**What is Exploratory Testing?**

**Ans: -** Exploratory Testing is a type of software testing where Test cases are not created in advance but testers check system on the fly. They may note down ideas about what to test before test execution. The focus of exploratory testing is more on testing as a “thinking” activity.

**What is traceability matrix?**

**Ans: -** The full form of RTM is Requirement traceability Metrix

2.RTM is a document that trace and maps the requirements ids with test case ids

3. the purpose of RTM is to confirm that all the requirements have covered along with test case.

4. RTM document is prepared to show the clients that coverage is complete with requirement and testing.

**What is Boundary value testing?**

**Ans: -**Boundary Value Testing (BVT) is a software testing technique used to identify errors at the edges of input ranges rather than within the range itself. It is based on the principle that errors often occur at the boundaries of input domains.

**What is Equivalence partitioning testing?**

**Ans: -** Equivalence Partitioning (EP) is a black-box testing technique used to divide input data into groups (partitions) that are expected to exhibit similar behaviour. Instead of testing every possible input, testers pick one representative value from each partition.

**What is Integration testing?**

**Ans: -** Integration Testing is a software testing technique where individual modules or components of an application are combined and tested as a group. The goal is to verify how well the different parts of the system work together.

**What determines the level of risk?**

Ana: - The level of risk in software testing is determined by the likelihood of a defect occurring and the impact it would have on the system or users.

**What is Apha testing?**

**Ans: -** 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.

**What is Beta testing?**

**Ans:** Beta Testing is performed by real users of the software application in a real environment. Beta testing is one type of User Acceptance Testing. A pre-release version of the product is made available for testing to a chosen set of external users or customers during the second phase of software testing.

**What is component testing?**

Ans: - Component Testing, also known as Module Testing, is a type of testing where individual components or modules of a software application are tested in isolation. The goal is to verify that each component functions correctly as per the requirements before integrating it with other components.

**What is functional system testing?**

Ans: - Functional System Testing is a type of black-box testing that verifies the entire system’s functionality based on the specified requirements. It ensures that all components work together correctly when integrated into a complete system.

**What is Non-Functional Testing?**

Ans: - Non-functional testing evaluates the performance, security, usability, and reliability of a software application. It focuses on how well the system works rather than what it does.

**What is GUI Testing?**

**Ans: -** GUI Testing (Graphical User Interface Testing) is a type of software testing that focuses on evaluating a software application's graphical user interface. The goal is to ensure that the interface meets the expected design, functions correctly, and provides a good user experience.

**What is Adhoc testing?**

**Ans: -** Adhoc testing is an informal and unstructured software testing approach where testers randomly explore the application without following predefined test cases or plans. It aims to find defects that might be missed in structured testing by relying on the tester’s intuition, experience, and creativity. Adhoc testing three type which is Buddy Testing, Monkey Testing, and Pair Testing.

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

**Ans: -** White box testing is a software testing technique that examines the internal structure, logic, and code of an application. Testers have complete knowledge of the source code and test it to ensure proper implementation and functionality.

**It has three types**

1.statement coverage

2.Branch coverage

3.Decision coverage

**What is black box testing? What is the different black box testing techniques?**

Ans: - 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. The tester provides input, observes the output, and checks whether the system behaves as expected.

**Different black box testing techniques**

1️. Equivalence Partitioning (EP)

2️. Boundary Value Analysis (BVA)

3️. Decision Table Testing

4. State Transition Testing

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

**Ans: -** there are many types of defects such as :

1.Data quality/Database Defects

2.critcal Functionality Defects

3.Functionaliy Defects

4.Securuty Defects

5.User Interface Defects

**Mention what big bang testing is?**

**Ans: -** Big Bang Testing is an integration testing approach where all modules or components of a software system are combined and tested as a whole, rather than integrating and testing them incrementally.

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

**Ans:** - Exit criteria are the conditions that must be met before moving to the next step in a process. They help ensure that everything is properly completed, reducing mistakes and risks.

**When should “Regression Testing” be performed?**

**Ans: -** Regression Testing should be performed in the following situations:

1.After a code Fix: - When a defect is fixed, regression testing ensures that the fix did not introduce new issues in other parts of the application.

2. After a New Feature is Added: - Whenever a new feature or functionality is added, regression testing ensures that existing functionalities still work properly.

3. After Performance Enhancements or Code Optimization: - If the development team optimizes the code or improves performance, regression testing checks that the changes did not break any existing functionality.

4.After Configuration or Environment Changes: - If there is a change in database versions, server configurations, or third-party integrations, regression testing ensures the software works as expected.

5. During Continuous Integration & Continuous Deployment (CI/CD): - Automated regression testing is often performed in CI/CD pipelines to ensure stability before deployment.

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

**Ans: -** 1. Testing shows presence of defect

2. Exhaustive testing is impossible

3. Early testing

4. Defect clustering

5. Pesticide paradox

6. Testing is context dependent

7. Absence of error fallacy

Explain in brief under: -

1. **Testing shows the presence of defects**: - testing shows the presence of defects but cannot prove that this product is 100 percent defect free. Testing shows the defects are precented.
2. **Exhaustive testing is impossible**: - that means you cannot test all condition and all scenarios they have many results so this is time consuming, risk, and will make costly as well. For example, one billing software calculates taxes on final price than you cannot check as on 500rs what amount tax deducted after same on 100,5000,2500 etc.
3. **Early testing:** - that means you should test early as possible from reporting analysis face so you can reduce number of defects from the beginning.
4. **Defect clustering: -** that means you found 2 or 3 defects in a single module so maximum chance to found most the defects in those module defects are not clustered in hole system. In a simple word we can say that there are 80 present defects will find in a 20 percent module. Defects are not spread in a system. They are clustered.
5. **The pesticide paradox**: -that means once a tester finds bugs on solve them after that minimum chance to find any bug so than you should revise your test document and then test on different module so you can find bug. For example, WhatsApp version 2 test document cannot work in version4
6. **Testing is context dependant:** - That means not all testing approaches work in each project. Some will in one environment, while others in other environments. For example, in gaming software tester check how much load on the ram, how much ram required for run this software etc. while in billing software tester check usability, convenience, bill processing load etc.
7. **Absence of error fallacy: -** That means in a software you do not find any error than you made some mistake so you should check the all-requirement document so you can find defect.

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

|  |  |  |
| --- | --- | --- |
| QA | QC | Tester |
| Preventing defects before they occur | Identifying and correcting defects in the final product. | Executing tests to find and report bugs |
| Reviewing and approving the improvement | Reporting and tracking defects. | Designing and executing test cases. |
| Process-oriented activities | Product-oriented activities | Product-oriented activities |
| It is a subset of SDLC | It is a subset of quality assurance | It is a subset of quality assurance |

**Difference between Smoke and Sanity?**

**Ans: -**

|  |  |  |
| --- | --- | --- |
| **Smoke Testing V/S Sanity Testing** | | |
| **Sr No.** | **Smoke Testing** | **Sanity Testing** |
| 1 | Ensures that the basic and critical functionalities of the application work. | Ensures that newly added or fixed features work correctly. |
| 2 | Conducted on a new build before moving to detailed testing. | Conducted after receiving a minor bug fix or enhancement in an existing build. |
| 3 | Covers all major functionalities at a high level. | Focuses only on the changed or affected areas. |
| 4 | Short and quick, usually a few minutes to an hour. | Slightly longer, but still quicker than regression testing. |
| 5 | Uses broad test cases covering major flows. | Uses deep, focused test cases for specific functions. |
| 6 | Smoke Testing like general Health Checkup. | Sanity Testing is like Specialized health checkup. |

**Difference between verification and Validation**

|  |  |  |
| --- | --- | --- |
| **Verification and Validation** | | |
| **Sr No.** | **Verification** | **Validation** |
| 1 | Verification Testing is a review activity | Validation testing is a test execution activity. |
| 2 | Verification Testing involves static testing technique | Validation involves dynamic testing techniques |
| 3 | Verification testing checks the document like: Plans, Requirement specification, Design specification, code, test case etc. | validation testing checks the actual product |
| 4 | QA team does verification testing | Testing team does Validation testing |
| 5 | Verification process come before validation testing | validation process come after verification testing |
| 6 | Here question is 'are you building it Right? | Here question is 'Have you build the right thing? |

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

**Ans: -** when developer makes mistake while coding it called error, then tester found this them it is defect, after developer accept it is a bug, defect are not fixed by developer it is failure.

**Difference between Priority and Severity**

**Ans: -**

|  |  |  |
| --- | --- | --- |
| **Priority and Severity** | | |
| **Sr No.** | **Priority** | **Severity** |
| 1 | Priority defines **how soon** a defect should be fixed. | Severity defines **how serious** the defect is in terms of system impact. |
| 2 | Focus on Business needs and urgency. | Focus on Technical impact on the system. |
| 3 | Usually set by the **Project Manager, Client, or Business Team**. | Usually set by the **Tester or Developer**. |
| 4 | Types: High, Medium and low | Types: Critical, Major (High), Moderate(medium), Minor (Low), Cosmetic. |

**What is Bug Life Cycle?**

**Ans: -** The Bug Life Cycle defines the various stages a defect goes through from identification to closure. It ensures that defects are tracked, managed, and resolved systematically.

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

**Ans: -**

|  |  |  |
| --- | --- | --- |
| **Functional testing V/s Non-Functional** | | |
| **Sr No.** | **Functional testing** | **Non-Functional** |
| 1 | Tests whether the application works as expected based on requirements. | Tests how well the application performs under various conditions. |
| 2 | Validates features, user interactions, and outputs. | Evaluates performance, security, usability, and other system attributes. |
| 3 | Ensures that the system meets functional requirements. | Ensures system efficiency, reliability, and user experience. |
| 4 | Login functionality, payment processing, form submission, navigation. | Load time, response speed, security vulnerabilities, accessibility. |
| 5 | Unit testing, integration testing, system testing, UAT (User Acceptance Testing). | Performance testing, security testing, usability testing, reliability testing. |
| 6 | Can be manual or automated using tools like Selenium, TestNG. | Often automated using tools like JMeter, LoadRunner, or Burp Suite. |
|  |  |  |
| 7 | Checking if a car's brakes work properly. | Checking how smoothly the car drives, fuel efficiency, and crash safety. |

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

**Ans: -**

|  |  |  |
| --- | --- | --- |
| **SDLC and STLC** | | |
| **Sr No.** | **SDLC** | **STLC** |
| 1 | SDLC is a development life cycle | STLC is a testing life cycle. |
| 2 | The main objective of the SDLC is to complete successful development of the software including testing and other phases. | The main objective of STLC is only Testing. |
| 3 | In SDLC, the business analyst gathers the requirements and creates a development plan. | In STLC, QA team analyse requirement documents and create a system test plan. |
| 4 | In SDLC, the real code is developed, and actual work takes place as per the design documents. | The testing team prepares the test case and executes them. |
| 5 | SDLC involved Requirement gathering, Design, Coding, Deployment and maintenance. | STLC is a part of SDLC, so it involves only the testing part like Requirement Analysis, Test Planning, Test Development, Test Environment Setup, Test Execution& Closure. |

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

**ANS: -**

|  |  |  |  |
| --- | --- | --- | --- |
| **Aspect** | **Test Scenario** | **Test Case** | **Test Script** |
| Definition | A high-level description of what needs to be tested. | A detailed document that defines the steps, input data, and expected output for a specific test. | A set of automated instructions written in a programming/scripting language to execute a test. |
| Purpose | Ensures all features and functionalities are covered. | Provides a clear step-by-step testing approach. | Automates repetitive testing tasks. |
| Format | Written in simple statements. | Structured format with steps, inputs, and expected results. | Code or script in a testing tool (e.g., Selenium, JUnit). |
| Who Writes It? | Testers or Business Analysts. | Testers or QA Engineers. | Automation Test Engineers. |
| Example | **Scenario:** Test the login functionality of an application. | **Test Case:** 1. Open the login page.  2. Enter a valid username and password.  3. Click the "Login" button.  4. Verify the user is redirected to the homepage. | **Test Script:** A Selenium WebDriver script automating the login process using Python. |

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

**Ans: -** A Test Plan is a document that outlines the testing strategy, objectives, schedule, resources, and scope of a software testing process. It serves as a blueprint for the testing team to ensure quality and completeness in testing.it is covered the information like Test plan ID, Introduction, Scope of Testing, Testing Objectives, Test Approach, Test Environment, Test Deliverables, Test schedule, Roles & Responsibilities, Defect Menagement Process, Risk & Mitigation, Entry & Exit Criteria.

**What is priority?**

**Ans: -** Priority in software testing defines how urgently a defect should be fixed based on its impact on business requirements. It is assigned by the Project Manager, Client, or Business Analyst and helps developers decide the order in which defects should be resolved.

**What is severity?**

**Ans: -** Severity defines the impact of a defect on the functionality or performance of the software. It is set by Testers or QA Engineers and helps determine how serious the defect is from a technical perspective.

**Bug categories are…**

**Ans: -** categories of Bug areData base bug, Critical functionality bug, Functionality bug, Security Bug, User interface bug.

**Advantages of Bugzilla.**

**Ans: -** Bugzilla is a popular open-source bug tracking system used for defect management in software projects. It helps teams log, track, and resolve bugs efficiently.

**Advantages:**

**1.Free & Open Source: -**No licensing cost, making it a budget-friendly option.

**2. Easy Bug Tracking & Management: -** Allows testers to log, track, and update defects efficiently. Developers can assign bug statuses (New, Assigned, Resolved, etc.).

**3. Email Notifications & Alerts: -** Sends automatic emails when bugs are updated, ensuring better team communication.

**4. Powerful Search & Filtering: -** Users can search, filter, and categorize bugs based on priority, severity, assignee, etc.

**5. Customizable Workflow: -** Can be tailored to match different project needs and development workflows.

**6. Integration with Other Tools: -**Supports integration with version control systems like Git, SVN, and CI/CD tools.

**7. User Access Control & Security: -** Allows role-based access (Tester, Developer, Manager) for better control.

**8. Reports & Charts for Analysis: -** Provides bug trends, graphs, and statistics to monitor project quality.

**9. Multi-Platform & Browser Support: -** Works on Windows, Linux, macOS, and supports all major browsers.

**10. Scalability: -** Can handle large projects with thousands of bugs efficiently.

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

**Ans: -** There are two types of methodologies in Agile Development Model

1.Scum and 2. Kanban

**Explain the difference between Authorization and Authentication in Web testing.**

|  |  |  |
| --- | --- | --- |
| **Authentication and Authorization** | | |
| **Sr No.** | **Authentication** | **Authorization** |
| 1 | Authentication Verifies **who** the user is. | Authorization Determines **what** the user can access. |
| 2 | Purpose of authentication is to Ensures the user is genuine. | Purpose of Authorization is to Ensures the user has proper permissions. |
| 3 | Process of authentication is Requires username, password, OTP, biometrics. | Process of authorization is based on roles, permissions, access control. |
| 4 | Example of authentication: Logging in with a username & password. | Example of authorization: Admin can access settings, but a regular user cannot. |
| 5 | Security Issues: Weak passwords, brute-force attacks. | Security Issues: Improper role-based access, privilege escalation. |

**What are the common problems faced in Web testing?**

**Ans: -** Web applications are complex, and testers often encounter various issues during testing. Below are the most common problems faced in Web Testing:

1.Functional Issues: - Website features not working as expected.

2.Broken Links: - Some links lead to 404 errors (Page Not Found).

3. Cross-Browser Compatibility: - Website looks or behaves differently on different browsers

4. Performance Issues: - Slow page load time, high server response time, and crashes under heavy traffic.

5. Security Vulnerabilities: - Website is vulnerable to hacking, data leaks, or unauthorized access.

6. Responsive Design Issues: - Website does not adjust properly on different screen sizes (Mobile, Tablet, Desktop).

7. Session Management Bugs: - Session timeout does not work, or the user remains logged in after logout.

8. Incorrect Data Handling: - Forms accept invalid input, causing database errors.

9. UI/UX Issues: - Overlapping text, broken images, unreadable fonts.

10. Integration Issues: - Problems with third-party APIs, payment gateways, or database connections.

**When to used Usability Testing?**

**Ans:-1.** Before Product Launch.

2. After Major UI/UX Updates

3. When There Are User Complaints

4. During Competitive Analysis

5. For Accessibility Testing

6. When Optimizing Conversion Rates

7. For Mobile Responsiveness & Cross-Platform Usability

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

**Ans: -**1. Requirement Analysis

2. Prepare the Test Environment

3. Design GUI Test Cases cover Visual testing, Functionality testing, usability testing, compatibility testing.

4. Execute GUI Test Cases

5. Report and Fix Issues

6. Perform Regression Testing

7. Final Validation