Assignment - Module-2

1] What is Exploratory Testing?

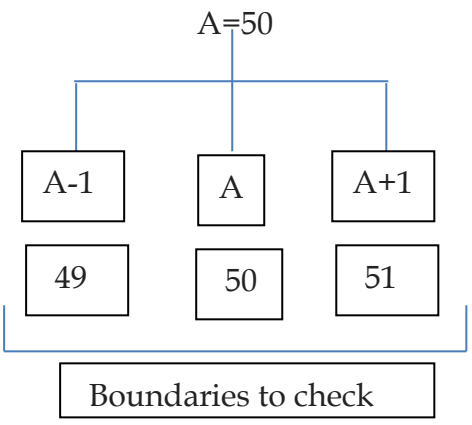
- Exploratory Testing is a type of software testing where Test cases are not created in advance but testers check system on the fly. They thought ideas about what to test before test execution. The focus of exploratory testing is more on testing as a "thinking" activity.
- Exploratory Testing is widely used in Agile models and is all about discovery, investigation, and learning. It emphasizes personal freedom and responsibility of the individual tester.

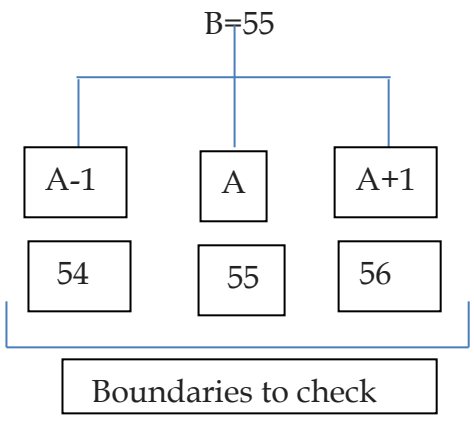
2] What is Traceability Matrix?

- Traceability matrix is a table type document that is used in the development of software application to trace requirements. It can be used for both forward (from Requirements to Design or Coding) and backward (from Coding to Requirements) tracing. It is also known as Requirement Traceability Matrix (RTM).
- It is prepared before the test execution process to make sure that every requirement is covered in the form of a Test case so that we don't miss out any testing.

3] What is Boundary value testing?

- Boundary value analysis is a methodology designing test cases to check the boundaries between the partitions. It is a method which refines equivalence partitioning.
- Boundary value analysis generates test cases the highlights error better than equivalence partitioning.
- For example, BVA for the range 50 to 55.B=55





4] What is Equivalence Partitioning testing?

- Equivalence partitioning testing is a methodology to design the test cases where divide your range into various class or partitions and from each classes you will test for single value. If that single or representative value is passed then the whole partition or class will be passed. If that single or representative value is failed then the whole partition or class will be failed.
- For example, checking EP for the range 1 to 500.

Partition of range as equivalent	Test value	Pass / Fail
-100 to 0	-73	Fail
1 to 100	60	Pass
101 to 200	111	Pass
201 to 300	226	Pass
301 to 400	305	Pass
401 to 500	451	Pass
501 to 600	566	Fail

5] What is Integration testing?

- Integration testing is a level of software testing process where individual unit or components are combined and tested as a group. The purpose of level of this testing is to expose faults in the interaction between integrated units or components.
- Integration testing becomes necessary to verify the software modules work in unity.
- Integration testing can be levelled in two parts 1) Component Integration Testing and 2) System Integration Testing. Where Component Integration is responsible to expose defects in the interfaces and interactions between integrated components. And System Integration testing is to validate all the software module dependencies.

6] What determines the level of risk?

- A risk is a factor that could result in future negative consequences.
- Risks are of two types:
  Project Risks
  Product Risk
- Example of Project risk is Senior Team Member leaving the project abruptly.

Every risk is assigned a likelihood i.e. chance of it occurring, typically on a scale of 1 to 10. Also the impact of that risk is identified on a scale of 1- 10 .

- Example of product risks would be Flight Reservation system not installing in test environment.
- Mitigation in this case would be conducting a smoke or sanity testing. Accordingly you will make changes in your scope items to include sanity testing

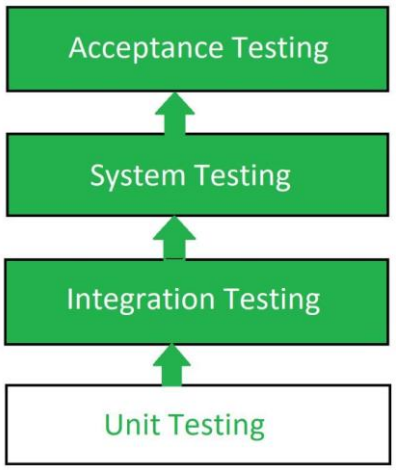
7] What is Alpha testing?

- Alpha Testing is a type of software testing performed to identify bugs before releasing the software product to the real users or public. It is a type of acceptance testing.
- The main objective of alpha testing is to refine the software product by finding and fixing the bugs that were not discovered through previous tests.
- This testing is referred to as an alpha testing only because it is done early on, near the end of the development of the software, and before Beta Testing.

8] What is beta testing?

- Beta Testing is performed by real users of the software application in a real environment. Beta testing is one of the types of User Acceptance Testing.
- A Beta version of the software, whose feedback is needed, is released to a limited number of end-users of the product to obtain feedback on the product quality. Beta testing helps in minimization of product failure risks and it provides increased quality of the product through customer validation. It is the last test before shipping a product to the customers. One of the major advantages of beta testing is direct feedback from customers.

9] What is component testing?



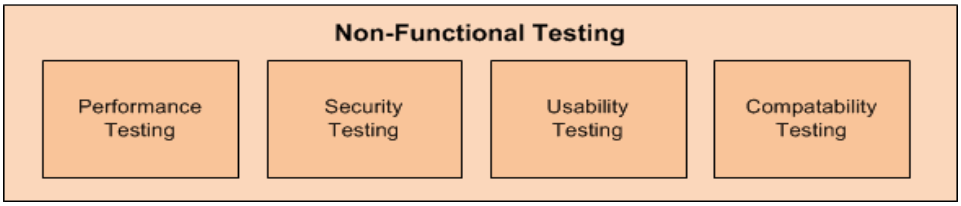
- Unit Testing is a level of the software testing process where individual units/components of a software/system are tested.
- Unit testing is the first level of testing and is performed prior to Integration Testing.
- Sometimes known as Unit Testing, Module Testing or Program Testing

10] What is functional system testing?

- Functional Testing is a type of Software Testing in which the system is tested against the functional requirements and specifications.
- A requirement that specifies a function that a system or system component must perform
- A Requirement may exist as a text document and/or a model

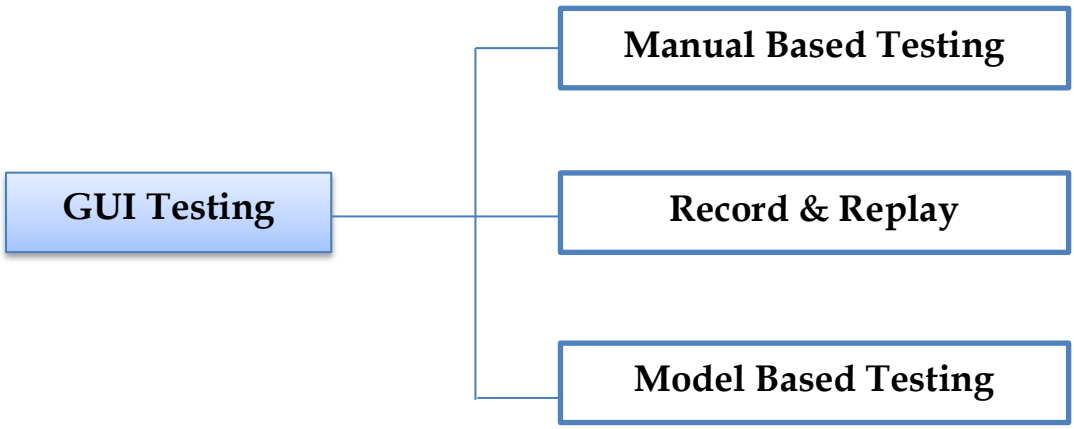
11] What is Non-Functional Testing?

- **Non-Functional Testing** is defined as a type of Software testing to check non-functional aspects (performance, usability, reliability, etc) of a software application.
- An excellent example of non-functional test would be to check how many people can simultaneously login into a software.
- Non-functional testing is equally important as functional testing and affects client satisfaction.



12] What is GUI Testing?

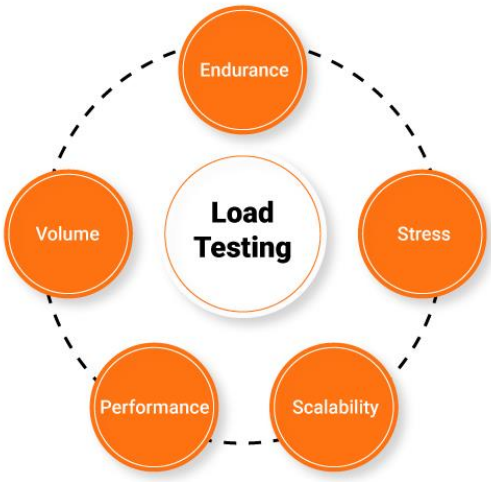
- Graphical User Interface (GUI) testing is the process of testing the system's GUI of the System under Test. GUI testing involves checking the screens with the controls like menus, buttons, icons, and all types of bars – tool bar, menu bar, dialog boxes and windows etc.
- GUI is what the user sees.



13] What is Adhoc testing?

- Adhoc testing is an informal testing type with an aim to break the system.
- It does not follow any test design techniques to create test cases.
- In fact is does not create test cases altogether!
- Main aim of this testing is to find defects by random checking.
- Adhoc testing can be achieved with the testing technique called Error Guessing.

14] What is load testing?



Load testing - Its a performance testing to check system behavior under load. Testing an application under heavy loads, such as testing of a web site under a range of loads to determine at what point the system's response time degrades or fails.

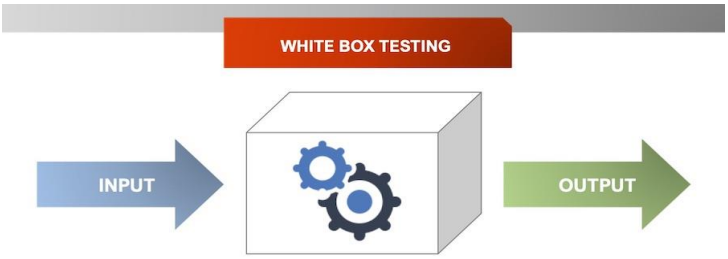
Load testing is a kind of performance testing which determines a system's performance under real-life load conditions. This testing helps determine how the application behaves when multiple users access it simultaneously.

15] What is stress Testing?

- System is stressed beyond its specifications to check how and when it fails. Performed under heavy load like putting large number beyond storage capacity, complex database queries, continuous input to system or database load.
- Stress testing is used to test the stability & reliability of the system. This test mainly determines the system on its robustness and error handling under extremely heavy load conditions. Stress testing is also known as endurance testing.

16] What is white box testing and list the types of white box testing?

- Testing based on an analysis of the internal structure of the component or system.
- Structure-based testing technique is also known as 'white-box' or 'glass-box' testing technique because here the testers require knowledge of how the software is implemented, how it works.



White box Testing Techniques

- Statement coverage
- Branch Coverage
- Decision Coverage

17] What is black box testing? What are the different black box testing techniques?

- Black-box testing: Testing, either functional or non-functional, without reference to the internal structure of the component or system.
- Specification-based testing technique is also known as 'black-box' or input/output driven testing techniques because they view the software as a black-box with inputs and outputs.
- The testers have no knowledge of how the system or component is structured inside the box. In black-box testing the tester is concentrating on what the software does, not how it does it.

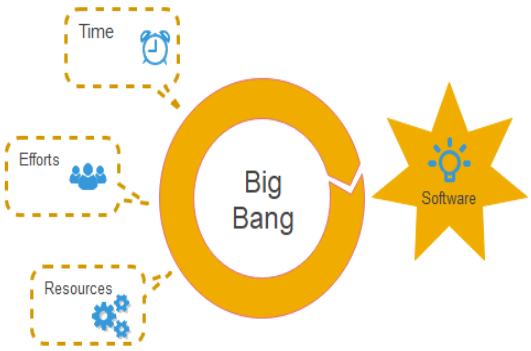
Black box Testing Techniques

- Equivalence Partitioning
- Boundary Value Analysis
- Decision Tables
- State Transition Testing

18] Mention what are the categories of defects?

- **Data Quality/Database Defects:** Deals with improper handling of data in the database.
- **Critical Functionality Defects:** The occurrence of these bugs hampers the crucial functionality of the application.
- **Functionality Defects:** These defects affect the functionality of the application.
- **Security Defects:** Application security defects generally involve improper handling of data sent from the user to the application. These defects are the most severe and given highest priority for a fix.
- **User Interface Defects:** As the name suggests, the bugs deal with problems related to UI are usually considered less severe.

19] Mention what big bang testing is?



In Big Bang integration testing all components or modules is integrated simultaneously, after which everything is tested as a whole.

Big Bang testing has the advantage that everything is finished before integration testing starts.

20] What is the purpose of exit criteria?

- ✓ All the high priority bugs are fixed.
- ✓ The rate at which bugs are found is too small.
- ✓ The testing budget is exhausted.
- ✓ The project duration is completed.
- ✓ The risk in the project is under acceptable limit.

21] When should "Regression Testing" be performed?

Regression testing should be carried out:

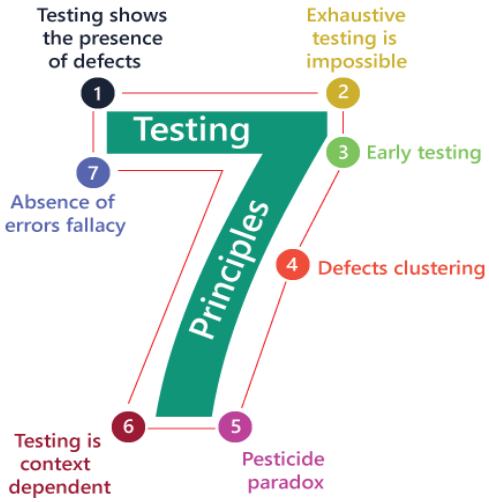
- ✓ when the system is stable and the system or the environment changes.
- ✓ when testing bug-fix releases as part of the maintenance phase.
- ✓ It should be applied at all Test Levels.
- ✓ It should be considered complete when agreed completion criteria for regression testing have been met.
- ✓ Regression test suites evolve over time and given that they are run frequently are ideal candidates for automation.
- ✓ Change in requirements and code is modified according to the requirement
- ✓ New feature is added to the software
- ✓ Defect fixing
- ✓ Performance issue fix

22] What is 7 key principles? Explain in detail?

❖ **Testing shows the presence of defects.**

Testing can show that defects are present, but cannot prove that there are no defects.

Testing reduces the probability of undiscovered defects remaining in the software but, even if no defects are found, it is not a proof of correctness.



❖ **Exhaustive testing is impossible.**

- Testing everything including all combinations of inputs and preconditions is not possible. So, instead of doing the exhaustive testing we can use risks and priorities to focus testing efforts. we cannot test everything (i.e. all combinations of inputs and pre-conditions)

❖ **Early Testing**

- Testing activities should start as early as possible in the software or system development life cycle, and should be focused on defined objectives. Testing activities should start as early as possible in the development life cycle.

❖ **Defect Clustering**

- A small number of modules contain most of the defects discovered during pre-release testing, or are responsible for the most operational failures. Defects are not evenly spread in a system. They are 'clustered'

❖ **Pesticide Paradox**

- If the same tests are repeated over and over again, eventually the same set of test cases will no longer find any new defects.
- To overcome this "pesticide paradox", the test cases need to be regularly reviewed and revised, and new and different tests need to be written to exercise different parts of the software or system to potentially find more defects.

❖ **Testing is Context Dependent**

- Testing is a context-dependent principle states that we have multiple fields such as e-commerce websites, commercial websites, and so on are available in the market. There is a definite way to test the commercial site as well as the e-commerce websites because every application has its own needs, features, and functionality.

❖ **Absence of errors fallacy**

- Once the application is completely tested and there are no bugs identified before the release, so we can say that the application is 99 percent bug-free. The absence of error fallacy means identifying and fixing the bugs would not help if the application is impractical and not able to accomplish the client's requirements and needs.

23] Difference between QA v/s QC v/s Tester.

QA (Quality Assurance)	QC (Quality Control)	Tester
QA ensures the implementation of processes, procedures	QC ensures the verification of developed software with respect to documented requirements.	Tester ensures the identification of bugs/error/defects in the Software.
Focuses on processes and procedures rather than conducting actual testing on the system.	Focuses on actual testing by executing Software with intend to identify bug/defect through implementation of procedures and process.	Focuses on actual testing.
Process oriented activities.	Product oriented activities.	Product oriented activities.
Preventive activities.	Corrective process.	Preventive process.
QA can be considered a subset of Software Test Life Cycle (STLC).	QC can be considered as the subset of Quality Assurance.	Testing is the subset of Quality Control.

24] Difference between Smoke and Sanity.

Smoke Testing	Sanity Testing
Whenever a new feature comes into the picture, we perform the smoke testing to check the functionalities are working fine.	Whenever code changes or bugs fixes comes into the picture, we perform the sanity testing to ensure no issues are introduced duo to changes.
Smoke testing is documented or scripted.	Sanity testing is undocumented or unscripted.
Smoke testing is a part of Regression testing.	Sanity testing is a part of Acceptance testing.
This testing is performed by the developers or testers.	This testing is usually performed by testers
Smoke testing exercises the entire system from end to end.	Sanity testing exercises only the particular component of the entire system.
Example : General Health Check up	Example : Specialized Health Check up

25] Difference between verification and Validation.

Verifcation	Validation
Verification is a process which is performed at development level.	Validation is a process which is performed at testing level.
Verification phases are : Business Requirement Analysis System Design/ System Requirement Architectural Design (Technical Specification) Module Design (Program Specification)	Validation Phases are : Unit Testing Integration Testing System Testing Acceptance Testing
It is the process of evaluating product of development to check whether the specified requirements meet or not.	It is the process of evaluating the product of development to check whether it satisfied business requirements or not.
Verification can be achieved by asking Are you building a product right.	Validation can be achieved by asking Are you building a right product.
The evaluation of verification can be achieved by planning, Requirement specification, Design Specification, Code specification, and test cases.	The evaluation of validation can be achieved as an actual product.
Verification activities are Reviews and Inspections.	Validation activity is Testing.

26] Explain types of Performance testing.

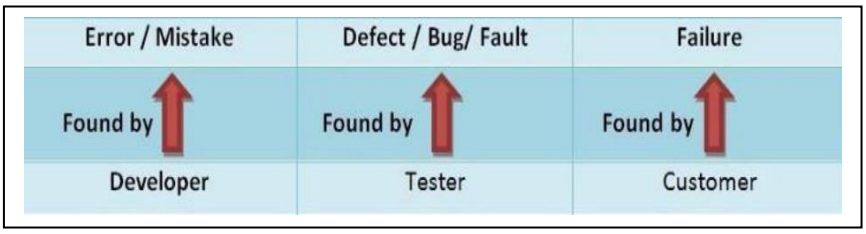
- **Load Testing –** Load Testing checks the application's ability to perform under expected user loads. The main objective of this testing is to identify performance bottlenecks before the software application goes live.
-
- **Stress Testing –** This testing involves testing an application under extreme workloads to see how it handles high traffic or data processing. Here, it identifies the breaking point of an application.

- **Endurance Testing –** This testing is done to make sure the software can handle the expected load over a long period of time.
- **Spike Testing –** It is used to test the software's reaction to sudden large spikes in the load generated by users.
- **Volume Testing** – Here, a large amount of data is populated in a database and the overall software system's behavior is monitored. The main objective of this testing is to check the software application's performance under varying database volumes.

- **Scalability Testing –** The main objective of scalability testing is to determine the software application's effectiveness in "scaling up" to support an increase in user load.

27] What is Error, Defect, Bug and failure?

- A mistake in codding is called an error. An error found by the tester is called Defect. Defect accepted by the Development team is known as Bug. Build does not meet requirements is called a bug.



28] Difference between Priority and Severity.

Severity	Priority
▪ Severity is absolute and Customer-Focused	▪ Priority is Relative and Business-Focused.
▪ Severity is a term that denotes how severely a defect can affect the functionality of the software.	▪ Priority is a term that defines how fast we need to fix a defect.
▪ Severity is basically a parameter that denotes the total impact of a given defect on any software.	▪ Priority is basically a parameter that decides the order in which we should fix the defects.
▪ Severity relates to the standards of quality.	▪ Priority relates to the scheduling of defects to resolve them in software.
▪ The value of severity is objective.	▪ The value of priority is subjective.
▪ The value of Severity changes continually from time to time.	▪ The value of Priority changes from time to time.
▪ The testing engineer basically decides a defect's severity level.	▪ The product manager basically decides a defect's priority level.
▪ There are 5 types of Severities: Cosmetic, Minor, Moderate, Major, and Critical.	▪ There are 3 types of Priorities: High, Medium, and Low.
▪ Severity is given by QA testers.	▪ Priority is given by Test lead or Project Manager.
▪ The test engineer determines the severity level of the defect.	▪ Priority of defect is decided in discussion with manager/client.

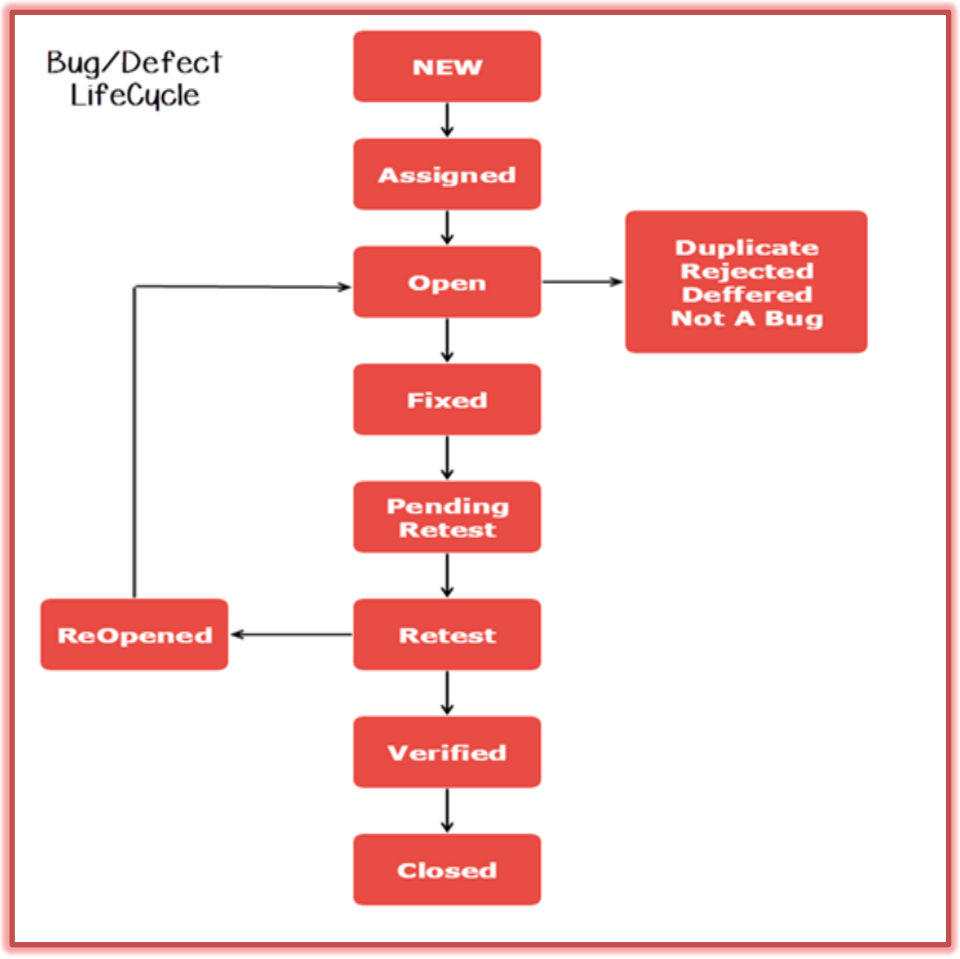
29] What is Bug Life Cycle?

- **Defect Life Cycle** or Bug Life Cycle in software testing is the specific set of states that defect or bug goes through in its entire life. The purpose of Defect life cycle is to easily coordinate and communicate current status of defect which changes to various assignees and make the defect fixing process systematic and efficient.

✓ **Defect States Workflow**

- The number of states that a defect goes through varies from project to project. Below lifecycle diagram, covers all possible states

- **New:** When a new defect is logged and posted for the first time. It is assigned a status as NEW.
- **Assigned:** Once the bug is posted by the tester, the lead of the tester approves the bug and assigns the bug to the developer team
- **Open:** The developer starts analysing and works on the defect fix
- **Fixed:** When a developer makes a necessary code change and verifies the change, he or she can make bug status as "Fixed."
- **Pending retest:** Once the defect is fixed the developer gives a particular code for retesting the code to the tester. Since the software testing remains pending from the testers end, the status assigned is "pending retest."
- **Retest:** Tester does the retesting of the code at this stage to check whether the defect is fixed by the developer or not and changes the status to "Re-test.



- **Verified:** The tester re-tests the bug after it got fixed by the developer. If there is no bug detected in the software, then the bug is fixed and the status assigned is "verified."
- **Reopen:** If the bug persists even after the developer has fixed the bug, the tester changes the status to "reopened". Once again the bug goes through the life cycle.
- **Closed:** If the bug is no longer exists then tester assigns the status "Closed."
- **Duplicate:** If the defect is repeated twice or the defect corresponds to the same concept of the bug, the status is changed to "duplicate."
- **Rejected:** If the developer feels the defect is not a genuine defect then it changes the defect to "rejected."
- **Deferred:** If the present bug is not of a prime priority and if it is expected to get fixed in the next release, then status "Deferred" is assigned to such bugs
- **Not a bug:** If it does not affect the functionality of the application then the status assigned to a bug is "Not a bug".

30] Explain the difference between Functional testing and Non Functional testing.

Functional Testing	Non Functional Testing
Functional testing is performed using the functional specification provided by the client and verifies the system against the functional requirements.	Non-functional testing checks the performance, reliability, scalability and other non-functional aspects of the software system.
Functional testing is executed first.	Non functional testing should be executed after Functional testing performed.
Business requirements are the inputs to functional testing	Performance parameters like speed , scalability are inputs to non-functional testing.
Functional testing describes what the product does.	Nonfunctional testing describes how good the product works.
Easy to do manual testing	Tough to do manual testing
Types of Functional testing are · Unit Testing · Smoke Testing · Sanity Testing · Integration Testing · White box testing · Black Box testing · User Acceptance testing · Regression Testing	Types of Nonfunctional testing are · Performance Testing · Load Testing · Volume Testing · Stress Testing · Security Testing · Installation Testing · Penetration Testing · Compatibility Testing · Migration Testing

31] To create HLR & Test Case of

→ Instagram (First Page).

HLR for Instagram	**Click →**	HLR_INSTAGRAM
TestCase for Instagram	**Click →**	TEST_CASE_INSTAGRAM

→ Facebook (First Page).

HLR for Facebook	**Click →**	HLR_FACEBOOK
TestCase for Facebook	**Click →**	TEST_CASE_FACEBOOK

32] What is the difference between the STLC (Software Testing Life Cycle) and SDLC (Software Development Life Cycle)?

SDLC	STLC
SDLC is mainly related to software development.	STLC is mainly related to software testing.
It helps in developing good quality software.	It helps in making the software defects free.
SDLC phases are completed before the STLC phases.	STLC phases are performed after SDLC phases.
Goal of SDLC is to complete successful development of software.	Goal of STLC is to complete successful testing of software.
In SDLC, development team makes the plans and designs based on the requirements.	In STLC, testing team(Test Lead or Test Architect) makes the plans and designs.
Coders create a well-organized development plan.	QA team defines the test plan.

33] What is the difference between test scenarios, test cases, and test script?

Test Scenario	Test Case	Test Script
Test scenario is any functionality that can be tested.	Test case is a set of actions executed to verify particular features or functionality.	Test script is a set of instructions to test an application functionality.
Is derived from test artifacts like Business Requirement Specification (BRS) and Software Requirement Specification (SRS).	Is mostly derived from Test Scenario.	Is derived from Test Cases.
Helps test the end-to-end functionality in an Agile way.	Helps in exhaustive testing of application.	Helps to test specific things repeatedly.

Test Scenario	Test Case	Test Script
Is more focused on what to test.	Is focused on what to test and how to test.	Is focused on the expected result.
Takes less time and fewer resources to create.	Requires more resources and time.	Requires less time or testing but more resources for scripts creating and updating.
Includes an end-to-end functionality to be tested.	Includes test steps, data, expected result for the test.	Includes different commands to develop a script.
The main task is to check the full functionality of a software application.	The main task is to verify compliance with the applicable standards, guidelines and customer requirements.	The main task is to verify that nothing is skipped, and the results are true as the desired testing plan.
Allows quickly accessing the testing scope.	Allows detecting errors and defects.	Allows carrying out an automatic execution to test cases.

34] Explain what Test Plan is? What is the information that should be covered.

➢ **Test Plan**

- Test Plan is a document describing the scope, approach, resources and schedule of intended test activities.
- Determining the scope and risks, and identifying the objectives of testing.
- Defining the overall approach of testing (the test strategy), including the definition of the test levels and entry and exit criteria.

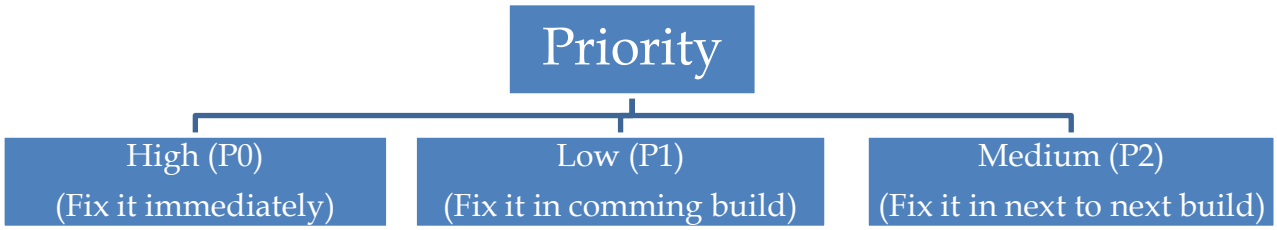
Test Plan should cover all tests planning process which is also called Test Plan Template which includes.

- Test Planning
- Test Planning Strategy
- Test Planning Factors
- Test Planning Activities
- Exit Criteria.

35] What is priority?

Priority

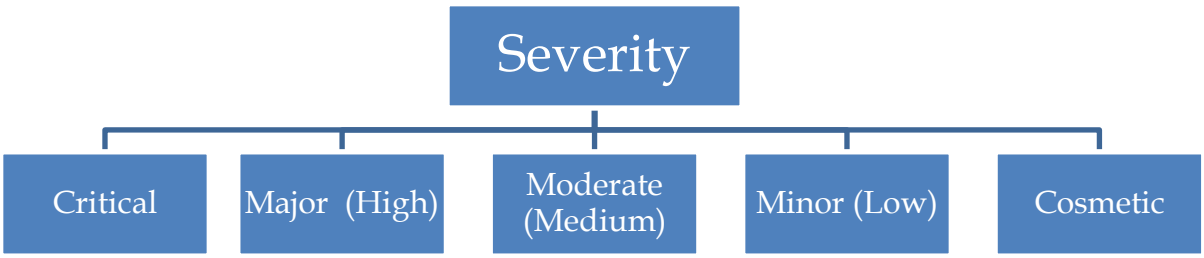
- If you are raising any bug for any application, how soon you want the developer to fix that bug is called priority.
- Priority is considered as customer's point of view. But priority can be set by the QA tester. Later on it can be changed by project manager.
- The importance given to the bug to fix it that is priority.
- The high priority indicates that the bug to fix it first.



36] What is severity?

Severity

- The impact of Defect /bug on the customer business workflow is known as Severity.
- If that impact is more then, there is high severity.
- If that impact is less then, there is low severity.



37] Bug categories are…

Types of Defect/Bug:

- **Data Quality/Database Defects:** Deals with improper handling of data in the database.
- **Critical Functionality Defects:** The occurrence of these bugs hampers the crucial functionality of the application. Examples: - Exceptions
- **Functionality Defects:** These defects affect the functionality of the application.
- Continuous execution of loops
- **Security Defects:** Application security defects generally involve improper handling of data sent from the user to the application. These defects are the most severe and given highest priority for a fix.
- **User Interface Defects:** As the name suggests, the bugs deal with problems related to UI are usually considered less severe.

38] Advantage of Bugzila .

Pros of Bugzilla

- Open source, free bug tracking tool.
- Automatic Duplicate Bug Detection.
- Search option with advanced features.
- File/Modify Bugs By Email.
- Move Bugs Between Installs.
- Multiple Authentication Methods
- Time Tracking.
- Automated bug reporting; has an API to interact with system.
- Integrated email capabilities.
- Detailed permissions system.
- Optimized database structure to enhance performance.
- Robust security.
- Powerful query tool.
- Ideal for small projects.

<

39] Difference between priority and severity.

Severity	Priority
▪ Severity is absolute and Customer-Focused	▪ Priority is Relative and Business-Focused.
▪ Severity is a term that denotes how severely a defect can affect the functionality of the software.	▪ Priority is a term that defines how fast we need to fix a defect.
▪ Severity is basically a parameter that denotes the total impact of a given defect on any software.	▪ Priority is basically a parameter that decides the order in which we should fix the defects.
▪ Severity relates to the standards of quality.	▪ Priority relates to the scheduling of defects to resolve them in software.
▪ The value of severity is objective.	▪ The value of priority is subjective.
▪ The value of Severity changes continually from time to time.	▪ The value of Priority changes from time to time.
▪ The testing engineer basically decides a defect's severity level.	▪ The product manager basically decides a defect's priority level.
▪ There are 5 types of Severities: Cosmetic, Minor, Moderate, Major, and Critical.	▪ There are 3 types of Priorities: High, Medium, and Low.
▪ Severity is given by QA testers.	▪ Priority is given by Test lead or Project Manager.
▪ The test engineer determines the severity level of the defect.	▪ Priority of defect is decided in discussion with manager/client.

40] What are the different Methodologies in Agile Development Model?

- **Individuals and interactions, Over processes and tools**
- Suppose the team finds any issue in software then they search for another process or tool to resolve the issue. But, in Agile, it is preferable to interact with client, manager or team regarding issue and make sure that the issue gets resolved.

- **Working software, Over comprehensive documentation**
- Documentation is needed, but working software is much needed. Agile is not saying that documentation is not needed, but working software is much needed. For example, you have 20-page documents, but you do not have a single prototype of the software. In such a case, the client will not be happy because, in the end, the client needs a document.

- **Customer collaboration, Over contract negotiation**
- Contract negotiation is important as they make the budget of software, but customer collaboration is more important than over contract negotiation. For example, if you stuck with the requirements or process, then do not go for a contract which we have negotiated. You need to interact with the customer, gather their requirements.

- **Responding to change, over following a plan**
- In the waterfall model, everything is planned, i.e., at what time, each phase will be completed. Sometimes you need to implement the new requirements in the middle of the software, so you need to be versatile to make changes in the software.

41] Explain the difference between Authorization and Authentication in Web testing. What are the common problems faced in Web testing?

- **Difference between Authorization and Authentication**

Authorization	Authentication
▪ Authorization determines what resources a user can access.	▪ Authentication verifies who the user is.
▪ Authorization always takes place after authentication.	▪ Authentication is the first step of a good identity and access management process.
▪ Authorization isn't visible to or changeable by the user.	▪ Authentication is visible to and partially changeable by the user.
▪ Authorization works through settings that are implemented and maintained by the organization.	▪ Authentication works through passwords, one-time pins, biometric information, and other information provided or entered by the user.

➢ **The common problems faced in Web testing**

- Insufficient testing for browser compatibility
- Failing to conduct thorough functional testing across mobile
- Failing to conduct thorough functional testing across desktop
- Poor data security
- Failing to provide an intuitive experience
- Not performing testing frequently enough
- Leaving digital accessibility to the last minute
- Releasing new features breaks the existing live system
- Localisation and the global experience

42] To create HLR and Test Case on this Link. https://artoftesting.com/

HLR for ArtofTesting (Contact-us)	→	**Click Here**
TestCase for ArtofTesting (Contact-us)	→	**Click Here**

43] Write a scenario of only Whatsapp chat messages.

Test Scenario for WhatsApp chat messages →	**Click Here**

44] Write a Scenario of Pen.

Test Scenario for Pen	→	**Click Here**

45] Write a Scenario of Pen Stand.

Test Scenario for Pen Stand	→	**Click Here**

46] Write a Scenario of Door.

Test Scenario for Door →	**Click Here**

47] Write a Scenario of ATM.

Test Scenario for ATM →	**Click Here**

48] When to used Usablity Testing?

- If possible, usability testing can and should be conducted on the current iteration of a product before beginning any new design work, after you've begun the strategy work around a brand new site or app.
- This will quickly identify areas for opportunity, and reduce the amount of assumptions your design team will make with regard to what the user wants.
- Additionally, after the usability tests analysis, the team should have the ability to pinpoint the steps needed to achieve the project goals with as little disruption as possible.

49] What is the procedure for GUI Testing?

- Graphical User-interface Testing or GUI testing is a process of testing the user interface of an application. The procedures for GUI testing are :

- **MANUAL BASED TESTING**
- Under this approach, graphical screens are checked manually by testers in conformance with the requirements stated in business requirements document.

- **RECORD AND REPLAY**
- GUI testing can be done using automation tools. This is done in 2 parts. During Record , test steps are captured into the automation tool. During playback, the recorded test steps are executed on the Application under Test.

- **MODEL BASED TESTING**
- A model is a graphical description of system's behavior. It helps us to understand and predict the system behavior. Models help in a generation of efficient test cases using the system requirements.

50] Write a scenario of Microwave Oven.

Test Scenario for Microwave Oven →	**Click Here**

51] Write a scenario of Coffee vending Machine.

Test Scenario for Coffee Vending Machine →	**Click Here**

52] Write a scenario of chair.

Test Scenario for Chair	→	**Click Here**

53] To Create Scenario (Positive & Negative)

Test Scenario for Facebook Chat on Mobile	→	**Click Here**
Test Scenario for Gmail (Receiving Mails)	→	**Click Here**
Test Scenario for Online shopping to buy product (flipkart)	→	**Click Here**

54] Write a scenario of Wrist Watch.

Test Scenario for Wrist Watch	→	**Click Here**

55] Write a scenario of Lift (Elevator).

Test Scenario for Lift (Elevator)	→	**Click Here**

56] Write a scenario of WhatsApp Group (generate group).

Test Scenario for WhatsApp Group (generate group)	→	**Click Here**