**Que.-1 what is software testing?**

Ans.-

* Testing is the process of evaluating a system or its component’s with the intent to find that whether it satisfies specified requirements or not.
* Software testing is process used to identify the correctness, completeness and quality to developed computer software.

**Que.-2 what is exploratory testing?**

Ans.-

* In exploratory testing, Tester focus more on how the software actually works, tester do minimum planning and maximum execution of software by which they get in depth idea about software functionality, once tester starts getting inside into the software he can make decisions to what to test next. Exploratory testing is mostly used if the requirements are incomplete and time to release the software is less.

**Que.-3 what is traceability matrix?**

Ans.-

* Traceability matrix (Requirement Traceability Matrix-RTM) is a table which is used to trace requirements during the SDLC. It can be used for forward tracing. (From requirement to design or coding) or backward (coding to requirement).

**Que.-4 what is boundary value testing?**

Ans.-

* Boundary testing is the process of testing between extreme ends or boundaries between partitions of input values.
* So, these are extreme ends like start-end, lower- upper, maximum-minimum, just inside- just outside values are called boundary values and testing is called **boundary testing**.

**Que.-5 what is equivalence partitioning testing?**

Ans.-

* Equivalence partitioning method is also known as equivalence class partitioning(ECP) that divides input domain into classes of data and with 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 EP, Equivalence classes are evaluated for given input condition. Whenever any input is given, then type of input condition is checked, then for this input conditions, Equivalence class represents or describe set of valid or invalid states.

**Que.-6 what is Integration testing?**

Ans.-

* Testing performed to expose defects in the interfaces and in the interaction between integrated components or system.
* Integration Testing is level of software testing process where individual units are combined and tested as a group.
* Integration testing tests integration or interfaces between components, interaction of the different parts of system such as operating system, file system and hardware or interfaces between them.
* When is Integration Testing performed? - Integration Testing is performed after Unit Testing and before System Testing.
* Who performs Integration Testing? - Either Developers themselves or independent Testers perform Integration Testing.

**Que.-7 what determines the level of risk?**

Ans.-

* The risk analysis is then used by Business owners to classify systems into one of three risk categories,

1. Low risk :

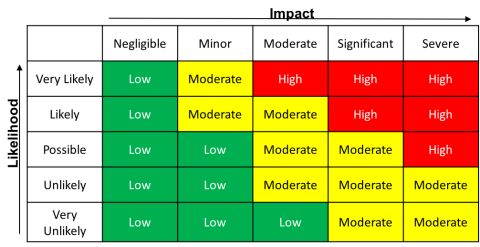
* System processes and/or store public data.
* System is easily recoverable and reproducible.
* System provides an informational/non- critical service.

1. Moderate risk :

* System processes and/or stores non-public or internal-use data
* System is internally trusted by other networked systems.
* System provides normal or important service

1. High risk

* System processes and/or stores confidential or restricted data
* System is highly trusted by UI networked systems
* System provides a critical or campus-wide service
* Risk Analysis must take into consideration the sensitivity of data processed and stored by the system, as well as likelihood and impact of potential threat events.
* A threat events is any event which may cause loss of confidential, integrity, or availability of the system and the data it stores and/or processes.



**Que.-8 what is Alpha testing?**

Ans.-

* **Alpha Testing**: - It is always performed by the developers at the software development site. It is always performed in Virtual Environment. Alpha Testing is always performed at the time of Acceptance Testing when developers test the product and project to check whether it meets the user requirements or not.

**Que.-**9 **what is Beta testing?**

Ans.-

* **Beta testing**: -. It is always performed by the customers at their own site. It is performed in Real Time Environment. Beta Testing is always performed at the time when software product and project are marketed. Beta testing can be considered “pre-release” testing.

**Que.-10 what is component testing?**

Ans.-

* **Component testing**: - Testing technique similar to unit testing but with a higher level of integration - testing is done in the context of the application instead of just directly testing a specific method. Can be performed by testing or development teams.

**Que.-11 what is functional system testing?**

Ans.-

* **Functional system testing**: - A requirement that specifies a function that a system or system component must perform
* A requirement may exit as a document and/or a model.
* There are two type of techniques

1. Requirement based functional testing
2. Business process based testing

* Functional system testing functionality as below:

1. **Accuracy –** Provision of right or agreed results or effect.
2. **Interoperability –** Ability to interact with specifies system.
3. **Compliance –** Adhere to applicable standards, conventions, regulation or laws.
4. **Auditability –** Ability to provide adequate and accurate audit data.
5. **Suitability –** Presence and appropriateness or functions for specifies tasks.

**Que.12 what is Non-functional system testing?**

Ans.-

* Testing of those requirements that do not relate to functionality
* Emphasis on non – functional requirements
* Performance
* Load
* Data volumes
* Storage
* Recovery
* Usability
* Stress
* Security
* The non-functional aspects of a system are all the attributes other than business functionality, and are as important as the functional aspects. These include:
* The look and feel and ease of use of the system.
* How quickly the system performs.
* How much the system can do for the user?
* It is also about:
* How easy and quick the system is to install.
* how robust it is
* how quickly the system can recover from a crash

**Que.13 What is GUI Testing?**

Ans.-

* GUI testing is the process of ensuring proper functionality of the graphical user interface (GUI) for a given application and making sure it conforms to its written specifications.

**Que.-14 what is Adhoc testing?**

Ans.-

* 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.
* The Error guessing is a technique where the experienced and good testers are encouraged to think of situations in which the software may not be able to cope.
* Some people seem to be naturally good at testing and others are good testers because they have a lot of experience either as a tester or working with a particular system and so are able to find out its weaknesses.

**Ans.-15 what is white box testing and list the types of white box testing?**

Que.-

* **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.
* **White Box Techniques**

1. Branch Condition testing
2. Branch Condition Combination testing
3. Modified Condition Decision testing
4. Dataflow testing

**Que.-16 what is black box testing? What are the different black box testing techniques?**

Ans.-

* **Black-box testing**: Testing, either functional or non-functional, without reference to the internal structure of the component or system.
* Techniques of Black Box Testing:

1. Equivalence partitioning
2. Boundary value analysis
3. Decision tables
4. State transition testing
5. Use-case Testing
6. Other Black Box Testing
7. Syntax or Pattern Testing

**Que.17 mention what are the categories of defects?**

Ans.-

* **Data Quality/Database Defects**: Deals with improper handling of data in the database.
* **Critical Functionality Defects**: The occurrence of these bugs hampers the crucial functionality of the application.
* **Functionality Defects**: These defects affect the functionality of the application.
* **Security Defects**: Application security defects generally involve improper handling of data sent from the user to the application. These defects are the most severe and given highest priority to fix.
* **User Interface Defects**: As the name suggests, the bugs deal with problems related to UI are usually considered less severe.

**Que.-18 Mention what big bang testing is?**

Ans.-

* **Big Bang Integration Testing**: Testing technique which integrates individual program modules only when everything is ready. It is performed by the testing teams.

**Que.-19 what is the purpose of exit criteria?**

Ans.-

* 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 (e.g. hand over from System Test to UAT)

**Que.-20 when should "Regression Testing" be performed?**

Ans.-

* Regression testing should be carried out when:-
* n 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
* t 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.-21 what is 7 key principles? Explain in detail.**

Ans.-

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.
* As we find more defects, the probability of undiscovered defects remaining in a system reduces.

1. **Exhaustive Testing is Impossible!**

* Testing everything including all combinations of inputs and preconditions is not possible so, instead of doing the exhaustive testing we can use risks and priorities to focus testing efforts.
* For example: In an application in one screen there are 15 input fields, each having 5 possible values, then to test all the valid combinations you would need 30 517 578 125 (515) tests.
* This is very unlikely that the project timescales would allow for this number of tests. So, accessing and managing risk is one of the most important activities and reason for testing in any project.
* We have learned that we cannot test everything (i.e. all combinations of inputs and preconditions).
* That is we must prioritise our testing effort using a Risk Based Approach.

1. **Why do not Testing Everything?**

* Exhaustive testing of complex software applications:
* requires enormous resources
* Is too expensive
* Takes too long
* It is therefore impractical
* Need an alternative that is pragmatic, affordable, timely and provides result

1. **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.
* These activities should be focused on defined objectives – outlined in the Test Strategy.
* Remember from our Definition of Testing, that Testing doesn’t start once the code has been written!

1. **Defect Clustering**

* A small number of modules contain most of the defects discovered during pre-release testing, or are responsible for the most operational failures.
* Defects are not evenly spread in a system. They are ‘clustered’.
* In other words, most defects found during testing are usually confined to a small number of modules.
* Similarly, most operational failures of a system are usually confined to a small number of modules

1. **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.
  1. **Testing is Context Dependent**
* Testing is done differently in different contexts.
* Different kinds of sites are tested differently. For example: Safety – critical software is tested differently from an e-commerce site.

**Que.-22 Difference between QA v/s QC v/s Tester.**

Ans.-

|  |  |  |
| --- | --- | --- |
| **Quality Assurance** | **Quality Assurance** | **Testing** |
| 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. |
| Activities which ensure the identification of bugs/error/defects in the Software. | Focuses on actual testing by executing Software with intend to identify bug/defect through implementation of procedures and process. | Focuses on actual testing. |
| Process oriented activities. | Product oriented activities. | Product oriented activities. |
| Preventive activities. | It is a corrective process | It is a preventive process. |
| It is a subset of Software Test Life Cycle (STLC). | QC can be considered as the subset of Quality Assurance. | QC can be considered as the subset of Quality Assurance. |

**Que.-23 Difference between Smoke and Sanity?**

Ans.-

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

**Que.-24 Difference between verification and Validation.**

Ans.-

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

**Que.-25 Explain types of Performance testing.**

Ans.-

1. **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.
* It even tests beyond the normal operating point and evaluates how the system works under those extreme conditions.
* Stress Testing is done to make sure that the system would not crash under crunch situations.

1. **Load Testing**

* Its a performance testing to check system behaviour 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.

1. **Endurance Testing**

* Type of testing which checks for memory leaks or other problems that may occur with prolonged execution. It is usually performed by performance engineers.

1. **Volume Testing**

* Testing which confirms that any values that may become large over time (such as accumulated counts, logs, and data files), can be accommodated by the program and will not cause the program to stop working or degrade its operation in any manner. It is usually conducted by the performance engineer.

1. **Scalability testing**

* Part of the battery of non-functional tests which tests a software application for measuring its capability to scale up - be it the user load supported, the number of transactions, the data volume etc. It is conducted by the performance engineer.

1. **Spike Testing**

* Spike testing is a type of performance testing in which an application receives a sudden and extreme increase or decrease in load.

**Que.-26 what is Error, Defect, Bug and failure?**

Ans.-

* **Error**:-The problem in code leads to error, which means that a mistake can occur due to the developer’s coding error as the developers misunderstood the requirement or the requirement was not defined correctly. The developer use term error.
* **Bug**: - In software testing, a bug is the informal name of defects, which means that software or application is not working as per the requirement. When we have some coding error, it leads a program to its breakdown, which is known as bug. The test engineer use terminology.
* **Defect**: - When the application not working as per requirement is known as defects. It is specified as the aberration from the actual and expected result of application or software.
* **Failure**: - Many defects lead to the software’s failure. Which means that a loss specifies a fatal issue in software or application or in its module, which makes the system unresponsive or broken.

**Que.27 Difference between Priority and Severity**

|  |  |
| --- | --- |
| **Priority** | **Severity** |
| Defined by impact on business | Defined by the product’s functionality |
| Categories decide by developers or business owners | Categories decide by testers |
| Deals with the timeframe or order to fix the defects. | Deals with the technical aspects of the product |
| Value of priority is subjective or may change after comparison with other defects. | The value doesn’t change with time. It’s fix |

**Que.-28 What is Bug Life Cycle?**

Ans.-

* Bug life cycle is nothing but the various phases a bug under goes after it is raised or reported.
* The different phases of Bug life cycle are,
* New or Opened
* Assigned
* Fixed
* Tested
* Closed

**Que.-29 Explain the difference between Functional testing and Non-Functional testing.**

Ans.-

|  |  |
| --- | --- |
| **Functional Testing** | **Non-Functional Testing** |
| Functional testing is performed using the functional specification provided by the client and verifies the system against the functional requirements. | Non-Functional testing checks the Performance, reliability, scalability and other non-functional aspects of the software system. |
| Functional testing is executed first | Non-functional testing should be performed after functional testing |
| Manual testing or automation tools can be used for functional testing | 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 | Non-functional testing describes how good the product works |
| Easy to do manual testing | Easy 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 Non-Functional testing are,   * Performance Testing * Load Testing * Volume Testing * Stress Testing * Security Testing * Installation Testing * Penetration Testing * Compatibility Testing * Migration Testing |

**Que.-30 what is the difference between the STLC (Software Testing Life Cycle) and SDLC (Software Development Life Cycle)?**

**Ans.-**

|  |  |
| --- | --- |
| **SDLC** | **STLC** |
| SDLC is mainly related to software development. | STLC is mainly related to software testing. |
| Besides development other phases like testing is also included. | It focuses only on testing the software. |
| 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. |
| SDLC phases are completed before the STLC phases. | STLC phases are performed after SDLC phases. |
| It helps in developing good quality software. | It helps in making the software defects free. |

**Que.-31 what is the difference between test scenarios, test cases, and test script?**

**Ans.-**

|  |  |  |
| --- | --- | --- |
| **Test Scenario** | **Test Case** | **Test Script** |
| Is any functionality that can be tested | Is set of action executed to verify particular feature or functionality | Is set of instruction to test an app automatically |
| Helps test the End-to-End functionality in agile way | Helps in executive testing of an app | Helps to specific things repeatedly |
| Is more focus on what to test | Is focused on what to test and how to test | Is focused on expected result |
| Takes less time and fewer resource to create | Required more time and resource | Required more time to testing but more resource for script creating and updating. |

**Que.-32 Explain what Test Plan is? What is the information that should be covered?**

**Ans.-**

* It is a high level document in which how to perform testing is described. The Test Plan document is usually prepared by the Test Lead or Test Manager and the focus of the document is to describe what to test, how to test, when to test and who will do what test.



* Master test plan: A test plan that typically addresses multiple test levels.
* Phase test plan: A test plan that typically addresses one test phase.
* A test plan will include the following.
* Introduction to the Test Plan document
* Assumptions when testing the application
* List of test cases included in Testing the application
* List of features to be tested
* What sort of Approach to use when testing the software
* List of Deliverables that need to be tested
* The resources allocated for testing the application
* Any Risks involved during the testing process
* A Schedule of tasks and milestones as testing is started

**Que.-33 what are the different Methodologies in Agile Development Model?**

**Ans.-**

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 group’s 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:
* Planning game
* Small releases
* Customer acceptance tests
* Simple design
* Pair programming
* Test-driven development
* Refactoring
* Continuous integration
* Collective code ownership
* Coding standards
* Metaphor
* Sustainable pace

1. **Dynamic Systems Development Method (DSDM)**

* The Dynamic Systems Development Method (DSDM) was born of the need for a common industry framework for rapid software delivery. Rework is to be expected, and any development changes that occur must be reversible. Like Scrum, XP, and FDD, DSDM uses sprints.

1. **Feature-driven development (FDD)**

* Feature-driven development is another software-specific agile framework. This methodology involves creating software models every two weeks and requires a development and design plan for every model feature. It has more rigorous documentation requirements than XP, so it’s better for teams with advanced design and planning abilities.

**Que.-34 Explain the difference between Authorization and Authentication in Web testing.**

**Ans.-**

|  |  |
| --- | --- |
| **Authentication** | **Authorization** |
| The process of establishing the identity of the user. | The process of determining that a requester is allowed to receive a service or perform an operation. |
| In the authentication process, users or persons are verified. | While in this process, users or persons are validated. |
| It is done before the authorization process. | While this process is done after the authentication process. |
| It needs usually the user’s login details. | While it needs the user’s privilege or security levels. |
| Authentication determines whether the person is user or not. | While it determines **What permission does the user have?** |
| **Example**: Employees in a company are required to authenticate through the network before accessing their company email. | **Example:** After an employee successfully authenticates, the system determines what information the employees are allowed to access. |

**Que.-35 what are the common problems faced in Web testing?**

**Ans.-**

We shall discuss challenges often faced by testers while testing a web application.

1. **Integration Testing**

* The rating of an application depends on its usability as well as functionality. Integration testing is a must thing to carry out at the user’s end to check whether the application is reliable, all the critical functionalities work properly as well as there is no significant impact on performance after merging new features.

1. **Security**

* If the application has features like online transaction and payment gateways, testing should be executed to ensure that there are no chance of any fraudulent activities and local storage of payment-related data in the device.

1. **Performance Testing**

* Often a web application gets too slow or crashes when the internet traffic increases all of a sudden. Performance testing should be carried out to ensure that there is no impact on the speed of performing an activity using the application.

1. **Application Getting Slow**

* It does not matter what device is used to access the application, due to poor network coverage or low configuration of processor or physical memory an application may run slower or take an infinite time to load a page. Testing should be conducted to ensure that it is properly optimized to run properly under any condition.

1. **Usability Testing**

* Interactive and dynamic web applications are always popular among users. Proper unit testing should be carried out across devices from the user’s perspective to ensure there are no such issues that may impact the usability of the application.

1. **Entry and Exit Points**

* There are stages when a user will need to navigate out from the application to a third-party website and redirect from another website or gateway to the application. It is a real challenge to test whether this feature works properly.

1. **User Input Validation**

* This is an important part often skipped during testing. Web applications may often feature forms to be filled up by users. Testing should be conducted to ensure that there are proper validation rules and in case of wrong input, the user is warned and further proceeding with the form submission is blocked until the input is corrected.
* That’s all from our side. Always remember to test the above-mentioned facts carefully before your website is launched. Do let us know if you have faced any other challenges while testing your web application.