

Module 2 Assignment

Question 1: What is Exploratory Testing?

Answer 1: “Exploratory Testing” – as the name suggests, is a simultaneous learning, test design, and test execution process. We can say that in this testing test planning, analysis, design, and test execution, are all done together and instantly.

Question 2: What is traceability matrix?

Answer 2: To protect against changes you should be able to trace back from every system component to the original requirement that caused its presence.

Question 3: What is Boundary value testing?

Answer 3: Boundary-value analysis is a software testing technique in which tests are designed to include representatives of boundary values in a range.

Question 4: What is Equivalence partitioning testing?

Answer 4: Equivalence partitioning or equivalence class partitioning (ECP) is a software testing technique that divides the input data of a software unit into partitions of equivalent data from which test cases can be derived. In principle, test cases are designed to cover each partition at least once.

Question 5: What is Integration testing?

Answer 5: Integration testing (sometimes called integration and testing, abbreviated I&T) is the phase in software testing in which individual software modules are combined and tested as a group.

Question 6: What determines the level of risk?

Answer 6: The likelihood of an adverse event and the impact of the event

Question 7: What is Alpha testing?

Answer 7: Alpha testing is the initial phase of validating whether a new product will perform as expected. Alpha tests are carried out early in the development process by internal staff and are followed up with beta tests, in which a sampling of the intended audience tries the product out.

Question 8: What is beta testing?

Answer 8: Beta testing is an opportunity for real users to use a product in a production environment to uncover any bugs or issues before a general release. Beta testing is the final round of testing before releasing a product to a wide audience.

Question 9: What is component testing?

Answer 9: Component testing is a form of closed-box testing, meaning that the test evaluates the behaviour of the program without considering the details of the underlying code. Component testing is done on the section of code in its entirety, after the development has been completed.

Question 10: What is functional system testing?

Answer 10: Functional testing is a quality assurance (QA) process and a type of black-box testing that bases its test cases on the specifications of the software component under test. Functions are tested by feeding them input and examining the output, and internal program structure is rarely considered unlike white-box testing.

Question 11: What is Non-Functional Testing?

Answer 11: Non-functional testing is a type of software testing that verifies non-functional aspects of the product, such as performance, stability, and usability. non-functional testing verifies how well the product performs.

Question 12: What is GUI Testing?

Answer 12: Graphic User Interface Testing (GUI) testing is the process of ensuring proper functionality of the graphical user interface (GUI) for a specific application. This involves making sure it behaves in accordance with its requirements and works as expected across the range of supported platforms and devices.

Question 13: What is Ad hoc testing?

Answer 13: Performing random testing without any plan is known as Ad Hoc Testing. It is also referred to as Random Testing or Monkey Testing. This type of testing does not follow any documentation or plan to perform this activity.

Question 14: What is load testing?

Answer 14: Load testing examines how the system behaves during normal and high loads and determines if a system, piece of software, or computing device can handle high loads given a high demand of end-users. This tool is typically applied when a software development project nears completion.

Question 15: What is stress Testing?

Answer 15: Stress testing is a form of deliberately intense or thorough testing, used to determine the stability of a given system, critical infrastructure, or entity. It involves testing beyond normal operational capacity, often to a breaking point, in order to observe the results.

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

Answer 16: White box testing is a technique that uses a program's internal or source code to design different test cases to check the quality of the program. In this technique, the internal structure and implementation of how an application works are known to the tester. Path testing, Loop testing, Condition testing, Testing based on the memory perspective, Test performance of the program.

Question 17: What is black box testing?

Answer 17: 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.

Question 18: What are the different black box testing techniques?

Answer 18: Equivalence partitioning, Boundary value analysis, Decision tables, State transition testing, Use-case Testing, Other Black Box Testing (Syntax or Pattern Testing).

Question 19: Mention what are the categories of defects?

Answer 19: Data Quality/Database Defects, Critical Functionality Defects, Functionality Defects, Security Defects, User Interface Defects.

Question 20: Mention what big bang testing is?

Answer 20: Big Bang Integration Testing is an integration testing strategy wherein all units are linked at once, resulting in a complete system. When this type of testing strategy is adopted, it is difficult to isolate any errors found, because attention is not paid to verifying the interfaces across individual units.

Question 21: What is the purpose of exit criteria?

Answer 21: Exit criterion is used to determine whether a given test activity has been completed or NOT. Exit criteria can be defined for all of the test activities right from planning, specification and execution. Exit criterion should be part of test plan and decided in the planning stage.

Question 22: When should "Regression Testing" be performed?

Answer 22: A new requirement is added to an existing feature, A new feature or functionality is added, the codebase is fixed to solve defects, The source code is optimized to improve performance, Patch fixes are added, Changes in configuration.

Question 23: What is 7 key principles? Explain in detail?

Answer 23:

1. Testing shows 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.
- We test to find Faults As we find more defects, the probability of undiscovered defects remaining in a system reduces.
- However, Testing cannot prove that there are no defects present.

2. 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 have learned that we cannot test everything (i.e., all combinations of inputs and pre-conditions).
- That is, we must Prioritise our testing effort using a Risk Based Approach.

3. 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

4. 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'.
- In other words, most defects found during testing are usually confined to a small number of modules

5. The Pesticide Paradox:

- If the same tests are repeated, 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.

6. Testing is Context Dependent:

- Testing is basically context dependent.
- Testing is done differently in different contexts.
- Different kinds of sites are tested differently.
- Whilst, Testing can be 50% of development costs, in NASA's Apollo program it was 80% testing.
- 3 to 10 failures per thousand lines of code (KLOC) typical for commercial software.
- 1 to 3 failures per KLOC typical for industrial software.
- 0.01 failures per KLOC for NASA Shuttle code.
- Also, different industries impose different testing standards.

7. Absence of Errors Fallacy:

- If the system built is unusable and does not fulfil the user's needs and expectations then finding and fixing defects does not help.
- Even after defects have been resolved it may still be unusable and/or does not fulfil the users' needs and expectations

Question 24: Difference between QA v/s QC v/s Tester

Answer 24:

Quality Assurance:

- Activities which ensure the implementation of processes, procedures and standards in context to verification of developed software and intended requirements.
- Focuses on processes and procedures rather than conducting actual testing on the system.
- Process oriented activities.
- Preventive activities.
- It is a subset of Software Test Life Cycle (STLC).

Quality Control:

- Activities which ensure the verification of developed software with respect to documented (or not in some cases) requirements.
- Focuses on actual testing by executing Software with intend to identify bug/defect through implementation of procedures and process.
- Product oriented activities.
- It is a corrective process.
- QC can be considered as the subset of Quality Assurance.

Testing:

- Activities which ensure the identification of bugs/error/defects in the Software.
- Focuses on actual testing.
- Product oriented activities.
- It is a preventive process.
- Testing is the subset of Quality Control.

Question 25: Difference between Smoke and Sanity?

Answer 25:

Smoke:

- Smoke Testing is performed to ascertain that the critical functionalities of the program is working fine.
- The objective of this testing is to verify "stability" of the system in order to with more rigorous testing.
- This testing is performed by the developers or testers.
- Smoke testing is usually documented or scripted is unscripted.
- Smoke testing is a subset of Regression testing.
- Smoke testing exercises the entire system from end to end.
- Smoke testing is like General Health Check Up.

Sanity:

- Sanity Testing is done to check the new functionality / bugs have been fixed.
- The objective of the testing is to verify the e "rationality" of the system in order proceed with more rigorous testing.
- Sanity testing is usually performed by testers.
- Sanity testing is usually not documented and is unscripted.
- Sanity testing is a subset of Acceptance testing.
- Sanity testing exercises only the component of the entire system.
- Sanity Testing is like specialized health Up check-up.

Question 26: Difference between verification and Validation

Answer 26:

Verification: Verification is a process of determining if the software is designed and developed as per the specified requirements.

Validation: Validation is the process of checking if the software (product) has met the client's true needs and expectations.

Question 27: Explain types of Performance testing.

Answer 27:

- 1. Load testing:** It is a performance testing to check system behaviour 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.
- 2. Stress testing:** Stress testing is to test the system behaviour under extreme conditions and is carried out till the system failure.

Question 28: What is Error, Defect, Bug, and failure?

Answer 28:

- 1. Failure:** 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.
- 2. Error:** A discrepancy between a computed, observed, or measured value or condition and the true, specified, or theoretically correct value or condition. This can be a misunderstanding of the internal state of the software, an oversight in terms of memory management, confusion about the proper way to calculate a value, etc.
- 3. Bug:** A fault in a program which causes the program to perform in an unintended or unanticipated manner. See: anomaly, defect, error, exception, and fault. Bug is terminology of Tester.
- 4. Defect:** Commonly refers to several troubles with the software products, with its external behaviour or with its internal features.

Question 29: Difference between Priority and Severity.

Answer 29:

Severity:

- Severity is absolute and Customer-Focused. It is the extent to which the defect can affect the software. In other words, it defines the impact that a given defect has on the system.

Priority:

- Priority is Relative and Business-Focused. Priority defines the order in which we should resolve a defect. Should we fix it now, or can it wait? This priority status is set by the tester to the developer mentioning the time frame to fix the defect. If high priority is mentioned then the developer must fix it at the earliest. The priority status is set based on the customer requirements.

Question 30: What is Bug Life Cycle?

Answer 30: A computer bug is an error, flaw, mistake, failure, or fault in a computer program that prevents it from working correctly or produces an incorrect result. Bugs arise from mistakes and errors, made by people, in either a program's source code or its design.

The duration or time span between the first-time defects is found and the time that it is closed successfully, rejected, postponed, or deferred is called as 'Defect Life Cycle'.

Question 31: Explain the difference between Functional testing and Non-functional testing

Answer 31:

Functional Testing:

- Functional testing is performed using the functional specification provided by the client and verifies the system against the functional requirements.
- Functional testing is executed first.
- Manual testing or automation tools can be used for functional testing.
- Business requirements are the inputs to functional testing.
- Functional testing describes what the product does.
- Easy to do manual testing.

Non-Functional:

- Non-Functional testing checks the Performance, reliability, scalability, and other non-functional aspects of the software system.
- Non-functional testing should be performed after functional testing.
- Using tools will be effective for this testing.
- Performance parameters like speed, scalability are inputs to non-functional testing.
- Non-functional testing describes how good the product works.
- Tough to do manual testing

Question 32: What is the difference between test scenarios, test cases, and test script?

Answer 32:

Test Scenario:

- A Scenario is any functionality that can be tested. It is also called Test Condition, or Test Possibility.
- Test Scenario is 'What to be tested'
- Test scenario is nothing but test procedure.
- Test scenarios are derived from use cases.
- Test Scenario represents a series of actions that are associated together.
- Scenario is thread of operations.

Test Case:

- Test cases involve the set of steps, conditions and inputs which can be used while performing the testing tasks.
- Test Case is 'How to be tested' Test case consist of set of input values, execution precondition, expected Results and executed post-condition developed to cover certain test Condition.
- Test cases are derived (or written) from test scenario.
- Test Case represents a single (low level) action by the user.
- Test cases are set of input and output given to the System.

Test Scripts (Test Procedures):

- The Test Procedures Specification specifies the sequence of actions for a test.
- It is also known as a Test Script.
- The Test Script can be manual or automated.

Question 33: Explain what Test Plan is? What is the information that should be covered.

Answer 33:

Question 34: What is priority?

Answer 34: Priority is Relative and Business-Focused. Priority defines the order in which we should resolve a defect. Should we fix it now, or can it wait? This priority status is set by the tester to the developer mentioning the time frame to fix the defect. If high priority is mentioned then the developer must fix it at the earliest. The priority status is set based on the customer requirements.

Question 35: What is severity?

Answer 35: Severity is absolute and Customer-Focused. It is the extent to which the defect can affect the software. In other words, it defines the impact that a given defect has on the system.

Question 36: Bug categories.

Answer 36: Bugs can be classified as Urgent, High, Medium, and Low.

Question 37: Advantage of Bugzilla.

Answer 37:

Advantages:

- It is an open-source widely used bug tracker.
- It is easy in usage and its user interface is understandable for people without technical knowledge.
- It easily integrates with test management instruments.
- It integrates with an e-mailing system.
- it automates documentation.

Question 38: Difference between priority and severity

Answer 38: ANSWER OF QUESTION 29.

Question 39: What are the different Methodologies in Agile Development Model?

Answer 39:

- Kanban.
- Scrum.
- Extreme Programming (XP).
- Crystal.
- Dynamic Systems Development Method (DSDM).
- Feature-Driven Development (FDD).
- Lean Software Development.
- Scaled Agile Frameworks (SAF).

Question 40: Explain the difference between Authorization and Authentication in Web testing. What are the common problems faced in Web testing?

Answer 40:

Authentication:

- Authentication verifies who the user is.
- Authentication works through passwords, one-time pins, biometric information, and other information provided or entered by the user.
- Authentication is the first step of a good identity and access management process.
- Authentication is visible to and partially changeable by the user.

Authorization:

- Authorization determines what resources a user can access.
- Authorization works through settings that are implemented and maintained by the organization.
- Authorization always takes place after authentication.
- Authorization isn't visible to or changeable by the user.

Common Problems:

- Insufficient testing for browser compatibility.
- Failing to conduct thorough functional testing across mobile.
- Failing to conduct thorough functional testing across desktop.
- Poor data security.
- The most common bugs.

Question 41: When to used Usability Testing?

Answer 41: Before Any Design Decisions Are Made. When It is Time to Evaluate and Iterate. After Launch. In High-Risk, Low-Certainty Situations.

Question 42: What is the procedure for GUI Testing?

Answer 42: Graphic User Interface Testing (GUI) testing is the process of ensuring proper functionality of the graphical user interface (GUI) for a specific application. This involves making sure it behaves in accordance with its requirements and works as expected across the range of supported platforms and devices.

GUI testing refers to testing the functions of an application that are visible to a user. In the example of a calculator application, this would include verifying that the application responds correctly to events such as clicking on the number and function buttons.

Question 43: What is the difference between the STLC (Software Testing Life Cycle) and SDLC (Software Development Life Cycle)?

Answer 43:

SDLC (Software Development Life Cycle):

- SDLC is mainly related to software development.
- Besides development other phases like testing is also included
- SDLC involves total six phases or steps.
- In SDLC, more number of members (developers) are required for the whole process.
- In SDLC, development team makes the plans and designs based on the requirements.
- Goal of SDLC is to complete successful development of software.
- It helps in developing good quality software.
- SDLC phases are completed before the STLC phases.
- Post deployment support, enhancement , and update are to be included if necessary.
- Creation of reusable software systems is the result of SDLC.

STLC (Software Testing Life Cycle):

- STLC is mainly related to software testing.
- It focuses only on testing the software.
- STLC involves only five phases or steps.
- In STLC, a smaller number of members (testers) are needed.
- In STLC, testing team (Test Lead or Test Architect) makes the plans and designs
- Goal of STLC is to complete successful testing of software.
- It helps in making the software defects free.
- STLC phases are performed after SDLC phases.
- Regression tests are run by QA team to check deployed maintenance code and maintains test cases and automated scripts.
- A tested software system is the result of STLC.