1. What is Exploratory Testing?

A. Though the current trend in testing is to push for automation exploratory testing is a new way of thinking Automation has its limits.

Is not a technique but it is an approach. What action you perform next is governed by what you are doing currently.

Is cognitively structured as structured as compared to the procedural structure of scripted testing. This structure comes from charter time boxing etc.

2. What is traceability matrix?

A. Make obvious to the client that the software is being developed as per the requirements.

The completed system may have “Extra” functionality that may have not been specified in the design specification, resulting in waste of manpower, time, and effort.

It will lead to miss some bugs in earlier test cycle which may arise in the later test cycle. Then a lot of discussions and arguments with other teams and managers before release.

3. What is Boundary value testing?

A. Boundary value analysis is a methodology for designing test cases that concentrates software testing effort on cases near the limits of valid ranges

Boundary value analysis is a method which refines equivalence partitioning

At those points when input values change from valid to invalid errors are most likely to occur.

Boundary value Analysis uses the same analysis of partitions as EP and is usually used in conjunction with EP in test case design

4. What is Equivalence partitioning testing?

A. Equivalence partitioning is technique that divides the input domain of a system into partitions or classes that are expected to produce the same output or behavior. For example, if a system accepts an integer between 1 and 100 as input, you can create four partitions: 1-10, 11-50, 51-99 and 100. Each partition represents a set of equivalent values that should trigger the same response from the system. You can select one value from each partition as a test case, rather than testing all 100 values.

5 . What is integration testing?

A. Integration Testing – Testing performed to expose defects in the interfaces and in the interactions between integration 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

6. What determines the level of risk?

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

Moreover, the resources, test environment, test limitation, and the testing schedule are also determined.

7. What is Alpha testing?

A. It is always performed by the developer at the software development site

Sometimes it is also performed by the 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 a virtual environment.

Alpha Testing is definitely performed and carried out at the developing organization 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.

Beta Testing is performed and carried out by users or you can say people at their own locations and site using customer data.

9. What is component testing?

A. component – 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.

10. What is functional system testing?

A. 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

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

Measuring the characteristics of the system /software that can be quantified on varying scale – e.g. performance test scaling.

12. What is GUI Testing?

A. Graphical user interface testing is process of testing the system’s GUI of the System under testing involves checking the screens with the controls like menus, buttons, icons, and all type of bars – toolbar menu bar, dialog boxes and windows etc.

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 fields.

13. What is adhoc testing?

A. Adhoc testing is an informal testing type with an aim to break the system.

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 the application

Adhoc testing can be achieved with the testing technique called Error Guessing.

14. What is load testing?

A. Load testing – It’s performance testing to check system behavior under load testing an application under heavy loads , such as testing of a web site under a range of loads to determine at what point the system’s response time degrades or fails.

Load testing is a kind of performance testing which determines a system’s performance under real -life load conditions. This testing helps determine how the application behaves when multiple users access it simultaneously.

15. What is stess testing?

A. 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, and continuous input to system or database load.

Stess 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.

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

A. 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 tester require knowledge of how the software is implemented, how the software is implemented, how it works.

* Statement coverage
* Decision coverage
* Condition coverage

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 know as ‘black-box’ or input/output driven testing techniques because the view the software as a black-box with inputs and outputs.

* There are four specification-based or black box tecbique.
* Equivalence partitioning
* Boundary value analysis
* Decision tables
* State transition testing
* Use case Testing
* Other Black box testing

18. Mention what are the categories of defects?

A. Defect is the variance from a desired product attribute (it can be a wrong missing or extra data).

* With the knowledge of testing so far gained, you can now be able to categorize the defects you have found.
* Defects can be categorized into different types basing on the core issues the address.
* Some defects address security or database issues while other may refer to functionality or UI issues.

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 .

Bing 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.

Since all module are tested at once, high risk critical modules are isolated and tested on priority, peripheral modules which deal with user interfaces are also not isolated and tested on priority.

20. What is the purpose of exit criteria?

A. Purpose of exit criteria is to define when we STOP testing either at the:

* End of all testing – i.e. product Go live
* End of phase of testing
* Thoroughness mesures, such as coverge of requirement or of code or risk coverage.
* Estimates of defect density reliability measure.
* Residual Ris, such as defects not fixed or lack of test coverage in certain areas.

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.

You also need to ensure that the modification have not caused unintended side effects elsewhere ad tht the modified system still meets its requirements- Regressing testing.

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

A. 1. Testing shows presence of Defects

* Testing can show hat 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.

2. Exhaustive Testing is impossible!

* Testing everything including all combinations of inputs and preconditions is not possible.
* So, instead of doing exhaustive testing we can use risk and priorities to focus testing efforts.

3. Early Testing

* Testing activities should start as early as possible I the software or system development life cycle, and should be focused on defined objectives.

4. Defect clustering

* A small number of models contain most of the defects discovered during pre -release testing, or are responsible for the most operational failures.

5. The Pesticide Paradox

* Therefore we must learn, create and use new tests based on new techniques to catch new bugs

6. Testing is Context Dependent

* Testing is basically context-dependent
* Testing is done differently in different contexts
* Different kinds of sites are tested differently

7. Absence of errors fallacy

* If the system built is unusable and does not fulfill the user’s neds and expectations then finding and fixing defects does not help.

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

A.

|  |  |  |  |
| --- | --- | --- | --- |
| S.N | 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 requirements. | Activities which ensure the identification of bug/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 bugs/defect through implementation of proedures 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 live cycle | QC cab be considered as the subset of Quality Assurance. | Testing is the subset of Quality Control |

24. Difference between Smoke and Sanity?

A. Smoke and Sanity testing are the most misunderstood topics in Software Testing. There is enormous amount of literature on the subjects, but most of them are confusing. The following article makes an attempt to address the confusion.

* If you are developing a simple computer program which consists of only one source code file, you merely need to compile and link this one file to produce cod file , you merely need to compile and link this one file, to produce an executable file. This process is very simple.

25. Difference between verification and validation

A.

|  |  |  |
| --- | --- | --- |
| Criteria | Verification | Validation |
| Definition | The process of evaluating work – products of a development phase to determine whether they meet the specified requirement 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 build according to the requirement and design specification | To ensure that the product actually meets the use’s needs, and that the specifications were correct in the first place In other words, to demonstrate that the product fulfills its intended use when placed in 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.

A. 1. Load testing

* It’s performance testing to check system behavior under load testing an application under heavy loads, such as testing of a website under a range of loads to determine at what point the system’s response time degrades or fails.

2. Stress testing

* System is stressed beyond its specification 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.

3. Endurance testing

* Endurance testing, also known as soak testing, is a sort of performance testing used to assess how a system performs under prolonged stress. This entails giving your system a lot of work and watching to see if it slows down or crashes.

4. Spike Testing

* Spike testing is a type of performance testing that involves flooding a site or application with sudden and extreme increases and decreases in load. Let’s dive deeper into this intriguing technique.

5. Volume Testing

* Volume testing is a type of performance testing that checks how a system or application performs when subjected to a large amount of data.

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

A. “ A mistake in coding is called an 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. Failure

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

2. 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 a terminology of tester.

3. Fault

* An incorrect step, process, or data definition in a computer program which causes the program to perform in an unintended or unanticipated manner. See: bug defect, error , exception.

4. Defect

* Commonly refers to several troubles with the software product, with its external behavior or with its internal features.

28. Difference between Priority and Severity

A. Severity is basically a parameter that denotes the total impact of a given defect on any software. Priority is basically a parameter that decides the order in which we should fix the defects. Severity relates to the standards of quality . Priority relates to the scheduling of defects to resolve them in software.

29. What is bug life cycle?

A. “ A computer bug is error, flaw, mistake, flaw, mistake, failure, failure or fault in a computer program that prevents it from working correctly or produces an incorrect result. Bugs arise from mistakes and errors, 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 and eventually reaches one of the terminal states , where it becomes inactive and closed.

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

A. Functional testing

* 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.

Non – Functional Testing

* It is the testing of “how” the system works. Non -functional testing may be 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 time for performance testing.
* Hence load testing is carried out to check systems performance at different load i.e. number of users accessing the system.

31. What is the difference between the STLC (Software Testing Life Cycle ) and SDLC (Software Develop life Cycle )

A.

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

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

A.

|  |  |  |
| --- | --- | --- |
| Test Scenario | Test Case | Test Script |
| Is any functionality that ca be tested. | Is a set of actions executed to verify particular features or functionality. | Is a set of instructions to test an app automatically |
| Is derived from test artifacts like Business Requirement Specification (BRS) and Software Requirements Specification (SRS). | Is mostly derived from test scenarios. | Is mostly derived from test cases. |
| Helps test the end-to-end functionality in an Agile way. | Help in exhaustive testing of an app. | Helps to test specific things repeatedly. |
| Is more foused on what to test. | Is focused on what to test and how to test | Is focused on the expected result. |
| Take less time and fewer resources to create. | Requires more resources and time. | Require less time for testing but more resources for script creatin and updating. |
| Includes an end-to-end functionality to be tested. | Includes test steps, and data expected results for testing. | Includes different commands to develop a script. |
| The main task is to check the full functionality of a software application. | The main task is to verify compliance with the applicable standards, guidelines and customer requirements. | The main task is to verify that nothing is skipped, and the results are true as the desired testing plan. |
| Allows quickly assessing the testing scope. | Allows detecting errors and defects. | Allows carrying out an automatic execution of test cases. |

33. Explain what Test plan is? What is the information that should be covered.

A. A test plan is a document that consists of all future testing -related activities. It is prepared at the project level and in general , it defines work products too be tested , how they will be tested , and test type distribution among the testers.

34. What is Priority?

A. 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 fix the defect. If high-priority status is set based on the customer's requirements.

* **For example:** If the company name is misspelled in the home page of the website then the priority is high and severity is low to fix it.

**35.** What is severity?

A. 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.

**36.** Bug categories are..

A. Software bugs can be classified into multiple categories based on their nature and impact. Broadly speaking, these categories include Function Bugs , Logical Bugs, Workflow Bugs, Unit level bugs, System-Level integration Bugs, out of Bound Bugs , and Security bugs.

37. Advantage of Bugzila.

A. It improves the quality of the product. It enhances the communication between the developing team and the testing tea. It has the capability to adapt to multiple situations.

38. Difference between priority and severity.

A. Severity is basically a parameter that denotes the total impact of a given defect on any software. Priority is basically a parameter that decides the order in which we should fix the defects. Severity relates to the standards of quality. Priority related to the scheduling of defects to resolve them in software.

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

A.

1) Kanban

2) Scrum

3) Extreme Programming

4) Crystal

5) Dynamic Systems Development Method

6) Feature – Driven Development

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

A. Authentication is the process of verifying who someone is , whereas authorization is the process of verifying what specific application, files, and data a user has access to. The situation is like that of an airline that needs to determine which people can come on board.