Software Testing Assignment

Module -2 (Manual Testing)

1. What is Exploratory testing?

1. Exploratory testing is a concurrent process where-

- Test design, execution and logging happen simultaneously.
- Testing is often nor recorded.
- Makes use of experience, heuristics and test patterns.

2. What is traceability matrix?

2. A traceability matrix is a document that details the technical requirements for a given test scenario and its current state.

3. What is Boundary Value Testing?

3. Boundary Value testing is a methodology for designing test cases that concentrate software testing efforts on cases near the limits of valid ranges.

4. What is Equivalence Partitioning Testing?

4. Equivalence Partitioning says that by testing just one value we have tested the partition (typically a mid-value is used). It assumes that:

- If one value finds a bug, the others probably will too.
- If one doesn’t find a bug, the others probably won’t either.
- The valid partition is bonded by the values 1 and 100.
- Plus, there are 2 invalid partitions.

5. What is Integration testing?

5. Testing performed to expose defects in the interface and in the interactions between integrated components or system. Integration testing is a level of the software testing process where individual units are combined and tested as a group.

6. What determines the level of risk?

6. A factor that could result in future negative consequences; usually expressed as impact and likelihood.

7. What is Alpha testing?

7. Alpha testing is the first phase of formal testing; it is performed by the developers at the software development site.

- Alpha testing is not open to the market and public.
- Sometimes it is also performed by independent testing team.
- It is always performed in virtual environment.

8. What is Beta testing?

8. In Beta testing, the software is tested by a large group of users, typically outside of the organization that developed it.

- Beta testing is always open to the market and public.
- It is not performed by independent testing team.
- It is usually conducted for software product.

9. What is component testing?

9. The testing of individual software component. It's also called unit testing. A minimal software item that can be tested in isolation. It means "A unit is the smallest testable part of software".

10. What is functional system testing?

10. Functional testing is performed using the functional specifications provided by the client and verifies the system against the fundamental requirement.

11. What is non-functional system testing?

11. Non-functional testing checks the performance, reliability, scalability and other non-functional aspects of the software system.

12. What is GUI testing?

12. Graphical User Interface (GUI) testing is the process of testing the system's GUI of the system under test.

13. What is Adhoc testing?

13. Adhoc testing is an informal testing type with an aim to break the system. It does not follow any test design techniques to create test cases. Testers randomly test the application without any test cases or any business requirement documents.

14. What is load testing?

14. Load testing is to test the system behavior under normal work load conditions, and it is just testing or simulating with the actual work load.

15. What is stress testing?

15. Stress testing is to test the system behavior under extreme conditions, and it is carried out till the system failure.

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

16. White box testing is also called glass testing or open box testing. In order to perform white box testing on an application, the tester needs to posses' knowledge of the internal working of the code.

There are 3 types of coverage:

- Statement Coverage
- Decision Coverage
- Condition Coverage

17. What is Black box testing? What are the different Black box testing techniques?

17. Specification based testing techniques is also known as 'Black box' or input-output driven testing techniques because they view the software as black box with inputs and outputs. The tester has 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.

There are 4 black box techniques:

- Equivalence Partitioning
- Boundary Value Analysis
- Decision Table
- State Transition Testing

18. Mention what are the categories of defects?

18. Categories of Defects: -

- Functional Defects
- Performance Defects
- Usability Defects
- Compatibility Defects
- Security Defects

- Reliability Defects
- Interface Defects
- Data Defects
- Configuration Defects
- Documentation Defects

19. Mention what Big Bang testing is?

19. In Big Bang integration testing all components or modules is integrated simultaneously, after which everything is tested as a whole.

20. What is the purpose of exit criteria?

20. exit criteria defines the items that must be completed before testing can be concluded.

21. When should be “Regression Testing” performed?

21. Regression Testing should be performed:

- When change in requirement and code is modified according to the requirement.
- When new feature is added to the software.
- When defect is fixing.
- When performance issue fix.

22. What is 7 key principles? Explain in detail?

22. 7 key principles are: -

- 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.

- 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.

  For example: In an application in one screen there are 15 input fields, each having 5 possible values, then to test all the valid combinations you would need 30 517 578 125 (515) tests.

- 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.

  These activities should be focused on defined objectives – outlined in the Test Strategy.

- 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

- The Pesticide Paradox: -

  If the same tests are repeated over and over again, 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.

- Testing is Context Dependent: -

  Testing is basically context dependent & It's done differently in different contexts.

  Different kinds of sites are tested differently. For example: - Safety – critical software is tested differently from an e-commerce site.

- Absence of Errors Fallacy: -

  If the system built is unusable and does not fulfill the user's needs and expectations then finding and fixing defects does not help.

  If we build a system and, in doing so, find and fix defects & it doesn't make it a good system.

  Even after defects have been resolved it may still be unusable and/or does not fulfil the users' needs and expectations.

23.Difference between QA v/s QC v/s Tester

S. No	Quality Assurance	Quality Control	Testing
1	Activities which ensure the implementation of processes, procedures and standards in context to verification of developed software and intended requirements	Activities which ensure the verification of developed software with respect to documented (or not in some cases) requirements.	Activities which ensure the identification of bugs/error/defects in the Software.
2	Focuses on processes and	Focuses on actual testing by executing Software with	Focuses on actual testing.

	procedures rather than conducting actual testing on the system.	intend to identify bug/defect through implementation of procedures and process.	
3	Process oriented activities	Product oriented activities	Product oriented activities.
4	Preventive activities.	It is a corrective process	It is a preventive process
5	It is a subset of Software Test Life Cycle (STLC).	QC can be considered as the subset of Quality Assurance	Testing is the subset of Quality Control.

24. Difference between Smoke and Sanity?

Smoke Testing	Sanity Testing
Smoke Testing is performed to ascertain that the critical functionalities of the program is working fine	Sanity Testing is done to check the new functionality / bugs have been fixed
The objective of this testing is to verify the "stability" of the system in order to proceed with more rigorous testing	The objective of the testing is to verify the "rationality" of the system in order to proceed with more rigorous testing
This testing is performed by the developers or testers	Sanity testing is usually performed by testers
Smoke testing is usually documented or scripted	Sanity testing is usually not documented and is unscripted
Smoke testing is a subset of Regression testing	Sanity testing is a subset of Acceptance testing
Smoke testing exercises the entire system from end to end	Sanity testing exercises only the particular component of the entire system
Smoke testing is like General Health Check Up	Sanity Testing is like specialized health check up

25.Difference between verification and Validation

Criteria	Verification	Validation
Definition	The process of evaluating work-products (not the actual final product) of a development phase to determine whether they meet the specified requirements for that phase.	The process of evaluating software during or at the end of the development process to determine whether it satisfies specified business requirements.
Objective	To ensure that the product is being built according to the requirements and design specifications. In other words, to ensure that work products meet their specified requirements.	To ensure that the product actually meets the user's needs, and that the specifications were correct in the first place. In other words, to demonstrate that the product fulfils its intended use when placed in its intended environment.
Question	Are we building the product, right?	Are we building the right product?
Evaluation Items	Plans, Requirement Specs, Design Specs, Code, Test Cases	The actual product/software.
Activities	Reviews, Walkthroughs, Inspections	Testing

26.Explain types of Performance testing?

26. Types of Performance testing: -

- Load testing
- Stress testing
- Endurance testing
- Spike testing
- Volume testing
- Scalability testing

27. What is Error, Defect, Bug and failure?

- Error: -

A mistake in coding is called error.

- Defect: -

Error found by tester is called defect.

- Bug: -

Defect accepted by development team then it is called bug.

- Failure: -

Build does not meet the requirements then it is failure.

28. Explain the difference between Functional testing and non-functional testing?

Functional testing	Non-Functional testing
It verifies the operations and actions of an application.	It verifies the behaviour of an application.
It is based on requirements of customer.	It is based on expectations of customer.
It helps to enhance the behaviour of the application.	It helps to improve the performance of the application.
Functional testing is easy to execute manually.	It is hard to execute non-functional testing manually.
It tests what the product does.	It describes how the product does.
Functional testing is based on the business requirement.	Non-functional testing is based on the performance requirement.
Examples: • Unit Testing • Smoke Testing • Integration Testing • Regression Testing	Examples: • Performance Testing • Load Testing • Stress Testing • Scalability Testing

29. 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)
STLC is mainly related to software testing.	SDLC is mainly related to software development.
It focuses only on testing the software.	Besides development other phases like testing is also included.
In STLC, less number of members (testers) are needed.	In SDLC, more number of members (developers) are required for the whole process.
In STLC, testing team (Test Lead or Test Architect) makes the plans and designs.	In SDLC, development team makes the plans and designs based on the requirements.
Goal of STLC is to complete successful testing of software.	Goal of SDLC is to complete successful development of software.
It helps in making the software defects free.	It helps in developing good quality software.
STLC phases are performed after SDLC phases.	SDLC phases are completed before the STLC phases.
STLC involves only five phases or steps.	SDLC involves total six phases or steps.
Regression tests are run by QA team to check deployed maintenance code and maintains test cases and automated scripts.	Post deployment support, enhancement, and update are to be included if necessary.
A tested software system is the end result of STLC.	Creation of reusable software systems is the end result of SDLC.

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

Test Scenarios	Test Cases	Test Script
A test scenario is any functionality that a software testing company can examine. It is also called a Test Condition or Test Possibility.	A test case is a document that lists the steps a QA engineer needs to execute.	A test script is a short program written in a programming language.

Any functionality that can be tested.	A set of actions executed to verify particular features or functionality.	A set of instructions to test an app automatically.
Its's derived from Software requirement specification (SRS).	It's derived from test scenarios.	It's derived from test cases.
It helps to test the end-to-end functionality in an Agile way.	It helps in exhaustive testing of an app.	It helps to test specific things repeatedly.
It is focused on what to test.	It is focused on what to test & How to test.	It is focused on the expected results.
It's takes less time & fewer resources to create.	It requires more resources & time.	It requires less time for testing but more resources for scripts rating & updating
It allows quickly assessing the testing scope.	It allows detecting errors & defects.	It allows carrying out an automatic execution of test cases

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

- A test plan is a detailed document that provides guidance around specific test activities, scope, deliverables, and resource projections for the project.

  A test plan should include objectives, scope, approach, resources, schedule, test deliverables, dependencies, test environment, risk management, roles and responsibilities, and a communication plan.

32. What are the different Methodologies in Agile Development Model?

- There are various methodologies present in agile development model: -
  - Scrum
  - Extreme Programming
  - Adaptive Software Development (ASD)
  - Dynamic System Development Method (DSDM)

- Test Driven Development (TDD)
- Feature Driven Development
- Kanban
- Behavior Driven Development (BDD)

33. When to used Usability Testing?

- Although you can usability test at any time (in development, pre-release, or post release) and still obtain valuable information, due to time and resource constraints, organizations often decide on specific times to perform usability testing.

34. What is the procedure for GUI Testing?

- It is the process for ensuring proper functionality of the graphical user interface (GUI) for a specific application. GUI testing generally evaluates a design of elements such as layout, colors and also fonts, font sizes, labels, text boxes, text formatting, captions, buttons, lists, icons, links, and content. GUI testing processes may be either manual or automatic and are often performed by third-party companies, rather than developers or end users.