1. **What is Exploratory Testing?**

* Exploratory Testing is a type of software testing where testers actively explore the application without predefined test cases.
* The goal is to discover defects through investigation, learning, and creativity.

1. **What is traceability matrix?**

* A traceability matrix in testing is a document that maps the relationship between requirement and testcases, ensuring that all requirement are covered by test.

1. **What is Boundary value testing?**

* **Boundary Value Analysis** is a black-box test design technique in which test cases are designed based on the **boundary values** of input domains.

1. **What is Equivalence partitioning testing?**

* Aim is to treat groups of inputs as equivalent and to select one representative input to test them all
* EP can be used for all Levels of Testing

1. **What is Integration testing?**

* Integration Testing is a level of the software testing process where individual units are combined and tested as a group.

1. **What determines the level of risk?**

* Project
* Product
* Business

1. **What is Alpha testing?**

* It is always performed by the developers at the software development site.
* Sometimes it is also performed by Independent Testing Team.
* Alpha Testing is not open to the market and public.

1. **What is beta testing?**

* It is always performed by the customers at their own site. It is not performed by Independent Testing Team.

1. **What is component testing?**

* A minimal software item that can be tested in isolation. It means “A unit is the smallest testable part of software.”
* The testing of individual software components.

1. **What is functional system testing?**

* A requirement that specifies a function that a system or system component must perform.

1. **What is Non-Functional Testing?**

* Testing the attributes of a component or system that do not relate to functionality, e.g. reliability, efficiency, usability, interoperability, maintainability and portability.

1. **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.

1. **What is Adhoc testing?**

* Main aim of this testing is to find defects by random checking. Adhoc testing can be achieved with the testing technique called Error Guessing.

1. **What is load testing?**

* load testing is carried out to check systems performance at different loads i.e. number of users accessing the system

1. **What is stress Testing?**

* Stress testing - System is stressed beyond its specifications to check how and when it fails.
* Stress testing is used to test the stability & reliability of the system.

1. **What is white box testing and list the types of white box testing?**

* White Box Testing: Testing based on an analysis of the internal structure of the component or system.
* Type:

1. Unit Testing
2. Integration Testing
3. Loop Testing
4. Control Flow Testing
5. Mutation Testing
6. **What is black box testing? What are the different black box testing techniques?**

* **Black Box Testing** is a software testing method where the tester evaluates the functionality of an application without knowing its internal code, structure, or implementation details.
* **Techniques:**

1. Equivalence partitioning
2. Boundary value analysis
3. Decision tables
4. State transition testing
5. Use-case Testing
6. Other Black Box Testing

1. **Mention what are the categories of defects?**

* 1) Functionality defect

2) Critical Functionality Defect

3) Data quality/ Database defect

4) Security Defect

5) User Interface Defect

1. **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.

1. **What is the purpose of exit criteria?**

* Purpose of exit criteria is to define when we STOP testing either at the:

End of all testing – i.e. product Go Live

End of phase of testing (e.g. hand over from System Test to UAT)

1. **When should "Regression Testing" be performed?**

* 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.

1. **What is 7 key principles? Explain in detail?**

1. Testing shows presence of Defects

2. Exhaustive Testing is Impossible!

3. Early Testing

4. Defect Clustering

5. The Pesticide Paradox

6. Testing is Context Dependent

7. Absence of Errors Fallacy

1. **Difference between QA v/s QC v/s Tester**

|  |  |  |
| --- | --- | --- |
| **Testing** | **QC** | **QA** |
| Subset of QC | Quality Control | Quality Assurance |
| Focus on test execution | Focus on product | Focus on process |
| Done during development or after development | Done after development | Done before development |
| Actual Testing | Finds Defect | Prevents Defect |

1. **Difference between Smoke and Sanity?**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Purpose** | **When to test** | **What to be Tested** | **Example** |
| Smoke | To check whether build is stable or not | When we get a new build | Basic and Critical functionality | 1. App Install 2. Login 3. Dashboard 4. Logout 5. App crash or not |
| Sanity | To check to specific function is working or not | When any new feature is added or bug fix | Bug fix or related feature | Scenario : app crashes with invalid data in login feature   1. Login 2. Dashboard 3. Logout |

1. **Difference between verification and Validation**

|  |  |
| --- | --- |
| **Verification** | **Validation** |
| 1.Static Testing (“Review” of  document) | 1.Dynamic Testing  (“Live Testing” of actual software) |
| 2.Are you building the product right | 2.Are you building the right product |
| 3.Review, walkthrough, inspection | 3.Testing |

1. **What is Error, Defect, Bug and failure?**

* **Error :** A mistake made by a **developer, designer, or analyst** while writing code, creating design documents, or preparing requirements.
* **Defect :** A deviation from the **expected requirement** found in the software during testing or review.
* **Bug :** A defect that accepted by developers team or developer it is bug.
* **Failure :** When the software **behaves incorrectly during execution**, in the real environment, often because a defect escaped to production.

1. **Difference between Priority and Severity.**

|  |  |
| --- | --- |
| **Priority** | **Severity** |
| 1.Priority is of 3 types: low, medium, and high. | 1.Severity is of 5 type: critical, major, moderate, minor, and cosmetic. |
| 2.The priority level will be so different, the priority will change time to time. The defect priority won’t be in same value all the time. | 2.The severity value won’t be change time to time, it is constant value. |
| 3.Product manager decided the priorities of defects. | 3.Testing engineer decided the severity level of the defects. |
| 4.Priority status is based on customer requirements. | 4.Severity status is based on the technical aspect of the product. |

1. **What is Bug Life Cycle?**

* The bug life cycle is the process a defect goes through from the movement it’s found until it’s fixed, re-tested, and closed.

1. **Explain the difference between Functional testing and Non-Functional testing.**

|  |  |
| --- | --- |
| **Functional** | **Non-Functional** |
| 1.Verifies what the software does. It ensures that the systems features and functionality work according to the specified requirement. | 1.Evaluates how well the software performs. It assesses non-functional attribute likes performance, usability, reliability, security, and scalability. |
| 2.Functional Testing is executed first. | 2.Non-functional testing should be performed after functional testing. |
| 3.Manual testing or automation tools can be used for functional testing. | 3.Using tools will be effective for this testing. |
| 4.Business requirements are the inputs to functional testing. | 4.Performs parameters like speed, scalability, are inputs to non-functional testing. |
| 5.Easy to do manual testing. | 5.Tough to do manual testing. |
| 6.Functional testing describes what the product does. | 6.Non-functional testing describes hoe good the product works. |
| 7.Types : -unit, white box, black box, smoke, sanity. | 7.Types : - Performance, load, stress security, volume. |

1. **What is the difference between the STLC (Software Testing Life Cycle) and SDLC (Software Development Life Cycle)?**

|  |  |
| --- | --- |
| **STLC (Software Testing Life Cycle)** | **SDLC (Software Development Life Cycle)** |
| 1.STLC is mainly related to software testing. | 1.SDLC is mainly related to software development. |
| 2.It helps in making the software defects free. | 2.It helps in developing good quality software. |
| 3.STLC phases are performed after SDLC. | 3.SDLC phases are completes before the STLC. |
| 4.In STLC less number of members are needed. | 4.In SDLC more number of members are required for the whole process. |
| 5.Tester design testcases, setup the environment, workout the RTM. | 5. Developers create the actual software. |

1. **What is the difference between test scenarios, test cases, and test script?**

|  |  |  |
| --- | --- | --- |
| **Test scenarios** | **Test cases** | **Test script** |
| Is any functionality that can be tested. | Is a set of actions executed to verify particular features or functionality. | Is a set of instructions to test an app automatically. |
| Is a more focused on what to test. | Is focused on what to test and hoe to test. | Is focused on the expected result. |
| Includes an end to end functionality to be tested. | Includes test steps, data, expected result for testing. | Includes different commands to develop a script. |
| Helps test the end to end functionality in an agile way. | Helps in exhaustive testing of an app. | Helps to test specific things repeatedly. |
| Takes less time and fever resources to create. | Requires more resources and time. | Requires less time for testing but more resources for scripts creating and updating. |

1. **Explain what Test Plan is? What is the information that should be covered.**

* A test plan is a formal document that describes the strategy, objective, resources, schedule and scope of the testing process for a project.
* Covered Information is: Test plan ID, Test items, features to be tested, objectives, test strategy, test environment, entry and exit criteria, deliverables.

1. **What is priority?**

* The **impact** a defect has on the functionality of the software.

1. **What is severity?**

* The **urgency** with which a defect needs to be fixed.

1. **Bug categories are…**

* Functional Bug
* Performance Bug
* Security Bug
* Usability Bug
* Compatibility Bug
* Logic Bug
* Integration and Regression Bug

1. **Advantage of Bugzilla**.

* Free & Open Source
* Supports Multiple Projects
* Powerful Search & Filtering
* Web-Based
* Integration (Works with version control tools like Git, CVS, and SVN.)
* Customizable Workflow

1. **What are the different Methodologies in Agile Development Model?**

* Customer satisfaction through early and continuous software delivery
* Accommodate changing requirements throughout the development process
* Frequent delivery of working software
* Collaboration between the business stakeholders and developers throughout the project
* Support, trust, and motivate the people involved
* Enable face-to-face co-located interactions
* Working software is the primary measure of progress
* Agile processes to support a consistent development pace
* Attention to technical detail and design enhances agility
* Simplicity
* Self-organizing teams encourage great architectures, requirements, and designs
* Regular reflections on how to become more effective

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

* Authentication: Accepting an invalid username/password
* Authorization: Accessibility to pages though permission not given User Interface Defects: As the name suggests, the bugs deal with problems related to UI are usually considered less severe.

1. **When to used Usability Testing?**

* **Usability Testing** is used **any time** you want to verify that real users can easily and efficiently complete tasks on your product — ideally before full release and after any major change.

1. **What is the procedure for GUI Testing?**

* Check all the GUI elements for size, position, width, length and acceptance of characters or numbers. For instance, you must be able to provide inputs to the input fields.
* Check you can execute the intended functionality of the application using the GUI Check Error Messages are displayed correctly.
* Check for Clear demarcation of different sections on screen.
* Check Font used in application is readable.
* Check the alignment of the text is proper.
* Check the Color of the font and warning messages is aesthetically pleasing.
* Check that the images have good clarity.
* Check that the images are properly aligned.
* Check the positioning of GUI elements for different screen resolution.