1. **What is Exploratory Testing?**

Ans:-

Exploratory Testing is a way of testing software without following a fixed set of test cases. Instead, the tester uses their own ideas and experience to check how the software works.

1. **What is traceability matrix?**

Ans:-

A Traceability Matrix is a document that helps you track and match requirements with test cases to make sure everything is tested.

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

Ans:-

**Boundary Value Analysis (BVA)** is a software testing method where **test cases** are designed to check inputs at the edges of the allowed range (minimum, maximum, and just outside them) because errors often occur at these points.

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

Ans:-

Equivalence partitioning is a software testing technique that divides the input data of a software unit into partitions or classes of data from which test cases can be derived.

1. **What is Integration testing?**

Ans:-

Integration Testing is a software testing method where test cases are designed to check whether different modules or components of a system work together correctly after being combined. It helps find problems in the interaction between modules.

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

Ans:-

The level of risk in software testing is determined by two main things:

Likelihood (Probability) of failure – How likely it is that something will go wrong.

Impact (Severity) of failure – How serious the effect will be if it does go wrong.

1. **What is Alpha testing?**

Ans:-

Alpha Testing is a type of software testing done before the product is released to real users.

It is performed by internal staff (developers, testers, or a selected group inside the company) to find bugs and issues early.

1. **What is beta testing?**

Ans:-

Beta Testing is a type of software testing where the product is given to real users outside the company to use in a real-world environment before the official release.

1. **What is component testing?**

Ans:-

Component Testing is a type of software testing where individual components (modules) of a software are tested in isolation to check if they work correctly.

E.G: If you are building an online shopping app, testing only the login module to see if it accepts correct usernames/passwords and rejects wrong ones — before connecting it to search, cart, or payment — is Component Testing.

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

Ans:-

Functional Testing: Testing based on an analysis of the specification of the functionality of a component or system.

where the entire system is tested against its functional requirements to ensure it works as expected.

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

Ans:-

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?**

Ans:-

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?**Ans:-

Adhoc Testing is a random, unplanned testing approach where testers check the application in an informal way, often focusing on areas they suspect may have issues, to uncover hidden defects.

1. **What is load testing?**Ans:-

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.

1. **What is stress Testing?**

Ans:-

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.

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

Ans:-

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.

**Types of white box testing: -**

Unit Testing

Integration Testing

Loop Testing

Mutation Testing

Condition Testing

Path Testing

1. **What is black box testing? What are the different black box testing techniques?**Ans:- Black-box testing: Testing, either functional or non-functional, without reference to the internal structure of the component or system.

**Black Box testing techniques: -**

* Equivalence partitioning
* Boundary value analysis Decision tables
* State transition testing Use-case Testing
* Other Black Box Testing.

1. **Mention what are the categories of defects?**Ans**:-**

* Functional Defects
* Performance Defects
* Usability Defects
* Compatibility Defects
* Security Defects
* Logical Defects
* Integration Defects

1. **Mention what big bang testing is?**

Ans:-

Big Bang Testing is an integration testing approach where all modules of the system are combined together at once and then tested as a whole, instead of testing them in phases.

1. **What is the purpose of exit criteria?**Ans**:-**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?**Ans:-

* **After fixing defects** – To ensure the fix didn’t break other parts of the system.
* **After adding new features** – To check that existing features still work as expected.
* **After code changes or enhancements** – To verify that updates didn’t cause unintended issues.
* **After performance improvements or** **optimizations** – To confirm no functionality was affected.
* **After environment changes** – Such as OS upgrades, browser updates, or database changes.

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

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**

|  |  |  |  |
| --- | --- | --- | --- |
| **Aspect** | **QA (Quality Assurance)** | **QC (Quality Control)** | **Tester** |
| **Focus** | Process-oriented | Product-oriented | Person doing testing |
| **Goal** | Prevent defects | Detect defects | Execute tests & find defects |
| **Activity Type** | Proactive (before testing) | Reactive (after development) | Operational (hands-on testing) |
| **Example** | Defining standards, creating checklists, process audits | Verifying product, running test cycles | Running functional tests, reporting bugs |
| **Responsibility** | Ensuring correct process is followed | Ensuring the product meets requirements | Performing actual testing work |
|  |  |  |  |

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

Ans:-

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | **Aspect** | **Smoke Testing** | **Sanity Testing** | | **Purpose** | To verify that the basic functionalities work and the build is stable enough for further testing. | To verify that specific bugs have been fixed and the related functionality works. | | **Level** | High-level testing | Narrow and deep testing | | **Scope** | Broad – covers all major functions | Narrow – focuses on specific areas | | **Performed by** | QA team | QA team | | **When performed** | After receiving a new build | After receiving a bug fix or minor change | | **Automated or Manual** | Often automated | Usually manual (can be automated if needed) | | **Time Required** | Short – quick checks | Short – but slightly more focused than smoke tests | | **Objective** | To accept or reject a build for further testing | To verify the correctness of a specific functionality | | **Example** | Launching the app, logging in, opening main screens | Checking if the login issue is fixed and works now | | **Build Acceptance** | Determines whether the build is stable enough for testing | Determines whether a specific function works after fixes | |  |  |

1. **Difference between verification and Validation**

Ans:-

|  |  |  |
| --- | --- | --- |
| **Aspect** | **Verification** | **Validation** |
| **Definition** | Verification ensures the product is being built according to the requirements. | Validation checks if the final product meets the user’s needs and expectations. |
| **Process** | It involves reviewing documents, designs, and code without executing the program. | It involves actual testing by running the software to ensure correct behavior. |
| **Focus** | Focuses on "Are we building the product, right?" | Focuses on "Are we building the right product?" |
| **Timing** | Performed during the development phase, before testing begins. | Performed after verification, usually at the end of development. |

1. **Explain types of Performance testing**

Ans:-

**1. Load Testing**

* **Purpose**: To check how the system behaves under expected user load.

**2. Stress Testing**

* **Purpose**: To test the system under extreme loads to determine the breaking point and how it recovers.

**3. Spike Testing**

* **Purpose**: To test the system’s reaction to sudden large spikes in user load.

**4. Endurance Testing (Soak Testing)**

* **Purpose**: To verify the system can handle the expected load over an extended period.

**5. Scalability Testing**

* **Purpose**: To determine how well the system scales up or down in response to changes in load.

**6. Volume Testing (Flood Testing)**

* **Purpose**: To test the system’s ability to handle a large volume of data.

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

Ans:-

A mistake in coding is called error, error found by tester is called defect, defect accepted by development team then it is called bug, build does not meet the requirements then it is failure.

**Error:** A human mistake in coding or design.

**Defect:** A flaw in the software caused by an error.

**Bug:** Another name for a defect.

**Failure:** The system's inability to perform as expected.

1. **Difference between Priority and Severity**

Ans:-

|  |  |  |
| --- | --- | --- |
| **Aspect** | **Priority** | **Severity** |
| **Meaning** | How soon the bug should be fixed | How serious the bug is |
| **Focus** | Business/Project urgency | Technical impact |
| **Decided by** | Project Manager / Business Analyst / QA Lead | QA / Tester |
| **Fix urgency** | High Priority = Fix it quickly | High Severity = Major system failure |
| **Example** | A typo on the homepage – high priority, low severity | System crash when clicking “Submit” – low priority, high severity |
| **Types** | Low, Medium, High, Urgent | Minor, Major, Critical, Blocker |

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

Ans:-

A **Bug Life Cycle** is the journey of a bug from the time it is found until it is fixed and closed, showing all the stages it passes through.

1. **Explain the difference between Functional testing and non-functional testing**

Ans:-

|  |
| --- |
|  |

|  |  |  |
| --- | --- | --- |
| **Aspect** | **Functional Testing** | **Non-Functional Testing** |
| **Focus** | What the system does | How the system works |
| **Checks** | Features, functions, and outputs | Performance, speed, security, usability |
| **Example** | Login, add to cart, submit form | Load time, stress test, data security |
| **Answer** | “Does it work correctly?” | “Does it work well?” |

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

Ans:-

|  |  |  |
| --- | --- | --- |
| **Aspect** | **SDLC (Software Development Life Cycle)** | **STLC (Software Testing Life Cycle)** |
| **Meaning** | |  | | --- | | Process of developing software |  |  | | --- | |  | | |  | | --- | | Process of testing software |  |  | | --- | |  | |
| **Focus** | |  | | --- | | Building the product |  |  | | --- | |  | | |  | | --- | | Ensuring product quality |  |  | | --- | |  | |
| |  | | --- | | **Phases** |  |  | | --- | |  | | |  | | --- | | Planning → Design → Coding → Testing → Deployment → Maintenance |  |  | | --- | |  | | |  | | --- | | Requirement Analysis → Test Planning → Test Design → Test Execution → Test Closure |  |  | | --- | |  | |
| |  | | --- | | **Done by** |  |  | | --- | |  | | |  | | --- | | Developers + Project Team |  |  | | --- | |  | | |  | | --- | | Testers / QA Team |  |  | | --- | |  | |
| |  | | --- | | **Output** |  |  | | --- | |  | | |  | | --- | | Working software/application |  |  | | --- | |  | | Tested and quality-assured software |

1. **What isthe difference between test scenarios, test cases, and test script?**Ans:-

|  |  |  |
| --- | --- | --- |
| **Term** | **Meaning** | **Example** |
| |  | | --- | | **Test Scenario** |  |  | | --- | |  | | |  | | --- | | High-level idea of what to test (covers one feature or functionality). |  |  | | --- | |  | | |  | | --- | | *“Verify login functionality.”* |  |  | | --- | |  | |
| |  | | --- | | **Test Case** |  |  | | --- | |  | | |  | | --- | | Step-by-step conditions, inputs, and expected results to validate a scenario. |  |  | | --- | |  | | |  | | --- | | *Enter valid username & password → Click login → Expected: Dashboard opens.* |  |  | | --- | |  | |
| |  | | --- | | **Test Script** |  |  | | --- | |  | | |  | | --- | | A set of instructions (manual or automated code) executed to test the functionality. |  |  | | --- | |  | | Automated Selenium script that enters login details and checks dashboard. |

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

Ans;-

A Test Plan is a document that explains how testing will be done for a software project.

It is like a roadmap for testing.

**It covered:-**

* **Scope** – What will be tested and not tested
* **Objectives** – Purpose of testing
* **Test Strategy** – How testing will be done (types of testing)
* **Resources** – People, tools, and environment needed
* **Schedule** – When testing will be done
* **Deliverables** – Test cases, reports, results
* **Entry & Exit Criteria** – When to start and stop testing
* **Risks** – Possible problems and solutions

1. **What is priority?**

Ans:-

Priority means how urgently a bug should be fixed.

It tells the order of fixing based on business needs or user impact.

High Priority → Must be fixed immediately (e.g., Login not working).

Medium Priority → Should be fixed soon, but not blocking major functions.

Low Priority → Can be fixed later, not urgent (e.g., minor spelling mistake).

1. **What is severity?**

Ans:-

Severity means how serious the impact of a defect is on the system.

It shows the effect on functionality or performance if the bug is not fixed.

High Severity → Major function is broken (e.g., payment not processed).

Medium Severity → Some functionality is affected but workaround exists.

Low Severity → Very minor issue (e.g., spelling error, UI alignment).

1. **Bug categories are…**

Ans:-

Functional Bugs

Performance Bugs

UI/Usability Bugs

Compatibility Bugs

Security Bugs

Integration Bugs

Data/Database Bugs

1. **Advantage of Bugzila .**

Ans:-

* Free and open-source bug tracking tool
* Web-based, easy to access and use
* Provides detailed bug reports & notifications
* Supports multiple projects with customizable workflow

1. **Difference between priority and severity**Ans:-

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| | **Aspect** |  |  |  | **Priority (Urgency)** | **Severity (Impact)** | | --- | --- | --- | --- | --- | --- | |  |  |  |  |  |  | | Meaning |  |  |  | How quickly the bug should be fixed | How badly the bug affects the system | | Focus |  |  |  | Business needs / Urgency | Technical impact / Functionality | | Set By |  |  |  | Product Manager / Business team | Tester / QA team | | Example |  |  |  | Spelling mistake on homepage = High Priority, Low Severity | Application crash = High Severity, Low Priority | |

1. **What are the different Methodologiesin Agile Development Model?**Ans:-

- Customer satisfaction through early and continuous software delivery.

- Accommodate changing requirement through the developments process.

- Frequent delivery of working software.

- Collaboration between the business stakeholders and developers through the project.

- Support, trust, and motivate the people involved.

- Enable face to face interactions.

- Working software is the primary measure of progress.

- Agile process to supports a consistent development race.

- Attention to technical and design enhances agility.

- Simplicity.

- Self- organizing teams encourage great architectures, requirements and designs.

- Regular reflection 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?**

Ans:-

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| | **Aspect** |  | **Authentication (Who you are)** | **Authorization (What you can do)** | | --- | --- | --- | --- | |  |  |  |  | | **Meaning** |  | Verifying user’s identity (are you really the user?) | Checking what actions/permissions the user has | | **Purpose** |  | Confirms *who* is logging in | Controls *what* the user can access | | **Example** |  | Entering username & password | Admin can access dashboard, normal user cannot | |

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

Ans:-

Usability Testing is done to ensure the product is simple, clear, and easy for end users. It is used before launch, after adding new features, or whenever user-friendliness needs to be checked.

1. **What is the procedure for GUI Testing?**Ans:-

* 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.