Software Testing Assignment

Q-What is Exploratory Testing?

A-Exploratory testing is a type of software testing approach that emphasizes on the simultaneous learning, test design, and test execution. It is an unscripted approach that allows testers to explore the software system under test and identify defects that may not be easily identified using scripted testing.

Q-What is traceability matrix?

A- A traceability matrix is a document that helps to track and manage the relationship between different requirements, test cases, and defects. It ensures that all requirements are covered by test cases, and all defects can be traced back to the corresponding requirement and test case.

Q-What is Boundary value testing?

A-Boundary value testing is a type of software testing technique in which test cases are designed to evaluate the behavior of a system at the boundary limits of its input domain. It helps to identify defects that may occur at the boundaries of input ranges.

Q-What is Equivalence partitioning testing?

A-Equivalence partitioning testing is a type of software testing technique in which the input domain of a system is partitioned into different equivalence classes based on the characteristics of the input. It helps to reduce the number of test cases required to test a system while ensuring that all possible input conditions are covered.

Q-What is Integration testing?

A-Integration testing is a type of software testing technique in which individual modules or components of a system are combined and tested as a group to ensure that they function together correctly. It helps to identify defects that may occur due to the interaction between different modules.

Q-What determines the level of risk?

A-The level of risk in software development is determined by the likelihood of defects occurring and the impact of those defects on the system and its users. The level of risk can be determined by analyzing factors such as the complexity of the system, the size of the user base, and the criticality of the system.

Q-What is Alpha testing?

A-Alpha testing is a type of software testing technique in which a group of users, typically developers, test a system before it is released to the public. It helps to identify defects and usability issues that may not be easily identified using other testing techniques.

Q-What is beta testing?

A-Beta testing is a type of software testing technique in which a group of end-users test a system in a real-world environment before its official release. It helps to identify defects and usability issues that may not be easily identified using other testing techniques.

Q-What is component testing?

A-Component testing is a type of software testing technique in which individual modules or components of a system are tested in isolation from the rest of the system. It helps to ensure that each component functions correctly before it is integrated into the larger system.

Q-What is functional system testing?

A-Functional system testing is a type of software testing technique in which the functionality of a system is tested against its specified requirements. It helps to ensure that the system meets the functional requirements specified for it.

Q-What is Non-Functional Testing?

A-Non-Functional Testing is a type of software testing technique in which the non-functional aspects of a system, such as performance, scalability, and security, are tested to ensure that they meet the specified requirements.

Q-What is GUI Testing?

A-GUI testing is a type of software testing technique in which the graphical user interface of a system is tested to ensure that it is user-friendly and functions correctly.

Q-What is Adhoc testing?

A-Adhoc testing is a type of software testing technique in which testers perform testing without a formal test plan or predefined test cases. It helps to identify defects that may not be easily identified using scripted testing.

Q-What is load testing?

A-Load testing is a type of software testing that tests the performance of a system under a heavy load or stress. It helps identify any performance bottlenecks in the system

Q-What is stress testing?

A-Stress testing is a type of testing that evaluates the behavior of a system or application under high load conditions. The purpose of stress testing is to identify the system's breaking point and determine whether it can handle heavy traffic, high data volumes, and other stress factors.

Q-What is white box testing and list the types of white box testing?

A-White box testing, also known as structural testing or glass box testing, is a testing technique that examines the internal workings of an application's code. The types of white box testing include:

Unit testing

Integration testing

System testing

Regression testing

Q-What is black box testing?

A-What are the different black box testing techniques? Black box testing is a testing technique that evaluates the functionality of an application without examining its internal workings. The different black box testing techniques include:

Functional testing

Non-functional testing

Regression testing

Smoke testing

Compatibility testing

Acceptance testing

Q-Mention what are the categories of defects? The categories of defects include:

A-Functional defects

Performance defects

GUI defects

Compatibility defects

Security defects

Q-Mention what big bang testing is?

A-Big bang testing is a testing approach where all the testing activities are performed at once, without any specific plan or strategy. This approach is usually used for small projects where testing activities are minimal.

Q-What is the purpose of exit criteria?

A-The purpose of exit criteria is to define the conditions that must be met before a testing phase can be considered complete and the software can be released. These criteria are defined to ensure that the software meets the required quality standards and that all test objectives have been achieved.

Q-When should "Regression Testing" be performed?

A-Regression testing should be performed whenever changes are made to the software, such as adding new features or fixing defects. Its purpose is to ensure that the changes made to the software have not introduced new defects or affected existing functionality.

Q-What is 7 key principles? Explain in detail? The 7 key principles of software testing are:

A-Testing shows the presence of defects

Exhaustive testing is impossible

Early testing saves time and money

Defect clustering

Pesticide paradox

Testing is context-dependent

Absence-of-errors fallacy

Q-Difference between QA v/s QC v/s Tester

A-QA (Quality Assurance) is a process-oriented approach to ensure the quality of a software product. QC (Quality Control) is a product-oriented approach that focuses on detecting and fixing defects. A tester is responsible for executing test cases and identifying defects in the software.

Q-Difference between Smoke and Sanity?

A-Smoke testing is a type of testing that ensures that the critical functionalities of the software are working as expected after a build or release. Sanity testing is a type of testing that ensures that the newly added functionality or bug fixes are working as expected.

Q-Difference between verification and Validation

A-Verification is the process of ensuring that the software meets the specified requirements and standards. Validation is the process of ensuring that the software meets the user's needs and expectations.

Q-Explain types of Performance testing.

A-The types of performance testing include:

Load testing

Stress testing

Endurance testing

Spike testing

Volume testing

Scalability testing

Q-What is Error, Defect, Bug and failure?

A-An error is a human action that produces an incorrect result. A defect is a deviation from the expected behavior of the software. A bug is a defect that causes the software to malfunction. A failure is the inability of the software to perform its intended function

Q-Difference between Priority and Severity?

A-Priority and Severity are two important attributes used in bug tracking and management.

Priority refers to the level of urgency or importance that should be given to a defect. It determines how quickly the defect needs to be fixed relative to other defects. Priority can be set as High, Medium, or Low.

Severity, on the other hand, is the degree of impact that a defect has on the system or application under test. It determines how critical the defect is in terms of its effect on the system's functionality or user experience. Severity can be classified as Critical, Major, Minor, or Cosmetic.

In short, Priority is about when a defect needs to be fixed, and Severity is about how important it is to fix the defect.

Q-WHAT IS Bug Life Cycle?

A-The Bug Life Cycle is the journey of a bug or defect from its discovery to its resolution. It involves a set of stages or states that a bug goes through before it is considered closed or resolved. The typical Bug Life Cycle stages include:

New: The bug is identified and reported by the tester.

Assigned: The bug is assigned to a developer for further analysis and fixing.

Open: The developer is working on fixing the bug.

Fixed: The developer has fixed the bug, and it is waiting for verification.

Verified: The tester verifies the fix and confirms that the bug is resolved.

Closed: The bug is confirmed to be resolved and closed.

The Bug Life Cycle helps to manage the process of identifying, tracking, and resolving bugs in a systematic way.

Q-Difference between Functional testing and Non-Functional testing?

A-Functional testing and Non-Functional testing are two types of software testing.

Functional testing is the process of testing the system or application against its functional requirements. It checks whether the system behaves as expected and meets the user's needs. Functional testing focuses on testing the software's features, functionalities, and business logic. It includes testing methods such as unit testing, integration testing, system testing, and acceptance testing.

Non-Functional testing, also known as performance testing, is the process of testing the system's non-functional requirements such as scalability, reliability, performance, security, and usability. Non-functional testing checks how well the system performs under stress, heavy loads, or adverse conditions. It includes testing methods such as load testing, stress testing, security testing, and usability testing.

In short, Functional testing is about testing what the software should do, while Non-Functional testing is about testing how well the software does it.

Q-Difference between STLC and SDLC?

A-The Software Testing Life Cycle (STLC) and Software Development Life Cycle (SDLC) are two important processes in software development.

SDLC is the process of developing software from planning to maintenance. It includes stages such as requirements gathering, design, development, testing, deployment, and maintenance.

STLC is the process of testing software from planning to closure. It includes stages such as test planning, test design, test execution, and test closure.

In short, SDLC is about developing the software, while STLC is about testing the software.

Q-Difference between test scenarios, test cases, and test scripts?

A-Test scenarios, test cases, and test scripts are all part of the software testing process, but they have different purposes.

Test scenarios are high-level descriptions of the end-to-end testing process. They are used to identify different testing conditions and determine the scope of the testing.

Test cases are detailed steps that need to be executed to verify a specific test scenario. They are used to ensure that the software meets the requirements and specifications.

Test scripts are automated scripts that execute a series of steps to perform a test case. They are used to automate the testing process and save time and effort.

In short, test scenarios define what needs to be tested, test cases describe how to test it, and test scripts automate the testing process.

Q-What is a Test Plan?

A-A Test Plan is a document that outlines the testing approach, objectives, scope, and resources for a software testing project. It describes how the testing process will be conducted, what types of testing will be performed, and what will be the criteria for test completion.

The information that should be covered in a Test Plan includes:

Introduction and overview of the testing project

Scope and objectives of the testing

Testing approach and strategies

Testing types and techniques to be used

Test environment and tools to be used

Test schedule and milestones

Test deliverables and reporting

Risks and contingencies

Roles and responsibilities

Q-What is Priority?

A-Priority is a measure of the urgency or importance of fixing a defect or issue. It indicates how soon the defect needs to be addressed relative to other defects. Priority levels can be set as High, Medium, or Low.

Q-What is Severity?

A-Severity is a measure of the impact or severity of a defect or issue on the software's functionality or user experience. It indicates the criticality of the defect in terms of its effect on the system. Severity levels can be classified as Critical, Major, Minor, or Cosmetic.

Q-Bug Categories are?

A-Bug categories are the different types of defects or issues that can occur in software testing. The common bug categories include:

Functional Bugs: Bugs that affect the functionality or behavior of the software.

Performance Bugs: Bugs that affect the performance or speed of the software.

Usability Bugs: Bugs that affect the user experience or usability of the software.

Compatibility Bugs: Bugs that affect the compatibility of the software with different systems or platforms.

Security Bugs: Bugs that affect the security or privacy of the software.

Installation/Configuration Bugs: Bugs that affect the installation or configuration of the software.

Q-Advantage of Bugzilla?

A-Bugzilla is an open-source bug tracking tool that offers several advantages, including:

It is free to use and customize

It is easy to install and configure

It offers a wide range of features for bug tracking and management

It provides a centralized platform for communication and collaboration among team members

It allows for easy reporting and analysis of bugs and issue

Q-Difference between Priority and Severity?

A-Priority and Severity are two important attributes in bug tracking and management.

Priority refers to the level of urgency or importance of fixing a defect or issue. It indicates how soon the defect needs to be addressed relative to other defects. Priority levels can be set as High, Medium, or Low.

Severity refers to the level of impact or severity of a defect or issue on the software's functionality or user experience. It indicates the criticality of the defect in terms of its effect on the system. Severity levels can be classified as Critical, Major, Minor, or Cosmetic.

In short, Priority determines the urgency of fixing an issue, while Severity determines the impact of the issue on the software.

Q-Different Methodologies in Agile Development Model?

A-There are several methodologies within the Agile Development Model, including:

Scrum: A framework that focuses on iterative and incremental development. It involves a team of cross-functional members working together in short sprints to deliver a potentially shippable product increment at the end of each sprint.

Kanban: A visual framework that focuses on the continuous delivery of software by using a pull-based system. Work items are visualized on a Kanban board and pulled through the workflow as capacity allows.

Extreme Programming (XP): A methodology that emphasizes software engineering practices such as continuous integration, automated testing, and pair programming. It also involves frequent releases and customer involvement in the development process.

Lean Development: A methodology that focuses on delivering value to the customer by reducing waste and optimizing the delivery process. It involves a continuous improvement cycle and a culture of experimentation and learning.

Feature-Driven Development (FDD): A methodology that focuses on delivering features in a timely and efficient manner. It involves a structured process of feature development, including domain walkthroughs, feature design, and feature build.

Crystal: A family of methodologies that focuses on the team's size, criticality of the project, and other factors. It involves a flexible approach to development and emphasizes communication, collaboration, and continuous improvement.