**MODULE-2**

Que:-1 What is Exploratory Testing?

* Exploratory testing is a concurrent process where
* Test design, execution and logging happen simultaneously
* Testing is often not recorded
* Makes use of experience, heuristics and test patterns
* Testing is based on a test charter that may include
* Scope of the testing (in and out)
* The focus of exploratory testing is more on testing as a “thinking” activity.
* A brief description of how tests will be performed
* Expected problems
* Is carried out in time boxed intervals

Que:-2 what is traceability matrix?

* Make obvious to the client that the software is being developed as per the requirements.
* To make sure that all requirements included in the test cases To make sure that developers are not creating features that no one has requested Easy to identify the missing functionalities.
* If there is a change request for a requirement, then we can easily find out which test cases need to update.
* The completed system may have “Extra” functionality that may have not been specified in the design specification, resulting in wastage of manpower, time and effort.
* No traceability or Incomplete Traceability Results into: Poor or unknown test coverage, more defects found in production It will lead to miss some bugs in earlier test cycles which may arise in later test cycles.
* Then a lot of discussions arguments with other teams and managers before release. Difficult project planning and tracking, misunderstandings between different teams over project dependencies, delays, etc

Que:-3 What is Boundary value testing?

* Boundary Value testing is based on testing the boundary values of valid and invalid partitions. The behaviour at the edge of the equivalence partition is more likely to be incorrect than the behaviour within the partition, so boundaries are an area where testing is likely to yield defects.
* It checks for the input values near the boundary that have a higher chance of error. Every partition has its maximum and minimum values and these maximum and minimum values are the boundary values of a partition

Que:-4 What is Equivalence partitioning testing?

* **Equivalence Partitioning Method** is also known as Equivalence class partitioning (ECP). It is a [software testing](https://www.geeksforgeeks.org/software-testing-basics/) technique or [black-box testing](https://www.geeksforgeeks.org/software-engineering-black-box-testing/) that divides input domain into classes of data, and with the help of these classes of data, test cases can be derived. An ideal test case identifies class of error that might require many arbitrary test cases to be executed before general error is observed.
* In equivalence partitioning, equivalence classes are evaluated for given input conditions. Whenever any input is given, then type of input condition is checked, then for this input conditions, Equivalence class represents or describes set of valid or invalid states.

Que:-5 What is Integration testing?

* Integration Testing - Testing performed to expose defects in the interfaces and in the interactions between integrated components or systems Integration Testing is a level of the software testing process where individual units are combined and tested as a group.
* The purpose of this level of testing is to expose faults in the interaction between integrated units. Test drivers and test stubs are used to assist in Integration Testing.
* Integration testing tests integration or interfaces between components, interactions to different parts of the system such as an operating system, file system and hardware or interfaces between systems.
* Integration testing is done by a specific integration tester or test team.
* Components may be code modules, operating systems, hardware and even complete systems
* There are 2 levels of Integration Testing
* Component Integration Testing
* System Integration Testing

Que:-6 What determines the level of risk?

* As 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. To have a low level of risk, we must have a somewhat limited probability and level of severity.

Que:-7 What is Alpha testing?

* 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 actually tries the product out.
* Alpha Testing is not open to the market and public
* It is conducted for the software application and project.
* It is always performed in Virtual Environment.
* It is always performed within the organization.
* It is the form of Acceptance Testing.
* Alpha Testing is definitely performed and carried out at the developing organizations location with the involvement of developers.

Que:-8 What is beta testing?

* It is always performed by the customers at their own site.
* It is not performed by Independent Testing Team.
* Beta Testing is always open to the market and public.
* It is usually conducted for software product
* It is performed in Real Time Environment.
* It is always performed outside the organization.
* It is also the form of Acceptance Testing.
* Beta Testing (field testing) is performed and carried out by users or you can say people at their own locations and site using customer data.
* It is only a kind of Black Box Testing.

Que:-9 What is component testing?

* Component (Unit) – A minimal software item that can be tested in isolation. It means A unit is the smallest testable part of software.
* Component Testing – The testing of individual software components.
* Unit Testing is a level of the software testing process where individual units/components of a software/system are tested. The purpose is to validate that each unit of the software performs as designed.
* A unit is the smallest testable part of an application like functions/procedures, classes, interfaces.
* The goal of unit testing is to isolate each part of the program and show that the individual parts are correct

Que:-10 What is functional system testing?

* Functional System Testing : A requirement that specifies a function that a system or system component must perform A Requirement may exist as a text document and/or a model
* There is two types of Test Approach Requirement
* Based Functional Testing
* Process Based Testing
* Functional System Testing Functionality As below:

|  |  |
| --- | --- |
| Accuracy | Provision of right or agreed results or effects |
| 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 |

Que:-11 What is Non-Functional Testing?

* Non-Functional Testing: Testing the attributes of a component or system that do not relate to functionality, e.g. reliability, efficiency, usability, interoperability, maintainability and portability
* May be performed at all Test levels (not just Non Functional Systems Testing) Measuring the characteristics of the system/software that can be quantified on a varying scale- e.g. performance test scaling
* Non-functional testing includes, but is not limited to, performance testing, load testing, stress testing, usability testing, maintainability testing, reliability testing and portability testing.

Que:-12 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.

Que:-13 What is Adhoc testing?

* 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.
* In fact is does not create test cases altogether! This testing is primarily performed if the knowledge of testers in the system under test is very high.
* Testers randomly test the application without any test cases or any business requirement document. Adhoc Testing does not follow any structured way of testing and it is randomly done on any part of application.
* Main aim of this testing is to find defects by random checking.
* Adhoc testing can be achieved with the testing technique called Error Guessing. Error guessing can be done by the people having enough experience on the system to “guess” the most likely source of errors.

Que:-14 What is load testing?

* Load time is normally the initial time it takes an application to start.
* This should generally be kept to a minimum. While some applications are impossible to make load in under a minute, Load time should be kept under a few seconds if possible.

Que:-15 What is stress Testing?

* System is stressed beyond its specifications to check how and when it fails.
* Performed under heavy load like putting large number beyond storage capacity, complex database queries, continuous input to system or database load. Stress testing is used to test the stability & reliability of the system.
* This test mainly determines the system on its robustness and error handling under extremely heavy load conditions.

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

* Testing based on an analysis of the internal structure of the component or system.
* Structure-based testing technique is also known as ‘white-box’ or ‘glass-box’ testing technique because here the testers require knowledge of how the software is implemented, how it works.
* Statement coverage
* Decision coverage and Condition coverage

Que:-17 What is black box testing? What are the different black box testing techniques?

* 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.
* Equivalence partitioning
* Boundary value analysis
* Decision tables State transition
* testing Use-case
* Testing Other Black Box Testing

Que:-18 Mention what are the categories of defects?

* Low: The defect is an irritant which should be repaired, but repair can be deferred until after more serious defect has been fixed.
* Medium: The defect should be resolved in the normal course of development activities. It can wait until a new build or version is created.
* High: The defect must be resolved as soon as possible because the defect is affecting the application or the product severely. The system cannot be used until the repair has been done.
* Critical: Extremely urgent, resolve immediately

Que:-19 Mention what big bang testing is?

* In Big Bang integration testing all components or modules is integrated simultaneously, after which everything is tested as a whole.
* Big Bang testing has the advantage that everything is finished before integration testing starts.
* The major disadvantage is that in general it is time consuming and difficult to trace the cause of failures because of this late integration.
* Here all component are integrated together at once, and then tested.

Que:-20 What is the purpose of exit criteria?

* Approach: Defining the overall approach of testing (the test strategy), including the definition of the test levels and entry and exit criteria.
* Integrating and coordinating the testing activities into the software life cycle activities: acquisition, supply, development, operation and maintenance.

Que:-21 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 suites evolve over time and given that they are run frequently are ideal candidates for automation

Que:-22 What is 7 key principles? Explain in detail?

General Testing Principles

1. Testing shows presence of Defects

2. Exhaustive Testing is Impossible!

3. Early Testing

4. Defect Clustering

5. The Pesticide Paradox

6. Testing is Context Dependent

7. Absence of Errors Fallacy

Que:-23 Difference between QA v/s QC v/s Tester

|  |  |  |  |
| --- | --- | --- | --- |
| S.N | Quality Assurance | Quality Control | Tester |
| 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 procedures rather than conducting actual testing on the system. | Focuses on actual testing by executing Software with intend to identify bug/defect through implementation of procedures and process. | Focuses on actual testing. |
| 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. |

Que:-24 Difference between Smoke and Sanity?

* Smoke Testing Smoke
* Testing is performed after software build to ascertain that the critical functionalities of the program is working fine. It is executed "before" any detailed functional or regression tests are executed on the software build.
* The purpose is to reject a badly broken application, so that the QA team does not waste time installing and testing the software application.
* In Smoke Testing, the test cases chosen cover the most important functionality or component of the system.
* The objective is not to perform exhaustive testing, but to verify that the critical functionalities of the system are working fine. For Example a typical smoke test would be – Verify that the application launches successfully, Check that the GUI is responsive ... etc.
* Sanity Testing
* After receiving a software build, with minor changes in code, or functionality, Sanity testing is performed to ascertain that the bugs have been fixed and no further issues are introduced due to these changes.
* The goal is to determine that the proposed functionality works roughly as expected.
* If sanity test fails, the build is rejected to save the time and costs involved in a more rigorous testing.
* The objective is "not" to verify thoroughly the new functionality, but to determine that the developer has applied some rationality (sanity) while producing the software. For instance, if you’re scientific calculator gives the result of 2 + 2 =5! Then, there is no point testing the advanced functionalities like sin 30 + cos 50.

Que:-25 Difference between verification and Validation

* Verification
* Architectural Design (Technical Specification): Architectural specifications are understood and designed in this phase. Usually more than one technical approach is proposed and based on the technical and financial feasibility the final decision is taken. System design is broken down further into modules taking up different functionality.
* This is also referred to as High Level Design (HLD). The data transfer and communication between the internal modules and with the outside world (other systems) is clearly understood and defined in this stage. With this information, integration tests can be designed and documented during this stage.
* Module Design (Program Specification): In this phase the detailed internal design for all the system modules is specified, referred to as Low Level Design (LLD). It is important that the design is compatible with the other modules in the system architecture and the other external systems. Unit tests are an essential part of any development process and helps eliminate the maximum faults and errors at a very early stage. Unit tests can be designed at this stage based on the internal module designs.
* Validation
* Unit Testing: Unit tests designed in the module design phase are executed on the code during this validation phase. Unit testing is the testing at code level and helps eliminate bugs at an early stage, though all defects cannot be uncovered by unit testing.
* Integration Testing: Integration testing is associated with the architectural design phase. Integration tests are performed to test the coexistence and communication of the internal modules within the system.
* System Testing: System testing is directly associated with the System design phase. System tests check the entire system functionality and the communication of the system under development with external systems. Most of the software and hardware compatibility issues can be uncovered during system test execution.
* Acceptance Testing: Acceptance testing is associated with the business requirement analysis phase and involves testing the product in user environment. Acceptance tests uncover the compatibility issues with the other systems available in the user environment. It also discovers the non-functional issues such as load and performance defects in the actual user environment.

Que:-26 Explain types of Performance testing.

* Software performance testing is a means of quality assurance (QA). It involves testing software applications to ensure they will perform well under their expected workload.
* Features and Functionality supported by a software system is not the only concern. A software application’s performance like its response time, do matter.
* The goal of performance testing is not to find bugs but to eliminate performance bottlenecks The focus of Performance testing is checking a software programs
* Speed – Determines whether the application responds quickly
* Scalability – Determines maximum user load the software application can handle.
* Stability – Determines if the application is stable under varying loads

Que:-27 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”
* 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.
* Defect: Commonly refers to several troubles with the software products, with its external behavior or with its internal features.
* Failure: The inability of a system or component to perform its required functions within specified performance requirements. See: bug, crash, exception, and fault.

Que:-28 Difference between Priority and Severity

* High Priority & High Severity: An error which occurs on the basic functionality of the application and will not allow the user to use the system. (Eg. A site maintaining the student details, on saving record if it, doesn’t allow to save the record then this is high priority and high severity bug.)
* High Priority & Low Severity: The spelling mistakes that happens on the cover page or heading or title of an application.
* High Severity & Low Priority: An error which occurs on the functionality of the application (for which there is no workaround) and will not allow the user to use the system but on click of link which is rarely used by the end user.
* Low Priority and Low Severity: Any cosmetic or spelling issues which is within a paragraph or in the report (Not on cover page, heading, title).

Que:-29 What is Bug Life Cycle?

* As you can see from above diagram, a defect‘s state can be divided into Open or Closed. When a bug reaches one of the Closed or Terminal states, its lifecycle ends.
* Each state has one or more valid states to move to. This is to ensure that all necessary steps are taken to resolve or investigate that defect.
* For example, a bug should not move from Submitted state to resolved state without having it open. In a typical scenario, as soon as a bug is identified, it is logged into the bug tracking system with status as Submitted.
* After ascertaining the validity of the defect, it is given the “Open” Status.

Que:-30 Explain the difference between Functional testing and NonFunctional testing.

|  |  |
| --- | --- |
| Functional Testing | Non-Functional Testing |
| Functional testing is performed using the Non-Functional testingchecksthe Performance, functional specification provided by the client and reliability, scalability and other non-functional aspects verifies the system against the functional requirements. | testingchecksthe Performance, functional specification provided by the client and reliability, scalability and other non-functional aspects verifies the system against the functional requirements. of the software system. |
| Functional testing is executed first | Non functional testing should be performed after functional testin |
| Manual testing or automation tools can be used for functional testing | Using tools will be effective for this testing |
| Business requirements are the inputs to functional testing | Performance parameters like speed , scalability are inputs to non-functional testing. |
| Functional testing describes what the product does | Nonfunctional testing describes how good the product works |
| Easy to do manual testing | Tough to do manual testing |
| Types of Functional testing are  ∙ Unit Testing  ∙ Smoke Testing  ∙ Sanity Testing  ∙ Integration Testing  ∙ White box testing  ∙ Black Box testing  ∙ User Acceptance testing  ∙ Regression Testing | Types of Nonfunctional testing are  ∙ Performance Testing  ∙ Load Testing  ∙ Volume Testing  ∙ Stress Testing  ∙ Security Testing  ∙ Installation Testing  ∙ Penetration Testing  ∙ Compatibility Testing  ∙ Migration Testing |

Que:-31 What is the difference between the STLC (Software Testing Life Cycle) and SDLC (Software Development Life Cycle)?

STLC

In an Ideal world, you will not enter the next stage until the exit criteria for the previous stage is met. But practically this is not always possible.

SDLC

: Testing is a process that’s take place throughout the Software Development Life Cycle (SDLC).

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

* **Test Scenarios:**  A Test Scenario is any functionality that can be tested. It is also called Test Condition or Test Possibility.
* **Test Cases:**  It is a document that contains the steps that has to be executed, it has been planned earlier.
* Top of Form
* **Test Script:**It is written in a programming language and it's a short program used to test part of functionality of the software system. In other words a written set of steps that should be performed manually.

Que:-33 Explain what Test Plan is? What is the information that should be covered?

1. What is a Test Plan? A Test Plan is a detailed document that catalogs the test strategies, objectives, schedule, estimations, deadlines, and resources required to complete that project. Think of it as a blueprint for running the tests needed to ensure the software is working correctly – controlled by test managers.What is a Test Plan? A Test Plan is a detailed document that catalogs the test strategies, objectives, schedule, estimations, deadlines, and resources required to complete that project. Think of it as a blueprint for running the tests needed to ensure the software is working correctly – controlled by test managers.What is a Test Plan? A Test Plan is a detailed document that catalogs the test strategies, objectives, schedule, estimations, deadlines, and resources required to complete that project. Think of it as a blueprint for running the tests needed to ensure the software is working correctly – controlled by test managers.What is a Test Plan? A Test Plan is a detailed document that catalogs the test strategies, objectives, schedule, estimations, deadlines, and resources required to complete that project. Think of it as a blueprint for running the tests needed to ensure the software is working correctly – controlled by test managers.What is a Test Plan? A Test Plan is a detailed document that catalogs the test strategies, objectives, schedule, estimations, deadlines, and resources required to complete that project. Think of it as a blueprint for running the tests needed to ensure the software is working correctly – controlled by test managers.Bottom of Form

* What is a Test Plan? A Test Plan is a detailed document that catalogs the test strategies, objectives, schedule, estimations, deadlines, and resources required to complete that project. Think of it as a blueprint for running the tests needed to ensure the software is working correctly – controlled by test managers.
* Product Analysis
* Designing Test Strategy
* Defining Objectives
* Establish Test Criteria
* Planning Resource Allocation
* Planning Setup of Test Environment
* Determine test schedule and estimation
* Establish Test Deliverables

Que:-34 What is 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 has to fix it at the earliest. The priority status is set based on the customer requirements.

Que:-35 what is 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.

Que:-36 Bug categories are…

* :Security, Database, Functionality (Critical/General), UI

Que:-37 Advantage of Bugzila .

* 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 withtest management instruments;
* it integrates with an e-mailing system;
* it automates documentation.

Que:-38 Difference between priority and severity

* Severity Means the severity of a defect the tester decides. After analyzing a defect’s impact on the application’s functions, the defect severity can be categorized as
* Severity means the urgency with which the defect needs to be fixed by the development team. Priority, however, answers how quickly the defect needs to be fixed.

Que:-39 What are the different Methodologies in Agile Development Model?.

* Agile SDLC model is a combination of iterative and incremental process models with focus on process adaptability and customer satisfaction by rapid delivery of working software product.
* Agile Methods break the product into small incremental builds.
* These builds are provided in iterations. Each iteration typically lasts from about one to three weeks.
* Every iteration involves cross functional teams working simultaneously on various areas like planning, requirements analysis, design, coding, unit testing, and acceptance testing.
* At the end of the iteration a working product is displayed to the customer and important stakeholders.

Que:-40 When to used Usability Testing?

* Before Any Design Decisions Are Made.
* When It's Time to Evaluate and Iterate.
* After Launch.
* In High-Risk, Low-Certainty Situations.

Que:-41 What is the procedure for GUI Testing?

* Check all the GUI elements for size, position, width, length and acceptance of characters or numbers
* For instance, you must be able to provide inputs to the input fields
* Check you can execute the intended functionality of the application using the GUI
* Check Error Messages are displayed correctly
* Check for Clear demarcation of different sections on screen
* Check Font used in application is
* readable Check the alignment of the text is proper Check the Color of the font and warning messages is
* aesthetically pleasing Check that the images have good clarity Check that the images are properly aligned
* Check the positioning of GUI elements for different screen resolution.

Que:-42 Explain the difference between Authorization and Authentication in Web testing.What are the common problems faced in Web testing?

Authentication

* Authentication verifies who the user is.
* Authentication works through [passwords](https://www.sailpoint.com/products/password-management/), 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.