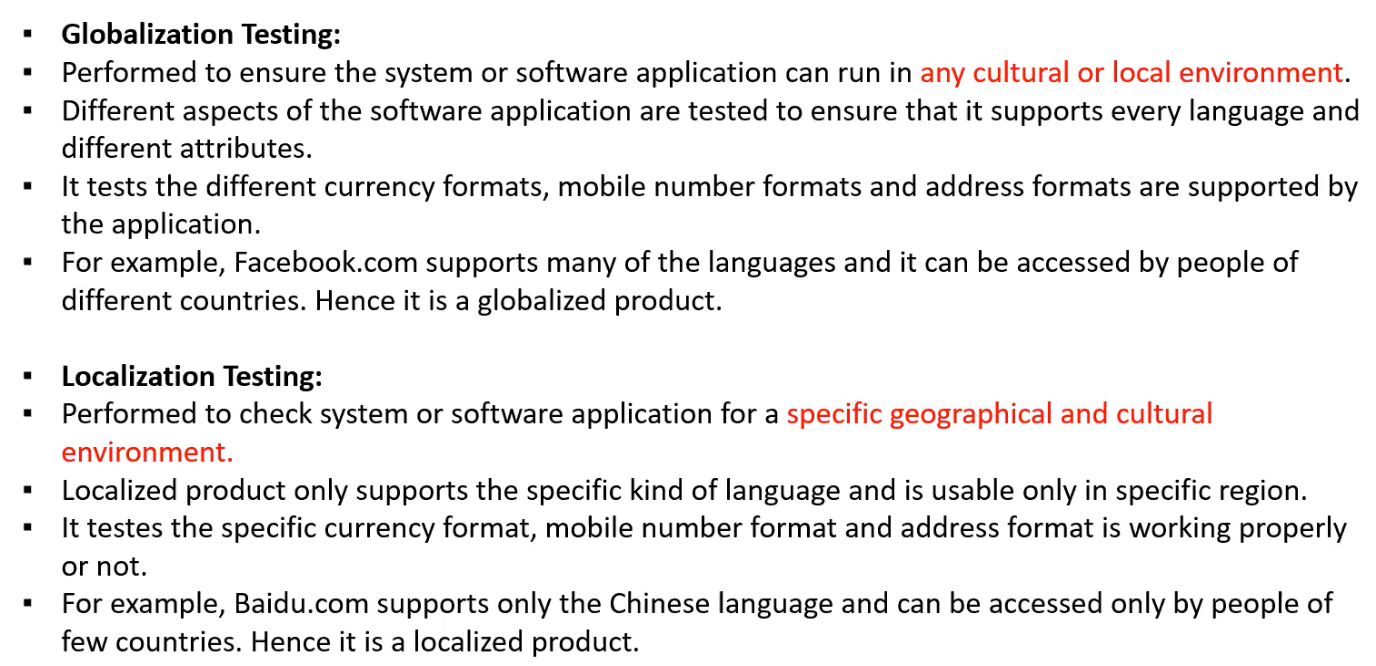
Grey Box Testing: Combination of White box and Black box testing.

In Database testing we can do,   
🡪Table level validations (column type, column length, number of columns in a table)  
🡪Relation between the tables (Normalization)  
🡪Functions  
🡪Procedures  
🡪Triggers  
🡪Indexes  
🡪Views  
etc.  
  
🡪Error Handling:  
Tester verify the error message while performing incorrect actions on the application.  
🡪Error should be readable.  
🡪User understandable / Simple language  
  
🡪Calculations / Manipulation Testing:  
Testing should verify the calculations.  
  
🡪Links Existence / Links Execution:  
Links are navigating to proper page or not --- Links Execution  
Where exactly the links are placed --- Links Execution.  
Internal links  
External links  
Broken links  
  
🡪Cookies & Sessions:  
Temporary files are created by browser while browsing the pages through internet.  
Sessions are time slots created by the server.  
Sessions will be expired after execution.  
  
d.Non – functional testing:  
🡪Once functionality is done, then we do non – functional testing. (Once the functionality is stable)  
🡪Focus on performance, load it can take and security etc.  
  
🡪Performance Testing:  
Speed of an application  
Load Testing: Increasing the load of the application slowly then checks the speed of the application.  
Stress Testing: Suddenly Increase / Decrease the load on the application and checks speed of the application.  
Volume Testing: Check how much data is able to handle by the applications.  
  
🡪Security Testing:  
How secure our application.  
Authentication🡪 Users are valid or not.  
Authorization / Access control 🡪 Permissions of the valid user.  
  
🡪Recovery Testing:  
Check the system change from abnormal to normal.  
  
🡪Compatibility Testing:  
Forward compatibility  
Backward compatibility  
Hardware compatibility (Configuration testing)  
  
🡪Installation Testing:  
Checks screens are clear to understand screens navigation.  
Simple or not  
Un – installation  
  
Sanity / Garbage Testing:  
If any application provides extra features / functionality then we consider them as bug.

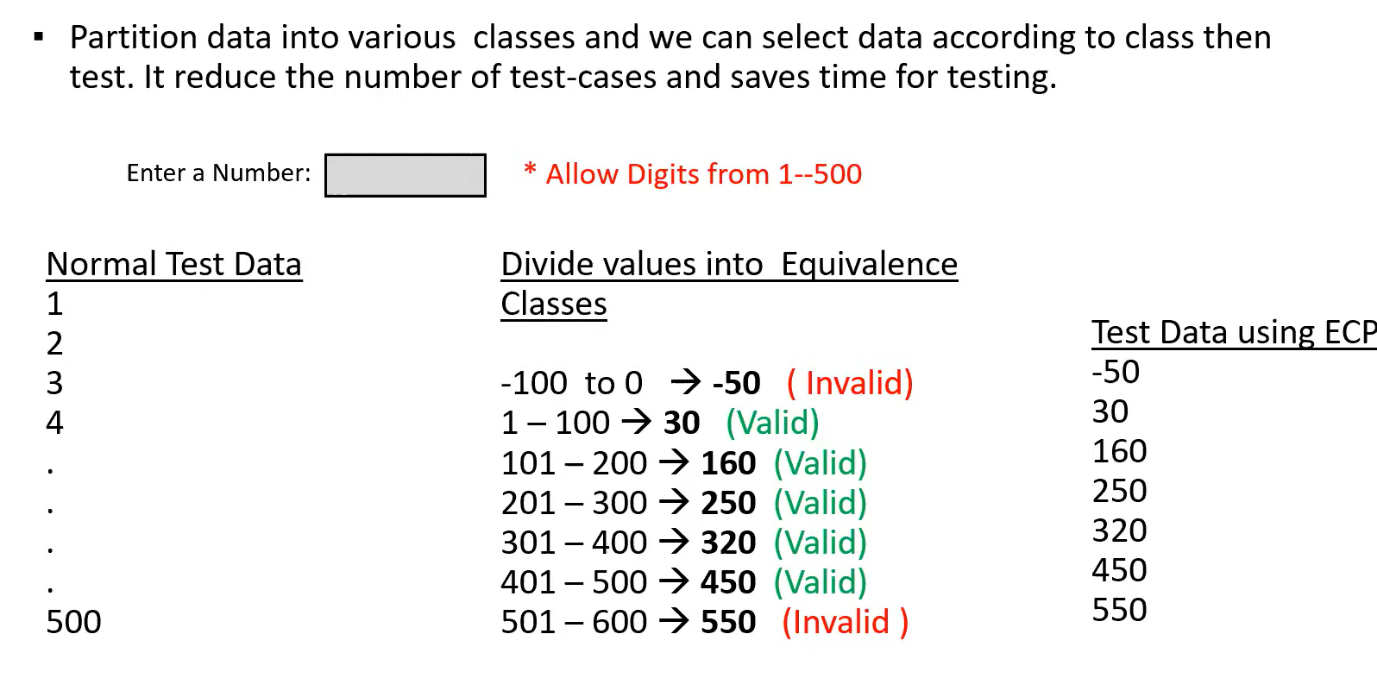
Differences Between Functional and Non – Functional Testing:  
  
  
15. Regression Testing:  
Testing conducts on modified build to make sure there will not be impact on existing functionality because of changes like adding / deleting / modifying features.  
  
🡪Unit Regression testing:  
Testing only the changes / modifications done by the developer.  
  
🡪Regional Regression Testing:  
Testing the modified modules along with the impacted modules.  
  
🡪Full Regression Testing:  
Testing the main feature & remaining part of the application.  
E.g. If Dev has done any changes in many modules, instead of identifying impacted modules, we perform one round of full regression.  
   
16.Re – Testing:  
Whenever the developer fixed a bug, tester will test the bug fix is called Re – Testing.  
Tester close the bug if it is worked otherwise re – open and send to the developer.

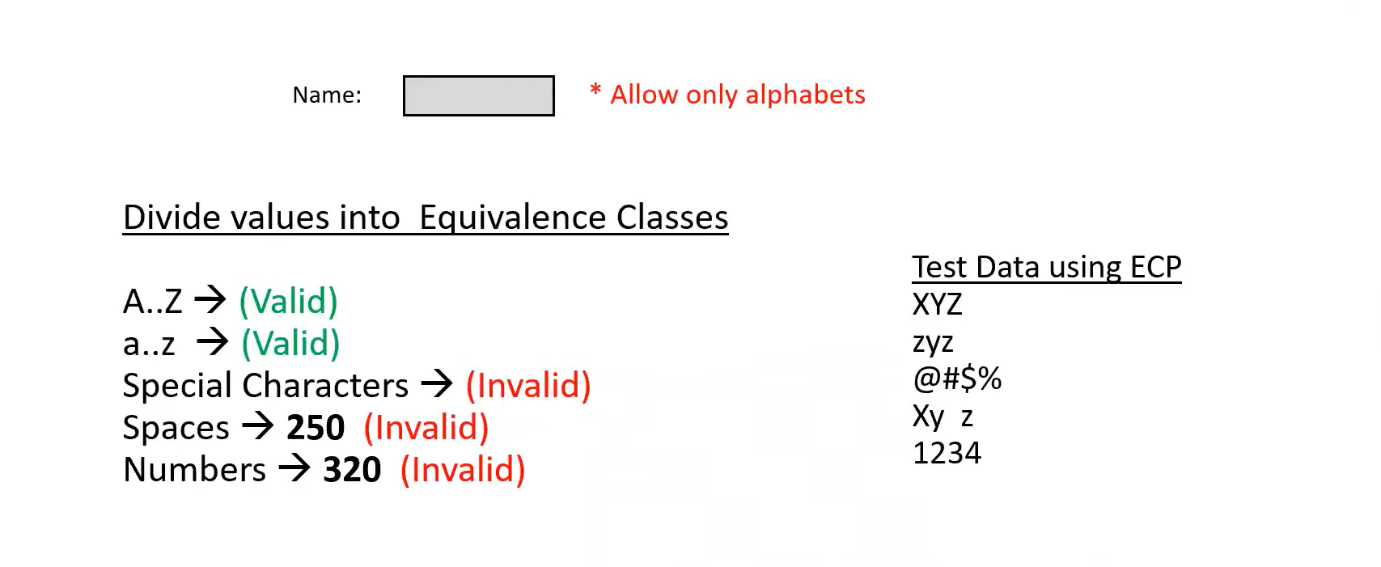
17. Smoke Testing VS Sanity Testing:  
  
  
  
  
  
18. Exploratory Testing:  
We have to explore the application, understand completely and test it.  
Under the application, identify all possible scenarios, document it then use it for testing.  
We do exploratory testing when the application ready but there is no requirement.

19. Ad hoc Testing:  
Ad hoc Testing is an informal testing type with an aim to break the system.  

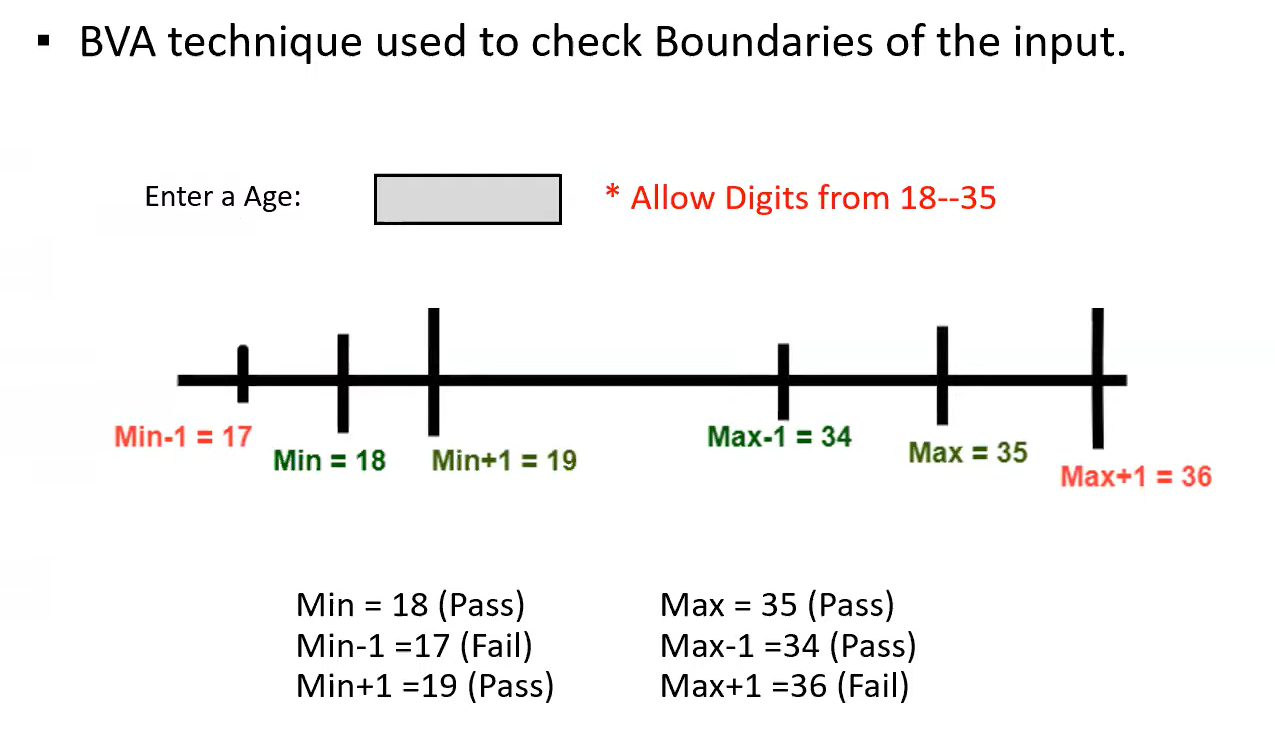

20. Monkey / Gorilla Testing:  
Testing the application randomly without any test cases or any business requirement document.  
Suitable for gaming applications.  
  
  
  
21. Positive Testing:  
Testing the application with valid inputs is called as positive testing.  
  
22. Negative Testing:  
Testing the application with invalid inputs is called as negative testing.  
  
23. END – TO – END Testing:  
1. Login  
2. Add new customer  
3. Edit customer  
4. Delete customer  
5. Logout  
  
  
24. Globalization and Localization testing  


25.Testing Design Techniques:  
Used to: a. Reduce the data  
b. More coverage  
Types:  
1. Equivalence class partitioning (ECP):  
1. Value check  
2. Classify / divide / partition data into multiple values

Examples:  


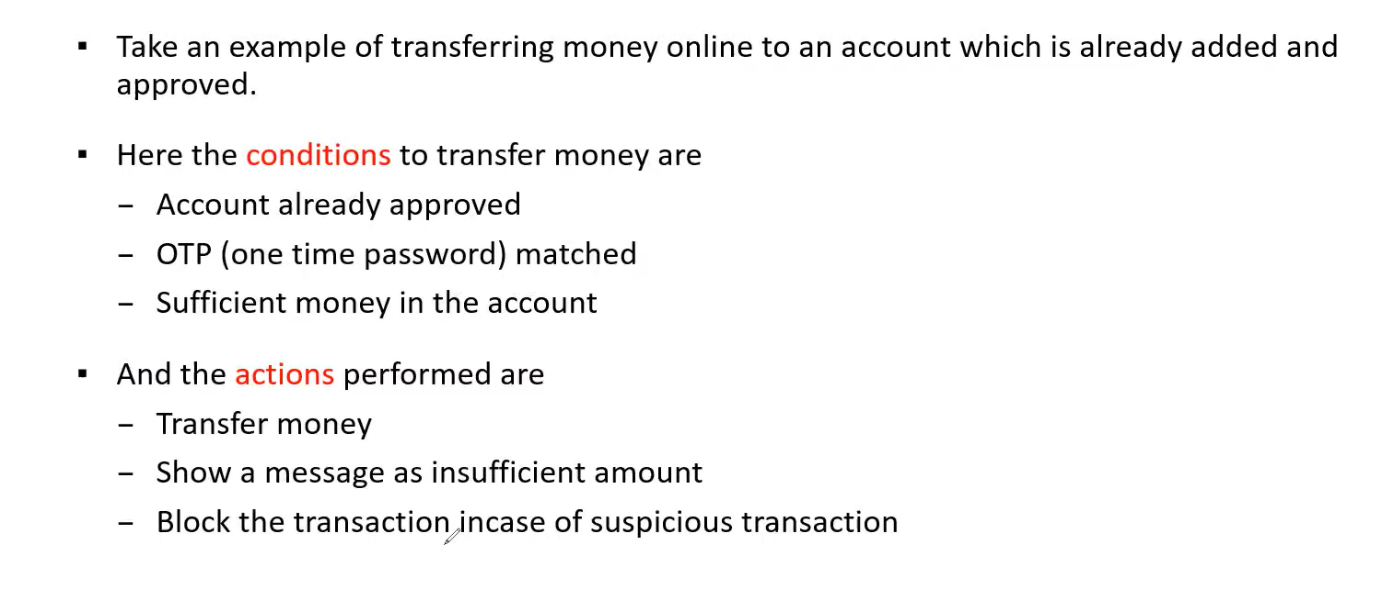


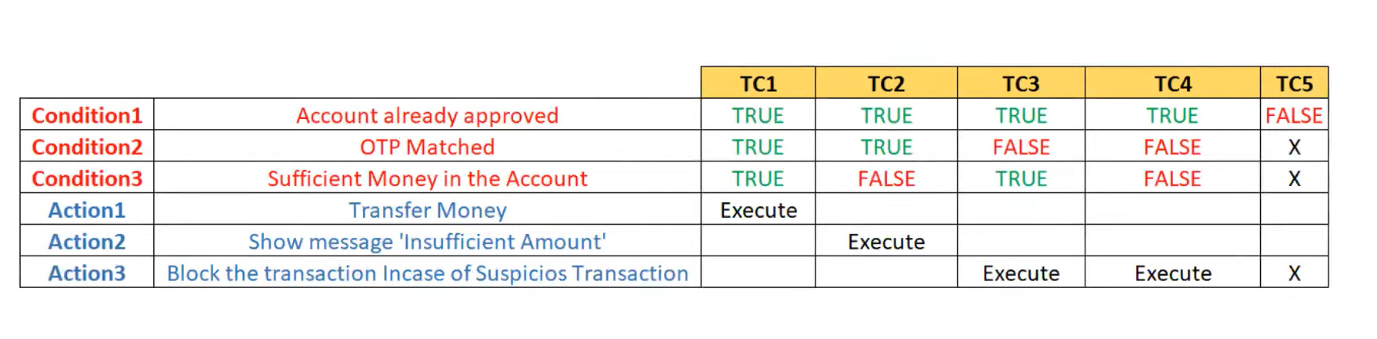
2.Boundary Value Analysis (BVA):  
Boundary of the values  
Min  
Min – 1  
Min + 1  
  
Max  
Max – 1  
Max + 1



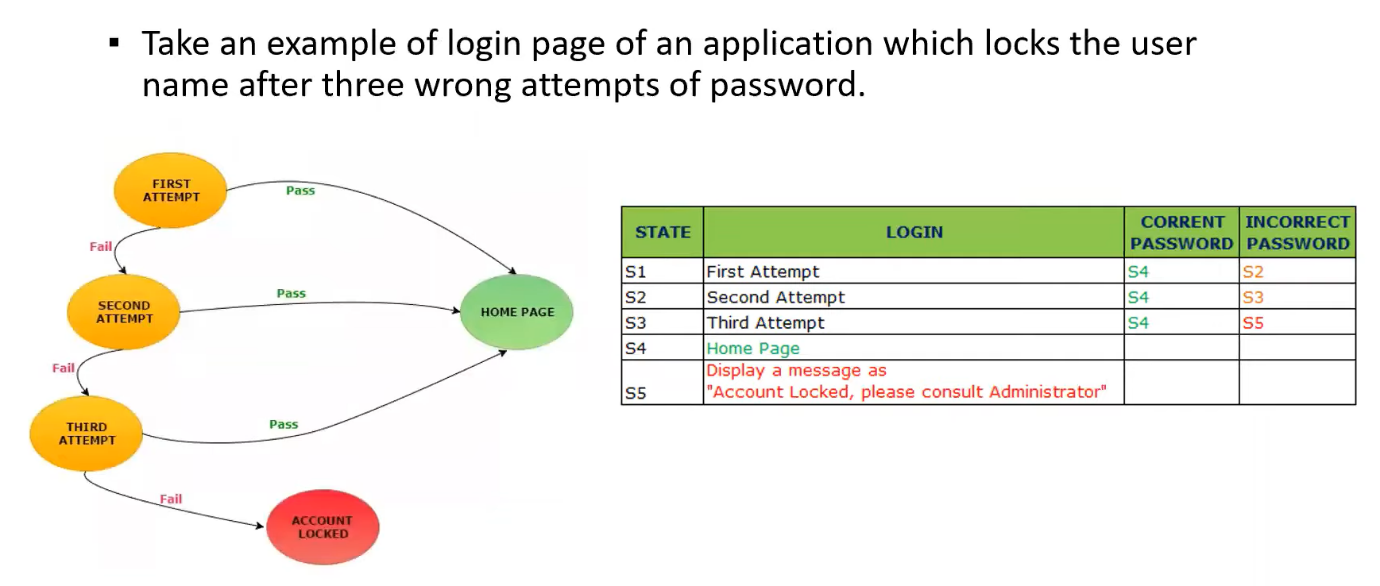
Input Domain Testing:  
The value will be verified in the text box / input fields.  
We use ECP and BVA techniques.

3.Decision table-based testing:  
a. Decision table is also called cause – effect table.  
b. This technique will be used if we have more conditions and corresponding actions.  
c. In Decision table technique, we deal with combination of inputs.  
d. To identify the test cases with decision table, we consider conditions and actions.



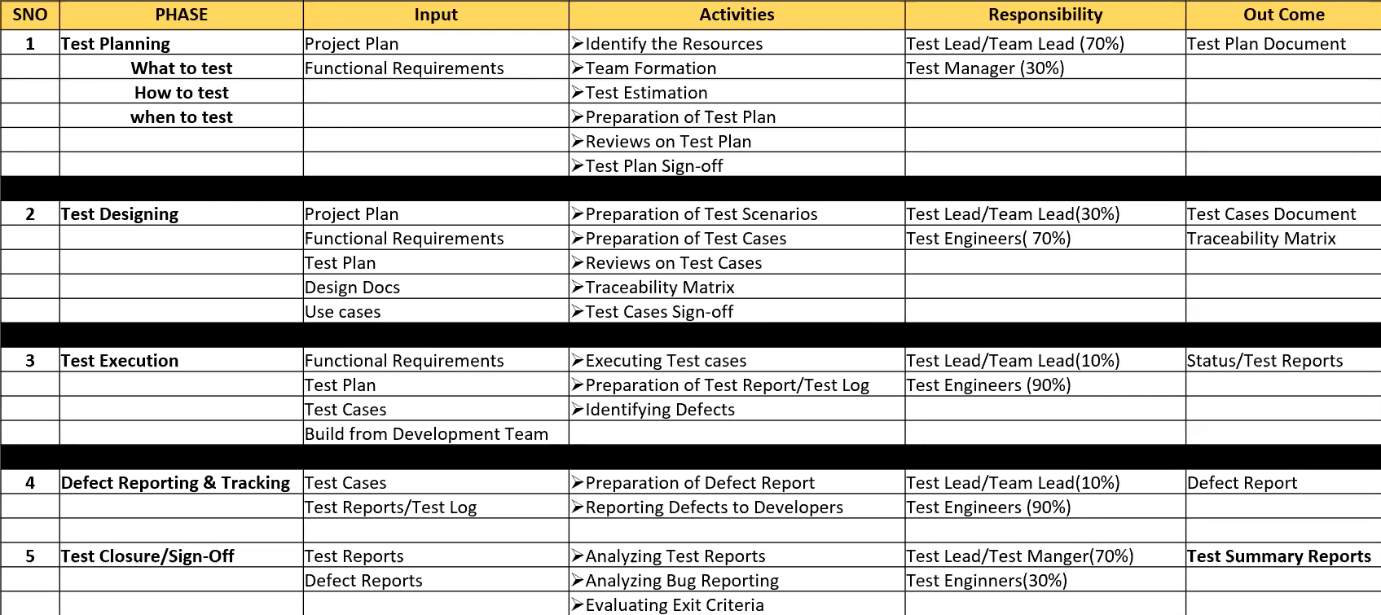


4.State Transition:  
a. In state transition technique, changes in input conditions changes the state of the application.  
b. This testing technique allows the tester to test the behaviour of an AUT.  
c. The tester can perform this action by entering various input conditions in a sequence.  
d. In state transition technique, the testing team provide both positive as well as negative test values for evaluating the system behaviour.



5.Error Guessing:  
a. Error guessing is one of the testing techniques used to find bugs in a software application based on tester’s prior experience.  
b. In error guessing we don’t follow any specific rules.  
c. It depends on tester analytical skills and experience.

26.STLC – Software Testing Life Cycle  
STLC is also a part of SDLC.  
Stages in STLC:  
1. Requirement Analysis.  
2. Test Planning.  
3. Test Design.  
4. Test Execution.  
5. Defect / Bug reporting & tracking.  
6. Test closure



Traceability matrix means mapping the requirements with test cases.

27. Test Plan Contents  
A Test plan is a document that describes the test scope, test strategy, objectives, schedule, deliverables and resources required to perform testing for a software product.

28. Test plan template contents  
🡪Overview  
🡪Scope  
1. Inclusion  
2. Test Environment  
3. Exclusion  
🡪Test strategy  
🡪Defect reporting procedure  
🡪Roles / responsibilities  
🡪Test schedule  
🡪Test deliverables  
🡪Pricing  
🡪Entry & exit criteria  
🡪Suspension and resumption criteria  
🡪Tools  
🡪Risks and mitigations  
🡪Approvals

29. Use case, Test scenario & Test case  
Use case:  
🡪Use case describes the requirement.  
🡪Use case contains THREE items.  
a. Actor, which is the user, which can be a single person or a group of people, interacting with a process.  
b. Action, which is to reach the final outcome.  
c. Goal / Outcome, which is the successful user outcome.

Test scenario:  
🡪A possible area to be tested (what to test)

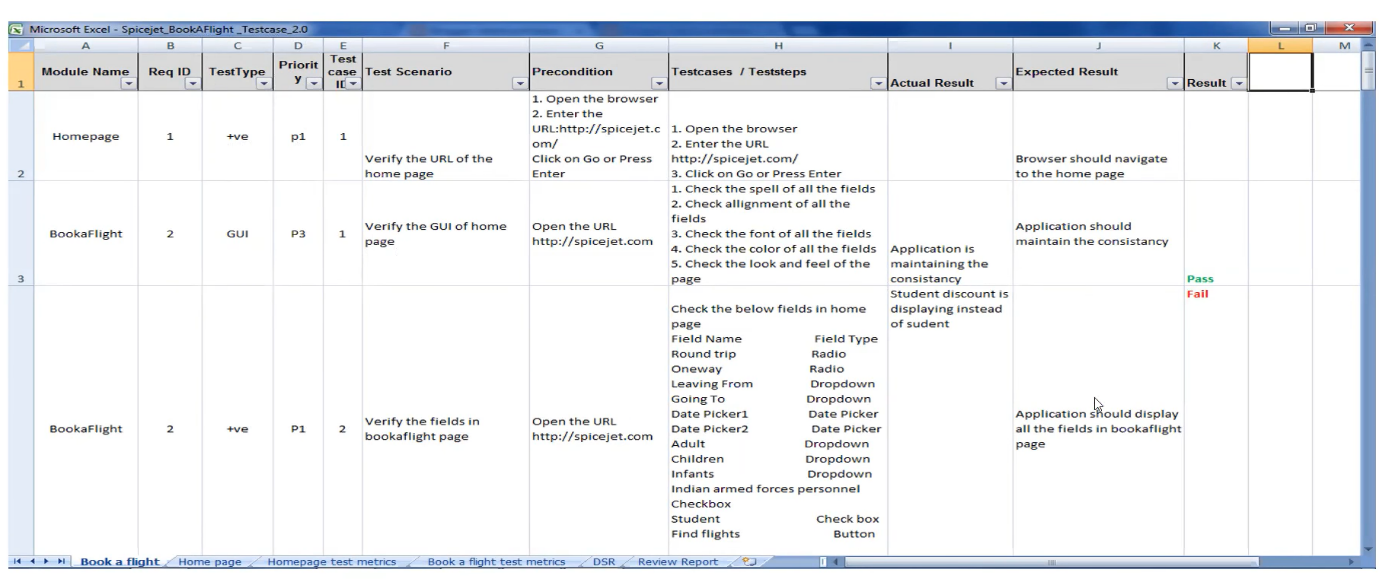
Test case:  
🡪Step by step actions to be performed to validate functionality of AUT (How to test)  
🡪Test case contains test steps, expected result & actual result.

Use case VS Test case:  
Use case: Describes functional requirement, prepared by Business Analyst (BA).  
Test case: Describes test steps / procedure, prepared by Test Engineers.

Test Scenario VS Test case:  
Test Scenario is ‘What to be tested’ and Test case is ‘How to be tested’.

30. Test Suite:  
Test Suite is a group of test cases which belongs to same category.

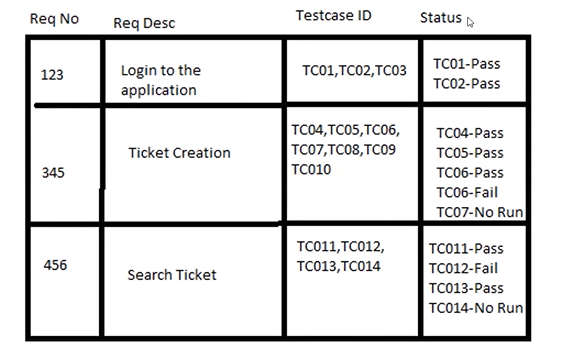
31. Test case:  
A test case is a set of actions executed to validate particular feature or functionality of your software application

Test case contents:  
🡪Testcase ID  
🡪Testcase title  
🡪Description  
🡪Pre – condition  
🡪Priority (p0, p1, p3) – order  
🡪Requirement ID  
🡪Steps / Actions  
🡪Expected result  
🡪Actual result  
🡪Test data  


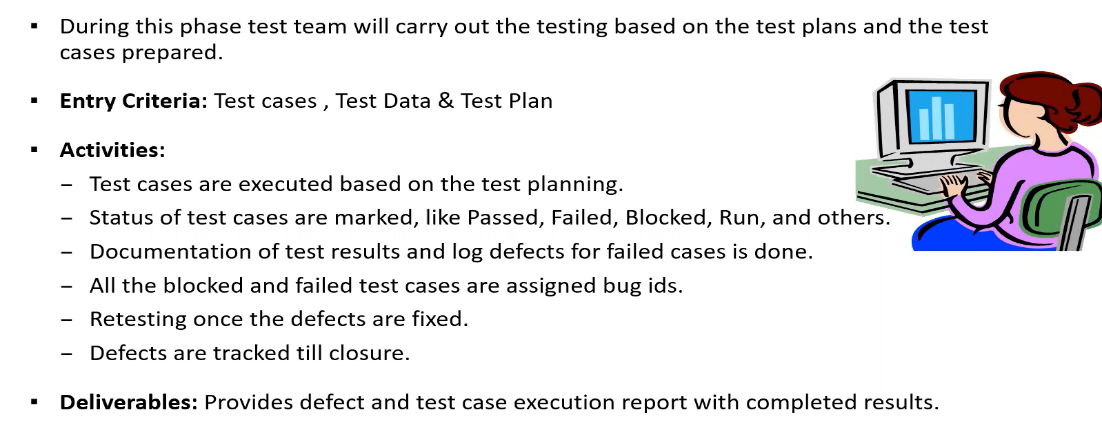
32. Requirement Traceability Matrix (RTM)  
a. What is RTM?

🡪RTM describes the mapping of requirements with the test cases.  
🡪The main purpose of RTM is to see that all test cases are covered so that no functionality should miss while during software testing.  
b. RTM – parameters include:  
🡪Requirement ID

🡪Req description  
🡪Test case ID’s



33. Test Environment:  
🡪Test environment is a platform specially build for testcase execution on the software procedure.  
🡪It is created by integrating the required software and hardware along with proper network configurations.  
🡪Test environment simulates production / real time environment.  
🡪Another name of Test Environment is ‘Test Bed’.

34. Test Execution:  


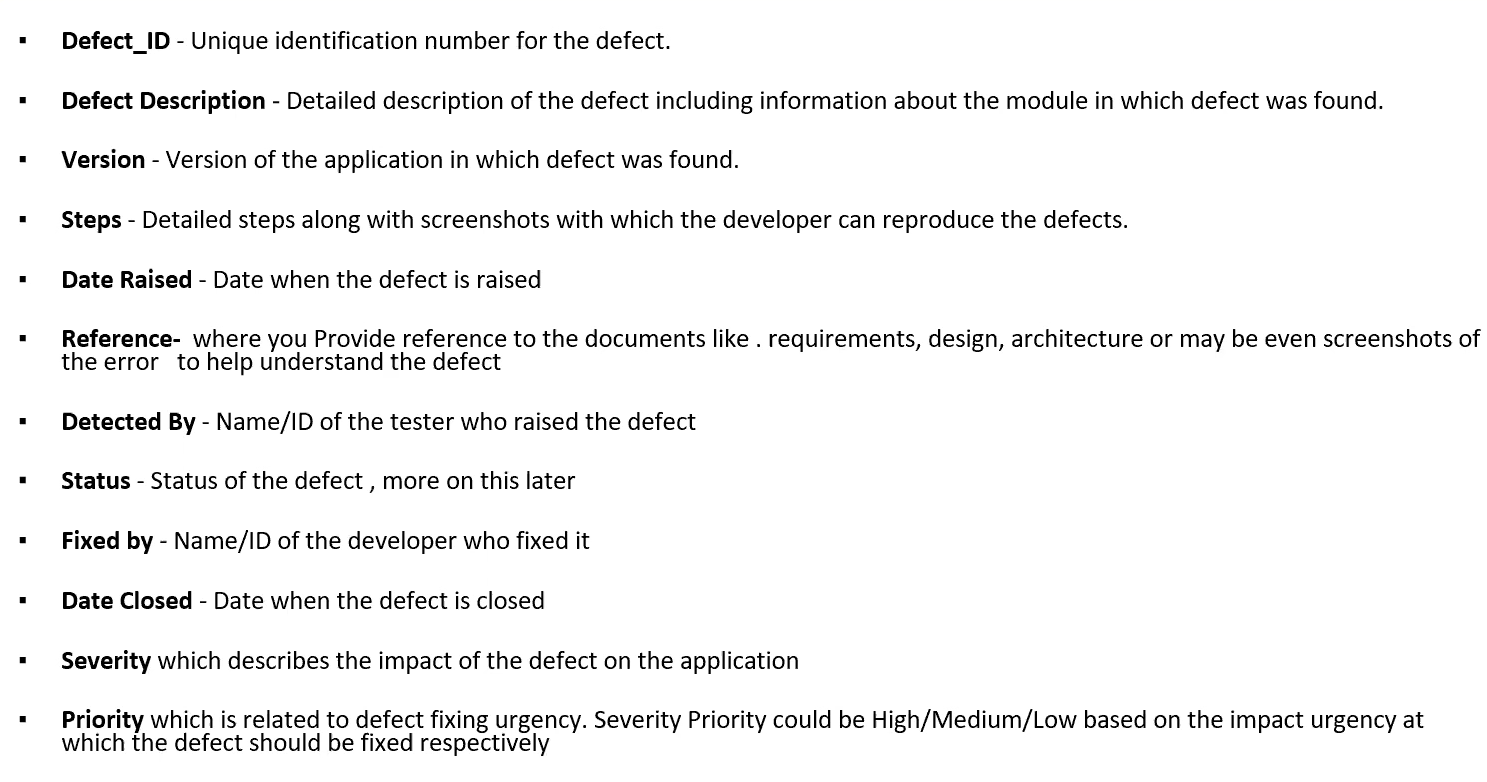
35.Guidelines for Test Execution:  
🡪The build being deployed to the QA environment is the most important part of the test execution cycle.

🡪Test execution is done in Quality Assurance (QA) environment.

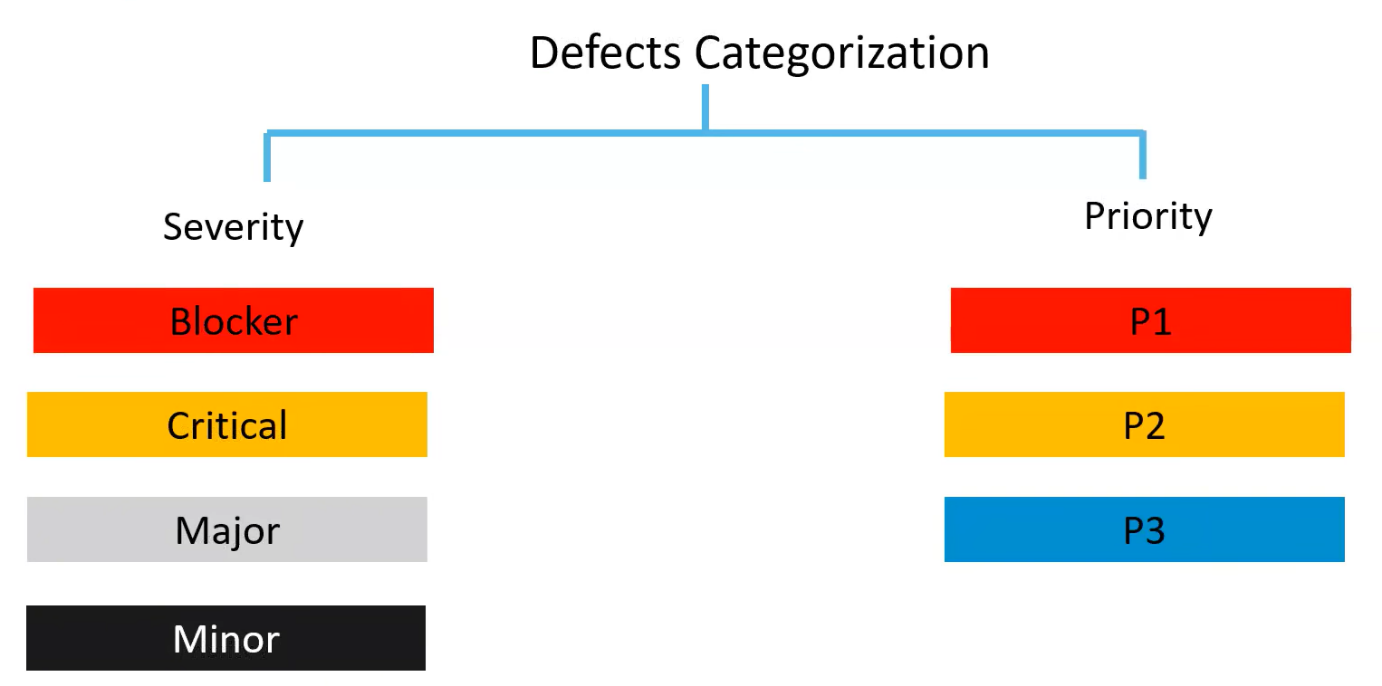
🡪Test execution happens in multiple cycles.

Defect / Bugs:  
Any mismatched functionality found in a application is called as Defect / Bug / Issue.

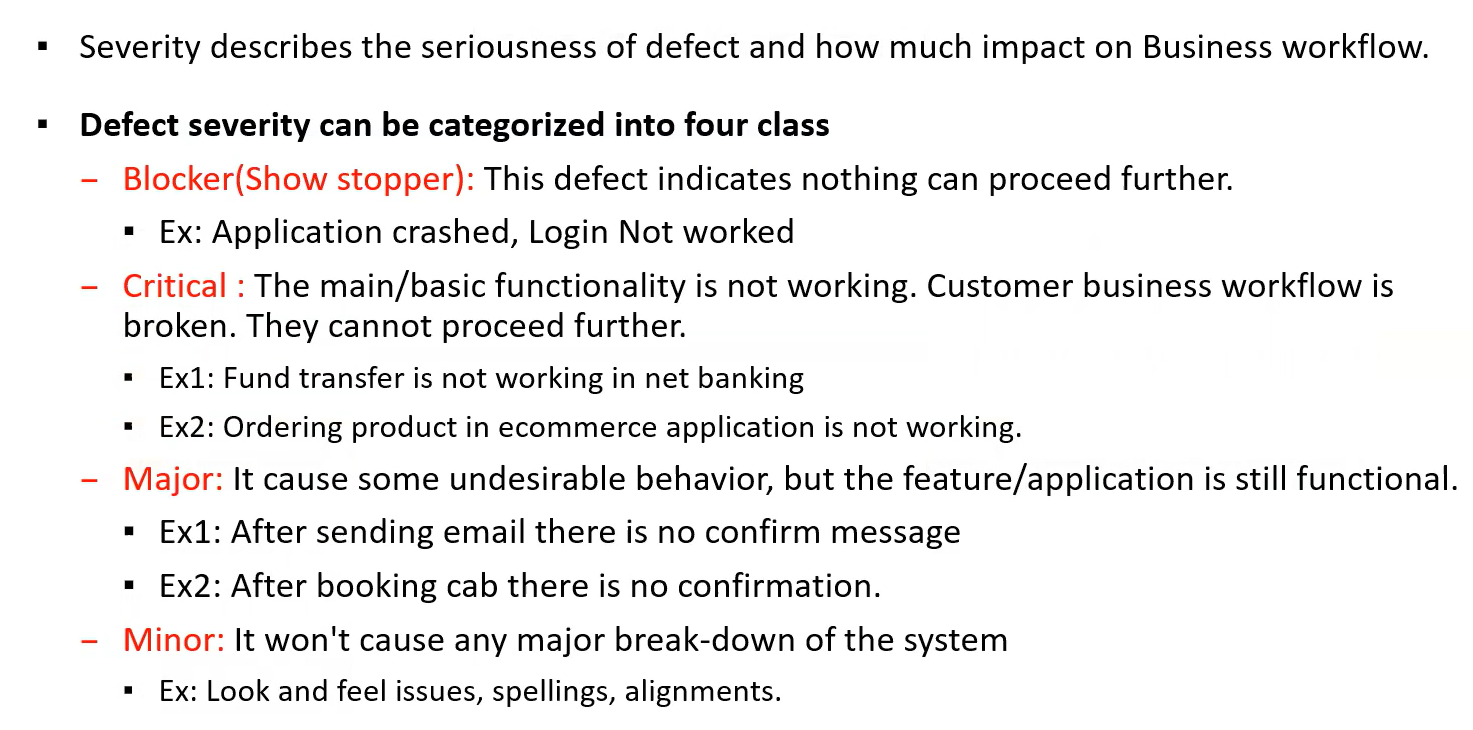
🡪Defect reporting tools:  
1. Jira  
2. Clear Quest  
3. Dev Track  
4. Quality Centre  
5. Bugzilla

Defect report contents

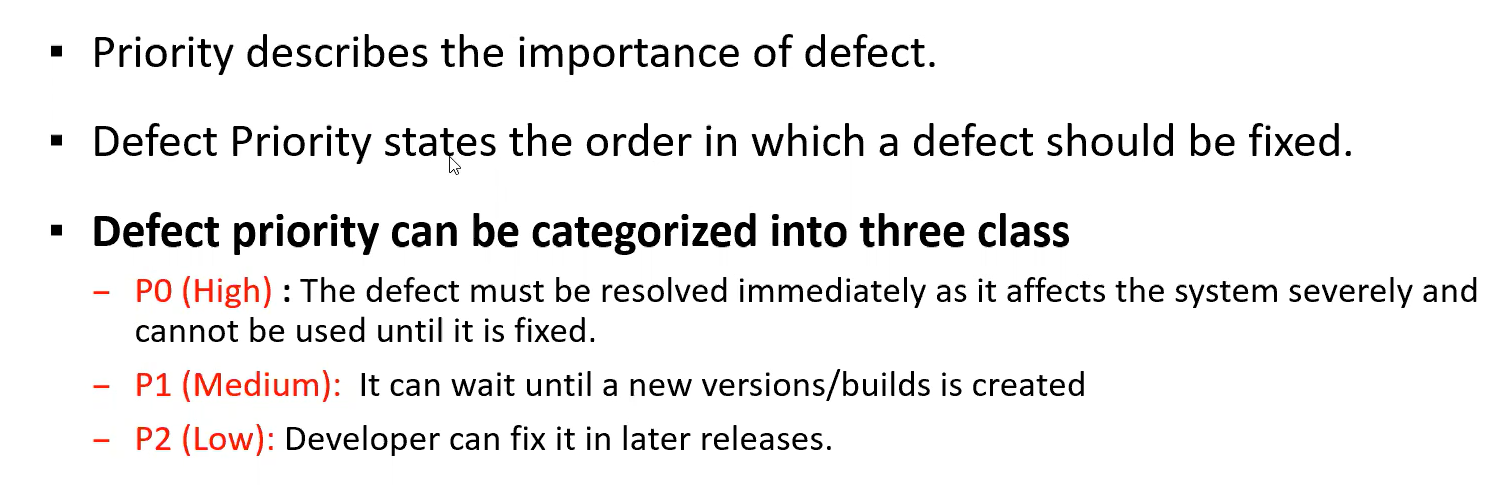
Defect classification:

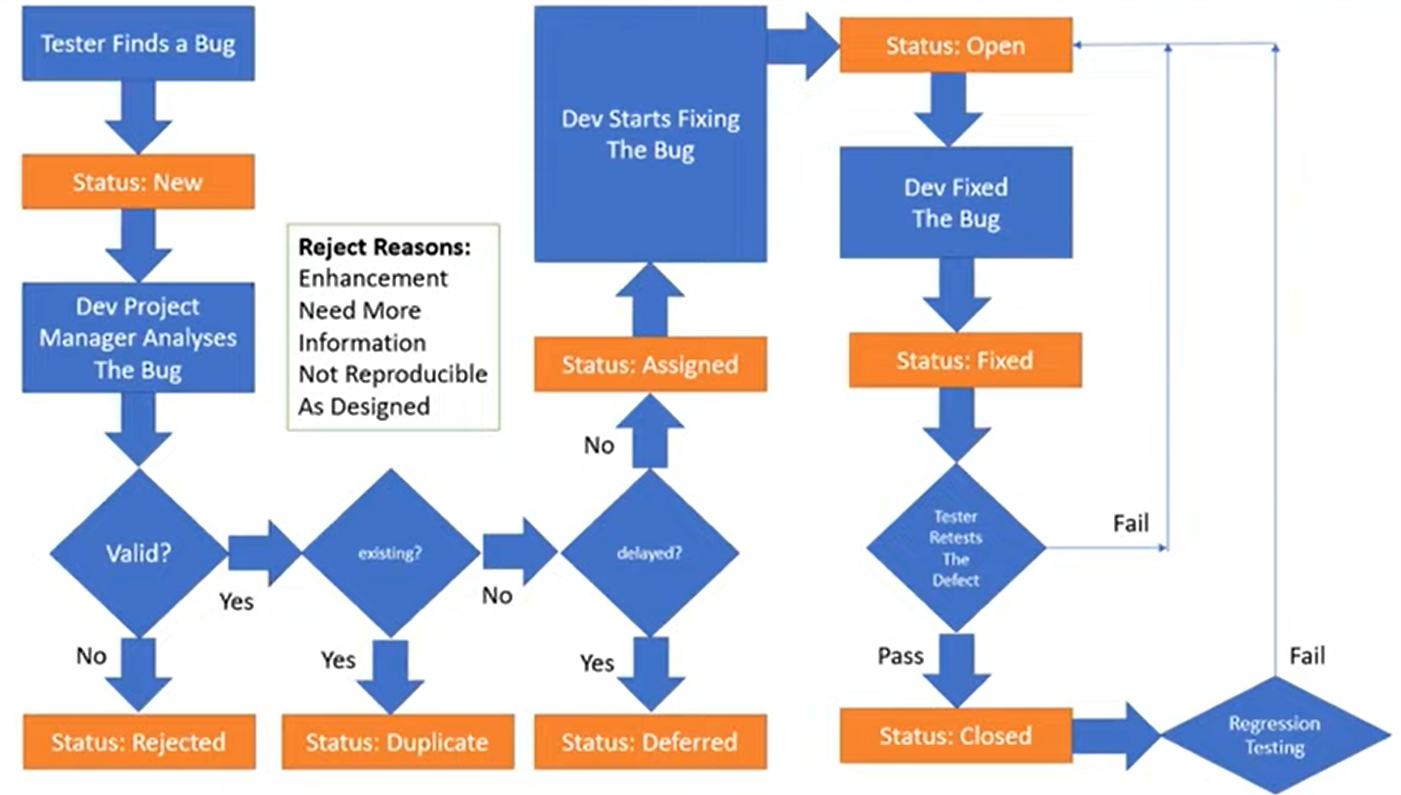


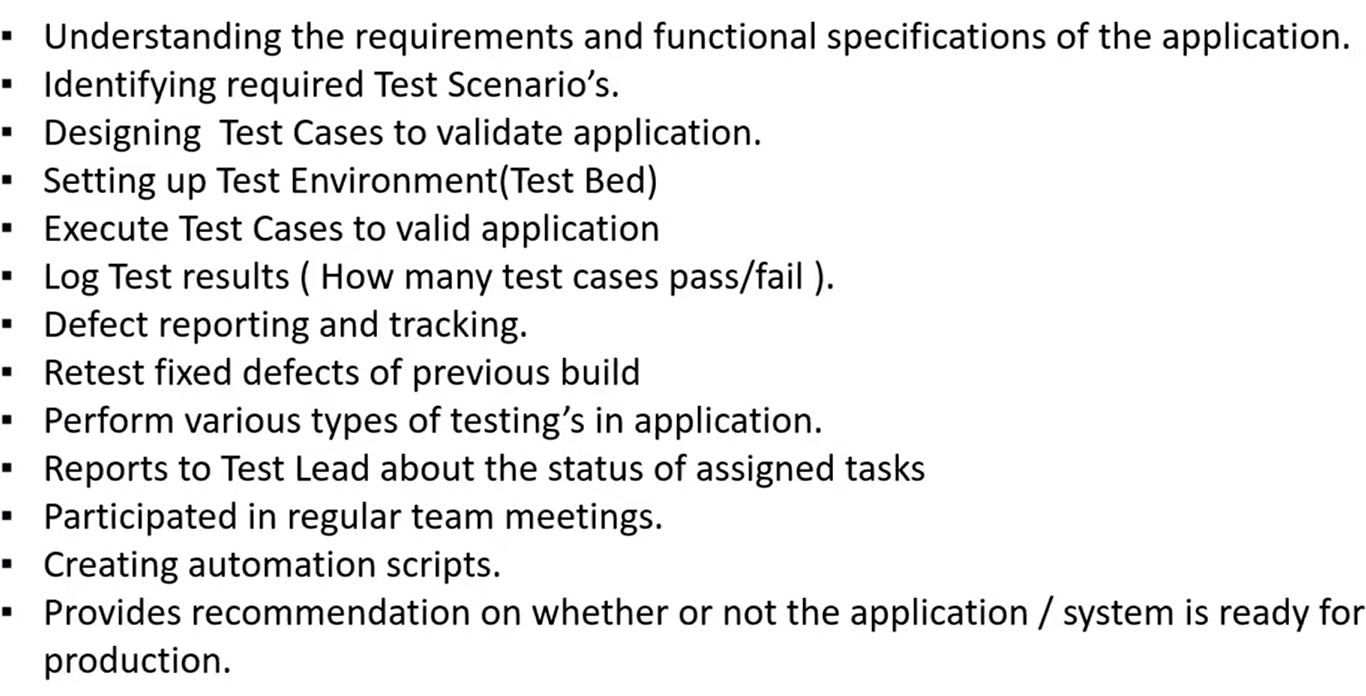
Defect severity:



Defect Priority:



Defect / Bug life cycle:  


QA Testing activities:  


Principles of software testing:  
