1.What is Exploratory Testing?

A.

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

->. 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

->. More structured than Error guessing

2.What is traceability matrix?

A. Test conditions should be able to be linked back to their sources in the test basis, this is known as traceability

->. To protect against changes you should be able to trace back from every System component to the original requirement that caused its presence.

3.What is Boundary value testing?

A. Boundary value testing 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.

->. A boundary value for a valid partition is a valid boundary value.

4.What is Equivalence partitioning testing?

A.

->. Aim is to treat groups of inputs as equivalent and to select one representative input to test the mall

->. EP can be used for all Levels of Testing

->. Equivalence partitioning is the process of defining the optimum number of tests by

->. Reviewing documents such as the Functional Design Specification and Detailed Design Specification, and identifying each input condition with in a function

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 tests tubs are used to assist in Integration Testing.

->. Integration testing tests integration or interfaces between 181 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

6.What determines the level of risk?

A.

->. 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.

7.What is Alpha testing?

A.

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

->. Sometimes it is also performed by Independent Testing Team.

->. 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 organization’s location with the involvement of developers.

8.What is beta testing?

A.

->. 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.

->. It is only a kind of Black Box Testing.

->. Beta Testing is always performed at the time when software product and project are marketed.

->. It is always performed at the user’s premises in the absence of the development team.

9.What is component testing?

A.

Component testing, also known as program or module testing, is done after unit testing.

->. In this type of testing those test objects can be tested independently as a component without integrating with other components

e.g. modules, classes, objects, and programs. This testing is done by the development team.

10.What is functional system testing?

A. Functional Testing: Testing based on an analysis of the specification of the functionality of a component or system.

‘Specification’ – E.g. Requirements specification, Use Cases, functional specification or maybe undocumented.

->. ‘Function’ – what the system does Considers the external (not internal) behaviour of the software. Black- Box testing. What it does rather than how it does it.

->. Functional testing verifies that each function of the software application operates in conformance with the requirement specification.

->. This testing mainly involves black box testing and it is not concerned about the source code of the application.

->. Each & every functionality of the system is tested by providing appropriate input, verifying the output and comparing the actual results with the expected results.

->. This testing involves checking of User Interface, APIs, Database, security, client/ server applications and functionality of the Application under Test. The testing can be done either manually or using automation

11.What is Non-Functional Testing?

A.

->. 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 tests calling.

->. Non-functional testing includes, but is not limited to, performance testing, load testing, stress testing, usability testing, maintainability testing, reliability testing and portability testing.

->. It is the testing of “how” the system works. Non-functional testing maybe performed at all test levels.

->. The term non-functional testing describes the tests required to measure characteristics of systems and software that can be quantified on a varying scale, such as response times for performance testing.

->. To address this issue, performance testing is carried out to check & fine tune system response times. The goal of performance testing is to reduce response time to an acceptable level

->. Hence load testing is carried out to check systems performance at different loads i.e. number of users accessing the system

12.What is GUI Testing?

A.

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.

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 section screen

->. Check Font used in application is readable

->. Check the alignment of the text is proper

->. Check the Colour of the front and warning messages is aesthetically pleasing

13.What is Adhoc testing?

A.

->. 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

->. Any business requirement document.

->. Adhoc Testing does not follow any structured way of testing and it is randomly done on any part of application.

14.What is load testing?

A.

Load testing is about creating production simulations within an application or system that is as near as possible to being a finished product ready to deploy and subject to the masses. By utilizing specialized testing software.

->. load testing allows dev teams to answer questions like “is my system doing what I expect under these conditions?” and “is its performance good enough?” as the Microsoft guide performance testing guidance for web applications states

->. Load testing is the process of putting simulated demand on software, an application or website in a way that tests or demonstrates it’s behaviour under various conditions.

15.What is stress Testing?

A.

Stress testing is a type of software testing that verifies stability & reliability of software application.

->. The goal of stress testing is measuring software on its robustness and error handling capabilities under extremely heavy load conditions. And ensuring that software doesn’t crash under crunch situations.

->. It even tests beyond normal operating points and evaluates how software works under extreme conditions.

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

A.

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.

->. In white-box testing the tester is concentrating on how the software does it.

->. For example, a structural technique may be concerned with exercising loops in the software.

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

A.

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.

Techniques of Black Box Testing There are four specification-based or black-box

->. Equivalence partitioning

->. Boundary value analysis

->. Decision tables

->. State transition testing

18.Mention what are the categories of defects?

A

The defects into three main categories minor, major and critical. The nature and severity of a defect.

Minor defect

Major defect

Critical defect

1.Minor defect: - Minor defects are usually small, insignificant issues that don’t affect the function or form of the item.

->. In most cases, the customer would not even notice a minor defect on product. And the customer wouldn’t likely return an item due to a minor defect alone.

2. Major defect: - Major defects are more serious than minor defects. A product with a major defect departs significantly from the buyer’s product specification.

->. Major defects are those which could adversely affect the function, performance or appearance of a product.

19.Mention what big bang testing is?

A.

->. 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.

20.What is the purpose of exit criteria?

A.

Successful Testing of Integrated Application.

->. Executed Test Cases are documented

->. All High prioritized bugs fixed and closed

->. Technical documents to be submitted followed by release Notes.

->. Any condition not specified in integration tests, apart from the confirmation of the execution of the design items is usually not tested.

21.When should "Regression Testing" be performed?

A.

Regression Testing: Testing of a previously tested program following modification to ensure that defects have not been introduced or uncovered in unchanged areas of the software, as a result of the changes made. It is performed when the software or its environment is changed.

->. If the test is re-run and passes you cannot necessarily say the fault has been resolved because.

->. You also need to ensure that the modifications have not caused unintended side-effects elsewhere and that the modified system still meets its requirements Regression Testing

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

A.

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

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

A.

|  |  |  |
| --- | --- | --- |
| Quality Analyst | Quality control | Tester |
| 1. QA includes activities that ensure the implementation of processes, procedures and standards in context to verification of developed software and intended requirements.  2. ->. Focuses on processes and procedures rather than conducting actual testing on the system.  3. Process-oriented activities | 1. It includes activities that ensure the verification of a developed software with respect to documented requirements.  2. It includes activities that ensure the verification of a developed software with respect to documented requirements.  3. Product-oriented activities. | 1. It includes activities that ensure the identification of bugs in a software.  2. Focuses on actual testing.  3. Product-oriented activities. |

24.Difference between Smoke and Sanity?

A.

|  |  |
| --- | --- |
| Smoke | Sanity |
| 1.Smoke Testing is performed after software build to as certain that the critical functionalities of the program is working fine.  2.It is executed "before" any detailed functional or regression tests are executed on the software build.  3.The purpose is to reject a badly broken application, so that the QA team does not waste time installing and testing the software application.  4.In Smoke Testing, the test cases chosen cover the most important functionality or component of the system.  5.The objective is not to perform exhaustive testing, but to verify that the critical functionalities of the system are working fine.  6.For Example, a typical smoke test would be – Verify that the application launches successfully, Check that the GUI is responsive etc | 1. 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.  2.The goal is to determine that the proposed functionality works roughly as expected.  3.If sanity test fails, the build is rejected to save the time and costs involved in a more rigorous testing.  4.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.  5.For instance, if you’re scientific calculator gives the result of 2 + 2 =5!  6.Then, there is no point testing the advanced functionalities like sin 30 + cos 50. |

25.Difference between verification and Validation?

A.

|  |  |
| --- | --- |
| Verification | Validation |
| 1.The process of evaluating work-products of a development phase to determine whether they meet the specified requirements for that phase.  2. 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.  3. Plans, Requirement Specs, Design Specs, Code, Test Cases | 1. The process of evaluating software during or at the end of the development process to determine whether it satisfies specified business requirements.  2. 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 full-fills its intended use when placed in its intended environment.  3. The actual software. |

26.Explain types of Performance testing?

A.

• Load testing

• Stress testing

• Endurance testing

• Spike testing

• Volume testing

• Scalability testing

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

A.

1.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

2.Defect: Commonly refers to several troubles with the software products, with its external behaviour or with its internal features.

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.Failure: The inability of a system or component to perform its required functions within specified performance requirements. See: bug, crash, exception, and fault.

28.Difference between Priority and Severity?

A.

|  |  |
| --- | --- |
| Priority | Severity |
| 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.  For example: If the company name is misspelled in the homepage of the website, then the priority is high and severity is low to fix it | 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.  For example: If an application or web page crashes when a remote link is clicked, in this case clicking the remote link by an user is rare but the impact of application crashing is severe.  So the severity is high but priority is low. |

29.What is Bug Life Cycle?

A.

“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’.

->. When a bug is discovered, it goes through several states and eventually reaches one of the terminal states, where it becomes inactive and closed.

->. The process by which the defect moves through the life cycle is depicted next slide.

30.Explain the difference between Functional testing and Non Functional testing?

|  |  |
| --- | --- |
| Functional testing | Non-Functional testing |
| 1.Functional testing is performed using the functional specification provided by the client and verifies the system against the functional requirements.  2.Functional testing is executed first.  3.Manual testing or automation tools can be used for functional testing.  4.Types of functional testing  1.Unit testing  2.Smoke testing  3.Sanity testing  4.Integration testing  5.White box testing  6.Black box testing | 1.Non-functional testing checks the performance, reliability, scalability and other non-functional aspects of the software system.  2.Non-functional testing should be performed after functional testing.  3.Using tools will be effective for this testing.  4.Types of non-functional testing  1.Performance testing  2.Load testing  3.Volome testing  4.Stress testing  5.Security testing  6.Installation testing |

• To create HLR & Test Case of 1) (Instagram, Facebook) only first page

2) Facebook Login Page : <https://www.facebook.com/>

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

|  |  |
| --- | --- |
| STLC | SDLC |
| 1.STLC is mainly related to software testing.  2.It focuses only on testing the software.  3.STLC involves only five phases or steps.  4.In STLC, testing team makes the plans and designs.  5.Goal of STLC to complete successful testing of software. | 1.SDLC is mainly related to software development.    2.Besides development other phases like testing is also included.  3.SDLC involves total six phases or steps.  4.In SDLC, more number of members are required for the whole process.  5.In SDLC, development team makes the plans and designs based on the requirements. |

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

|  |  |  |
| --- | --- | --- |
| Test scenario | Test case | Test script |
| 1.Is any functionality that can be tested.  2.Is derived from test artifacts like (b.r.s.) and (s.r.s.).  3.Helps test the end-to-end functionality in an agile way.  4.Is more focused on what to test.  5.Takes less time and fewer resources to create. | 1.Is a set of actions executed to verify particular features or functionality.  2.Is mostly derived from test scenarios.  3.Helps in exhaustive testing of an app.  4.Is focused on what to test and how to test.  5.Requires more resources and time. | 1.Is a set of instructions to test an app automatically.  2.Is mostly derived test cases.  3.Helps to test specific things repeatedly.  4.Is focused on the expected result.  5.Requires less time for testing but more resources for scripts creating and updating. |

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

a.

Test Planning in STLC is a phase in which a Senior QA manager determines the test plan strategy along with efforts and cost estimates for the project.

->. Moreover, the resources, test environment, test limitations and the testing schedule are also determined.

->. The Test Plan gets prepared and finalized in the same phase

Preparation of test plan/strategy document for various types of testing.

->. Test tool selection

->. Test effort estimation

->. Resource planning and determining roles and responsibilities.

->. Training requirement

->. Deliverables of Requirement Phase Testing

->. Test plan document.

• What is priority?

Priority is defined as the order in which the defects should be resolved.

->. The priority status is usually set by the testing team while raising the defect against the dev team mentioning the time frame to fix the defect.

->. The priority status is set based on end users requirement.

• What is severity?

->. Priority means how soon the bug should be fixed.

->. The test engineer determines the severity level of the defect.

->. Priority of defects is decided in discussion with the manager.

->. It is driven by functionality.

• Bug categories are…

1. Performance Bugs

2. Security Bugs

3. Unit Level Bugs

4. Functional Bugs

5. Usability Bugs

6. Logic Bugs

• Advantage of Bug zila.

->. Open source, free bug tracking tool.

->. Automatic Duplicate Bug Detection.

->. Search option with advanced features.

->. File/Modify Bugs By Email.

->. Move Bugs Between Installs.

->. Multiple Authentication Methods (LDAP, Apache server).

->. Time Tracking.

->. Automated bug reporting; has an API to interact with system.

• Difference between priority and severity

|  |  |
| --- | --- |
| priority | severity |
| Priority is a term that defines how fast we need to fix a defect.  Priority is basically a parameter that decides the order in which we should fix the defects.  Priority relates to the scheduling of defects to resolve them in software.  The value of priority is subjective.  The value of Priority changes from time to time. | Severity is a term that denotes how severely a defect can affect the functionality of the software.  Severity is basically a parameter that denotes the total impact of a given defect on any software.  Severity relates to the standards of quality.  The value of severity is objective.  The value of Severity changes continually from time to time. |

• What are the different Methodologies in Agile Development Model?

Agile refers to the methods and best practices for organizing projects based on the values and principles documented in the Agile Manifesto.

However, there’s no one right way to implement Agile and many different types of methodologies from which to choose. Here are some of the most common Agile frameworks.

1. Kanban

Kanban is a simple, visual means of managing projects that enables teams to see the progress so far and what’s coming up next.

Kanban projects are primarily managed through a Kanban board, which segments tasks into three columns: “To Do,” “Doing,” and “Done.”

1. Scrum

Scrum is similar to Kanban in many ways. Scrum typically uses a Scrum board, similar to a Kanban board, and groups tasks into columns based on progress.

Unlike Kanban, Scrum focuses on breaking a project down into sprints and only planning and managing one sprint at a time. Scrum also has unique project roles: Scrum master and product owner.

1. Extreme Programming (XP)

Extreme Programming (XP) was designed for Agile software development projects.

It focuses on continuous development and customer delivery and uses intervals or sprints, similar to a Scrum methodology. However, XP also has 12 supporting processes specific to the world of software development:

1. Planning game
2. Small releases
3. Customer acceptance tests
4. Simple design
5. Pair programming
6. Test-driven development
7. Refactoring
8. Continuous integration
9. Collective code ownership
10. Coding standards
11. Metaphor
12. Sustainable pace

• Explain the difference between Authorization and Authentication in Web testing.

|  |  |
| --- | --- |
| Authorization | Authentication |
| 1.Authorization determines what resources a user can access.  2.Authorization works through settings that are implemented and  maintained by the organization.  3.Authorization always takes place after authentication.  4.Authorization isn’t visible to or changeable by the user.  5.Example: Once their level of access is authorized, employees and  HR managers can access different levels of data based on the permissions  set by the organization.. | 1.Authentication verifies who the user is.  2.Authentication works through passwords, one-time pins,  biometric information, and other information provided or entered by the user.  3.Authentication is the first step of a good identity and access management process.  4.Authentication is visible to and partially changeable by the user.  5.Example: By verifying their identity, employees can gain access to an  HR application that includes their personal pay information, vacation time, and 401K data. |

What are the common problems faced in Web testing?

a.

1. Insufficient testing for browser compatibility

2. Failing to conduct thorough functional testing across mobile

3. Failing to conduct thorough functional testing across desktop

4. Poor data security

5. Failing to provide an intuitive experience

6. Not performing testing frequently enough

7. Leaving digital accessibility to the last minute

8. Releasing new features breaks the existing live system

9. Localisation and the global experience

10. The most common bugs

• To create HLR & Test Case of Web Based (WhatsApp web, Instagram)

WhatsApp Web: <https://web.whatsapp.com/>

• Write a scenario of only Whatsapp chat messages

• Write a Scenario of Pen

• Write a Scenario of Pen Stand

• Write a Scenario of Door

• Write a Scenario of ATM

• When to used Usablity Testing?

a. If possible, usability testing can and should be conducted on the current

->. iteration of a product before beginning any new design work, after you’ve begun the strategy work around a brand new site or app. This will quickly

->. identify areas for opportunity, and reduce the amount of assumptions

->. design team will make with regard to what the user wants. Additionally, after the usability tests analysis, the team should have the ability to pin

->. point the steps needed to achieve the project goals with as little disruption as possible.

• What is the procedure for GUI Testing?

a.

1. Testing the size, position, height, width of the visual elements.
2. Verifying and testing the error messages are displayed or not.
3. Testing different sections of the display screen.
4. Verifying the usability of carousel arrows.
5. Checking the navigation elements at the top of the page

• Write a scenario of Microwave Owen

• Write a scenario of Coffee vending Machine

• Write a scenario of chair

• To Create Scenario (Positive & Negative)

facebook Chat on Mobile

•Write a Scenario of Wrist Watch

• Write a Scenario of Lift(Elevator)

• Write a Scenario of whatsapp Group (generate group)

• Write a Scenario of instagram ( video call with chat )

• Write a Scenario of Whatsapp payment