MODULE-2

(MANUAL TESTING)

1.What is traceability matrix ?

→ To protect against changes you should be able to trace back from every system component to the original requirement that caused its presents.

→ It is used to track the requirements and to check the current project requirements are met.

→ Requirement traceability matrix is a document that maps and traces user requirement with test cases.

❖ Types of traceability matrix

1. Forward traceability

→ This matrix is used to check whether the project progresses in the desired direction and for the right product.

2. Backward or reverse traceability

→ It is used to ensure whether the current product remains on the right track.

3. Bi-directional traceability

→ This traceability matrix ensure that all requirements are covered by test cases.

2. What is integration testing ?

→ Integration testing is testing performed to exposed defects in the interfaces and interaction between integrated components.

→ Integration testing is defined as a type of testing where software modules are integrated logically and tested as a group.

❖ Types of integration testing

1.Big bang approach

2.Incremental integration approach

→ Top down approach

→ Bottom up approach

3.What is alpha testing ?

→ It is always performed by the developers at the software development site.

→ Alpha testing is a type of software testing performed to identify bugs before releasing the software product to the real user or public.

→ It is the final testing stage before the software is release into the real world.

4.What is beta testing ?

→ It is always performed by the customers at their own site.

→ Direct feedback from customer is a major advantage of beta testing.

→ This testing helps to test product in customers environments.

5.What is component testing ?

→ A minimal software item that can be tested in isolation. It means “A unit is the smallest testable part of software.

→ Component level testing is deal with testing these components individually.

→ Unit testing is a white box of testing technique that is usually performed by the developer.

6.What is functional system testing?

→ A requirement that specifies a function that a system or system component must performed.

❖ There are two types of approach

→ Requirement based functional testing

→ Process based testing

❖Functional system testing functionality

Accuracy	Provision of right or agreed result or effect
Interoperability	Ability to interact with specified systems
Compliance	Adhere to applicable standards, conventions, regulations or laws
Auditability	Ability to provide adequate and accurate audit data
Suitability	Presence and appropriateness of functions for specified tasks

7. What is boundary value testing?

- → Boundry value analysis is a methodology for designing test cases that concentrates software testing effort on cases near the limits of valid ranges.
- → Boundry value analysis is a testing the boundaries at partitions.
- → The minimum and maximum values of a partition are its boundary values.
- → Boundary value analysis can perform at all test levels, and its primarily used for a range of numbers, dates and time.

8. What is the equivalent partitioning testing?

- → Aim is to treat groups of inputs as equivalent and to select one representative input to test them all
- → EP can be used for all levels of testing.
- → Partitioning usually happens for test objects, which include inputs, output, time related value and interface parameters.
- → If one of the input conditions fails, then all other input conditions within the partition will fail as well.

→ If one of the input conditions passes, then all other input conditions within the partition will pass as well.

9.What is non-functional testing?

→ Non- functional testing check 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 is input to non-functional testing.

→ Non-functional testing describes how good the product works.

→ Tough to do manual testing.

→ Types of non-functional testing are

1.Performance testing

2.Load testing

3.Volume testing

4.Stress testing

5.Security testing

10. What is black box testing? What are the different black box testing techniques ?

- → Black box testing is either functional or non-functional, without reference to the internal structure of the component or system.
- → Thes are scientific and time tested techniques that will help us to get maximum coverage in minimum test cases.
- → Black box testing is test without execute the code.

- ❖ Types of black box testing techniques
  1. Equivalence partitioning
  2. Boundry value analysis
  3. Decision table testing
  4. State transition testing
  5. Use case testing

11.Mention what big bang testing is ?

- → In big bang integration testing all components or modules are integrated simultaneously, after which everything is tested as whole.
- → Big bang testing has the advantage that everything is finished before integration testing starts.
- → This combined set of components is considered as an entity while testing. If all of the components in the unit are not completed, the integration testing process will not execute.

12.What are the 7 key principles? Explain in detail?

1.Testing shows presence of defect

2.Exhaustive testing is impossible

3.Early testing

4.Defect clustering

5.The pesticide paradox

6.Testing is context dependent

7.Absence of error fallacy

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.

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.

→ That we must priorities 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.

→ So that any defect in the requirements or design phase are captured in early stage. It is much cheaper to fix a defect in the early stage of testing.

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 failure.

→ Defects are not evenly spread in a system.

→ They are clustered

→ Approximately 80% of the problems are found in 20% of module

5. 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.

→ Testers cannot simply depend on existing test techniques. He must look out continually to improve the existing methods to make testing more effective.

6. Testing is context dependent

→ Testing is basically context dependent.

→ Testing is context dependent which basically means that the way you test an e-commerce site will be different from the way you test a commercial off the shelf application.

→ All the developed software's are not identical. You might use a different approach, methodologies, techniques, and types of testing depending upon the application type.

7. Absence of error fallacy

→ If the system is build unusable and does not fulfil the users' needs and expectations, then finding and fixing defects does not help.

→ It is possible that software which is 99% bug free is still not unusable.

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

Quality Assurance	Quality control	Testing
Activities which ensure the implementation of processes, procedure and standards in context to verification of developed software and intended requirements.	Activities which ensure the verification of developed software with respect to documented requirements.	Activities which ensure the identification of bugs/error/defects in the software.
Focus on processes and procedures rather than conducting actual testing on the system.	Focus on actual testing by executing software with intent to identify bug/defect	Focus on actual testing.

	through implementation of procedures and process.	
Process-oriented activities.	Product oriented activities.	Product oriented activities.
Preventing activities.	It is a corrective process.	It is a preventive process.
It is subset of software testing life cycle (STLC)	QC can be considered as the subset of Quality assurance.	Testing is a subset of Quality control

14. Difference between smoke and sanity?

Smoke	Sanity
→ Smoke testing is performed to ascertain that the critical functionality of the	→ Sanity testing is done to check the new functionality/ bugs have been fixed.

program is working fine.	
→ The objective of this testing is to verify the “stability” of the system 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 developer or tester	→ Sanity testing in software testing is usually performed by testers
→ Smoke testing is documented or scripted	→ Sanity testing usually not documented and is unscripted
→ Smoke testing is subset of acceptance testing	→ Sanity testing is a subset of regression testing

15. Difference between verification and validation

Verification	Validation
→ The verifying process includes	→ It is a dynamic mechanism of testing

checking documents, design, code and program	and validating the actual product
→ It does not involve executing the code	→ It always involves executing the code
→ Verification use methods like revies, walkthrough, inspection and desk-checking	→ It uses methods like black box testing, white box testing, and non- functional testing
→ It finds the bugs early in the development cycle	→ It can find bugs that the verification process cannot catch
→ It comes before validation	→ It comes after verification

16. What is Error, defect, bug and 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

1. Errors

→ Error is a situation that happens when the development team or the developer fails to understand a requirement definition and hence that misunderstanding gets translated to buggy.

2. Defect

→ A defect refers to the situation when the application is not working as per the requirement and the actual and expected result of the application or software are not in sync with each other.

3. Bug

→ A bug refers to a defect which means that the software product or the application is not working as per the adhered requirements.

4. Failure

→ The inability of a system or component to perform its required functions within specified performance requirements.

17.Explain types of performance testing?

1. Stress Testing

→ This test pushes an application beyond normal load conditions to determine which components fails first. Stress testing attempts to find the breaking point of the application and is used to evaluate the robustness of the application's data processing capabilities and response to high volume of traffic.

2. Spike Testing

→ This testing evaluates the ability of the application to handle sudden volume increases. It is done by suddenly increasing the load generated by a very large number of users.

3. Load Testing

→ The purpose of the load testing is to evaluate the application's performance under an increasingly high number of users.

4. Endurance Testing

→ Endurance testing evaluate the performance of the system under load over time.

5. Volume Testing

→ Also known as flow testing, this testing evaluates the application's ability to handle a large volume of data. The impact of response time and the behavior of the application are analyzed.

6. Scalability Testing

→ This testing is used to determine your application’s ability to handle increasing amount of load and processing.

18. Explain the difference between Functional and Non-Functional testing?

Functional	Non-Functional
→ Functional testing is performed before Non-functional testing.	→ It is performed after the functional testing
→ It is based on customer requirements.	→ It focusses on the customer’s expectations.

→ It is easy to define functional requirements.	→ It is difficult to define the requirements for non-functional testing.
→ Helps to validate the behavior of the application	→ Helps to validate the performance of the application
→ Functional testing is easy to execute by manual testing.	→ It is very hard to perform non-functional testing manually.
→ It describes what the product does.	→ It describes how the product works.
→ Unit testing → Smoke testing → User acceptance → Integration testing	→ Performance testing → Volume testing → Scalability testing → Load testing

19. What is GUI testing?

→ 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.

- It provides a customizable test report.
- It is run test in parallel or distributed on a selenium grid with build in selenium web driver.
- It allows you to test the functionality from a user's perspective.

20. What is ad hoc testing?

→ Ad hoc testing is an informal testing type with an aim to break the system.

→ In fact is does not create test cases altogether

→ It is also known as random testing or monkey testing.

→ Ad hoc testing is not performed is an structured way so it is not based on any methodological approach.

→ Ad hoc testing can be achieved with the testing technique called error guessing.

❖ Types of ad hoc testing

1. Buddy testing

→ Two buddies mutually work on identifying defects in the same module. Mostly one buddy will be from the development team and another person will be from the testing team. Buddy testing helps the tester develop better test cases and development team can also make design changes early. This testing usually happens after unit testing.

2. Pair testing

→ Two testers are assigned modules, share ideas and work on the same machines to find the defects. One person can execute the tests and another person can take notes on the findings. The role of the person can be a tester and scriber during testing.

3. Monkey testing

→ Rendomly test the product or application without test cases with a goal to break the system.

21. What is load testing?

- → Load testing is type of performance testing that determines the performance of the system, software product or software application under real life-based load condition.
- → Load testing determines the behavior of the application when multiple users use it at the same time.
- → The goal of load testing is to identify bottlenecks and determine the maximum number of users transaction of the system can handle.

❖ Load testing tools

1. LoadRunner
2. Web load
3. Astra load test
4. Review’s Web load
5. Studio, Rational site load
6. Silk performer

22. What is stress testing?

- → Stress testing is a software testing technique that determines the robustness of software by testing beyond the limits of normal operation.

→ Stress testing ensure that system failure doesn't cause security issues.

→ Stress testing makes the system work in normal as well as abnormal conditions in an appropriate way.

❖ Stress testing tools

1. JMeter
2. LoadRunner
3. Stress tester
4. Neo load

23. What is exploratory testing?

→ Exploratory testing is a type of software testing in which the tester is free to select any possible methodology to test the software. It is unscripted approach for software testing.

→ Exploratory testing checks the functionality and operations of the software as well as identifying the functional and technical faults in it.

- → The aim of exploratory testing is to optimize and improve the software in every possible way.
- → It finds critical defects very quickly.

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

- → Testing is based on an analysis of the internal structure of the component or system.
- → Testing based upon the structure of the code
- → White box testing is the detailed investigation of internal logic and structure of the code.
- → Testers can identify defects that cannot be detected through other testing techniques.
- → Tester can ensure that the code meets coding standards and is optimized for performance.

❖ Types of white box testing

1. Statement coverage
2. Decision coverage
3. condition coverage

25. Mention what are the categories of defect?

1. Database defect

   → Deale with improper handling of data in the database.

2. Critical functionality defect

   → The occurrence of these bugs hampers the crucial functionality of the application.

3. Functionality defect

   → These defects affect the functionality of the application.

4. Security defect

   → Application security defects generally involve improper handling of data sent from the user to the application.

5. User interface defects

   → As the name suggests, the bugs deal with problems related to UI are usually considered less severe.

26. When should “regression testing” be performed?

- → When the system is stable and the system or the environment changes.
- → When testing bug-fix releases as part of the maintenance phase .
- → It should be applied at all test levels
- → It should be considered complete when agreed completion criteria for regression testing have been met.
- → Regression test suits involve over time and given that they are run frequently are ideal candidates for automation

27. What is the purpose of the exit criteria?

- → Exit criteria is used to determine whether a given test activity has been completed or not.

28. What determines the levels of risk?

- → As a risk is determined by a combination of probability and severity, the main area of the matrix reveals the risk levels. The levels are low, medium, high and extremely high.

29. Difference between priority and severity?

Severity	Priority
→ Severity is parameter to demote the impact of a particular defect on the software.	→ Priority is a parameter to decide the order in which defect should be fixed.
→ Severity means how sever defect is affecting the functionality	→ Priority means how fast defect has to be fixed.
→ Severity is related to the quality standard.	→ Priority is related to resolve the problem.
→ Its value is objective.	→ Its value is subjective.
→ Its value doesn't change from time to time.	→ Its value changes from time to time.

30. What is bug life cycle?

→ **Defect Life Cycle** or Bug Life Cycle in software testing is the specific set of states that defect or bug goes through in its entire life. The purpose of Defect life cycle is to easily coordinate and communicate current status of defect which changes to various assignees and make the defect fixing process systematic and efficient.

31. What is the difference between test scenario, test cases and test script?

1. Test scenario

→Test scenario is the least detailed and high-level type of documentation. A test scenario is a description of an objective a user might face when using the program. They cover an end-to-end functionality which is to be tested.

2. Test case

→A test case is a documented set of preconditions (prerequisites), procedures (inputs/actions) and postconditions (expected results) which a tester uses to determine whether a system under test satisfies requirements or works correctly. Test cases have a great impact on the testing phase.

4. Test script

→A test script is the most detailed way to document software testing. It typically has 'steps' in the form of code that should be performed manually.

32. Explain what test plan is? What is the information that should be covered.

→A **Test Plan** is a detailed document that describes the test strategy, objectives, schedule, estimation, deliverables, and resources required to perform testing for a software product. Test Plan helps us determine the effort needed to validate the quality of the application under test.

33.What is priority?

→ Defect priority is relative and business focused. Priority defines the order in which we should resolve a defect.

34. What is severity?

→ Severity is absolute and customer focused. It is the extent to which the defect can affect the software.

35. Bug categories are

1.Database defect

2.Critical functionality defect

3. Functionality defect

4. Security defects

5.User interface defects

36. Bugzilla advantages

→ It is an open source widely used bug tracker.

→ It is an easy in usage and its user interface is understandable for people without technical knowledge.

→ It easily integrates with test management instruments.

→ It automates documentation.

37. Write test scenario of pen

→ Cap is easily open and close.

→ Between writing pen is not stop.

→ During writing ink is not overflow.

→ During writing pen point is not bend.

→ If pen material can not strong so pen should break easily.

→ During writing pen is properly working.

→ In multicolor pen every color pen can work.

→ Refill can easily open and close.

38. Write test scenario of door

→ Door can easily open and close.

→ During open and close door should not produce noise.

→ During open the door handle should not loose.

→ After open the door can easily hold.

→ While holding magnet is work properly.

→ Door stopper can easily look and unlock.

→ If door with password locking system so locking system should working condition.

→ If door size is different so door can produce noise.

→ If door with sensor so sensor should be working condition.

→ If door is rotating so door can easily rotate.

→ During rotating door should not produce noise.

39.write test scenario of chair

→ Chair having strong material if chair should not strong so after long time chair can break easily

→ If chair is foldable so chair can fold easily

→ During folding chair should not produce noise.

→ If chair with wheels so chair can easily move.

→ During movement wheels should not produce noise.

→ If chair is adjustable so chair can easily adjust.

→ During adjustment should not produce noise.

40.When to used usability testing?

→ Usability testing test before a redesign

→ This might sound silly because you know you're going to do a redesign, but usability testing of the existing product can provide all sorts of ideas for the redesign. It also means that you can identify the biggest pain points of the current offering and work to resolve them.

41. What is the procedure for GUI testing?

42. What are the different methodologies in agile development model?

1.Scrum

→ Scrum is agile development method which concentrates particular on how to manage tasks within a team based development environment.

2. Kanban

→ Kanban is very popular framework for development in the agile software development methodologies.

43. Write test scenario of microwave Owen

→ Verify that the dimensions of the oven are as per the specification provided.

→ Verify that the oven material is optimal for its use as an oven and as per specification.

→ Verify that the oven heats the food at the desired temperature properly.

→ Verify that the ovens functionality with maximum attainable temperature.

→ Verify that the oven plate rotation speed optimal and not too high to spill the food kept over it

44. Write test scenario of coffee vending machine

→ Check if the power button of the coffee vending machine is working correctly after pressing the power button.

→ Check the indicator lights are displaying correctly when the coffee vending machine is going to switch off or on.

→ Check all the buttons of the coffee vending machine have an image text on them, which indicates what task will be performed if you press the button.

→ Check the complete quantity of copy poured in a single operation, and no coffees are stored in the nozzle area.

→ Check the former up; the coffee vending machine is working as expected.

→ Check the temperature of is served coffee should be the same temperature

45. Write test scenario of lift

→ Verify that the lift material is good quality.

→ Verify that the during up and down lift is properly working condition.

→ Verify that the during up and down lift should not produce noise.

→ Verify that the door is auto open or not.

→ Verify that the lift fan is properly working.

→ Verify that the door sensor is properly in working condition.

→ Verify that the ground floor button is properly working.

→ Verify that the basement button is properly working or not.