**Q1. What is Endurance Testing?**

**Endurance testing:** in this testing we test application behavior against the load and stress applies over application for a long duration of time. The goal of this testing are:

* To determine the how the application is going to responds for high load and stress conditions in the real scenario.
* To ensure that the response times in highly load and stress conditions are within the user’s requirement of response time.
* Checks for memory leaks or other problems that may occur with prolonged execution.

**Q2. What is End-to-End testing?**

End-to-End Testing we take the application from the starting phase of the development cycle till the ending of the development cycle. We can simple say that it comes into play when we take requirement from the customer till the end of the delivery of the application. The purposes of End-to-End testing are:

* Validates the software requirements and checks it is integrated with external interfaces.
* Testing application in real world environment scenario.
* It involves testing of interaction between application and database.
* Executed after functional and system testing.
* End-to-End testing is also called Chain Testing

**Q3. What is Gorilla Testing?**

A test technique that involves testing with various ranges of valid and invalid inputs a particular module or component functionality extensively. In Gorilla testing test case and test data are not required. It uses random data and test cases to perform testing of application. The purpose of Gorilla testing is to examine the capability of single module functionality by applying heavy load and stress to it. And determine how much load and stress it can tolerate without getting crashed.

**Q4. Why we need Localization Testing?**

Localization testing mainly deals with the functionality of application and GUI of the application. The purposes of using Localization testing are following:

* Mainly deal with internationalization and localization aspects of software.
* Evaluate how successfully the language is interpreted into a specific language.
* Translate GUI of application so that it can adapt to a particular region language and interface.

**Q5. What is Metric?**

Metric is a standard of measurement. Software metrics uses the statistical method for explaining the structure of the application. The software metric tells us the measurable things like number of bugs per lines of code. We can take the help of software metrics to make the decision regarding the application development. The test metrics is derived from raw test data because what cannot be measured cannot be managed. Software metric also helps the Project Management team to manage the project like schedule for development of each phase.

**Q6. Explain Monkey testing.**

Monkey testing is a type of Black Box Testing used mostly at the Unit Level. In this tester enter the data in any format and check the software is not crashing. In this testing we use Smart monkey and Dumb monkey.

* Smart monkeys are used for load and stress testing, they will help in finding the bugs. They are very expensive to develop.
* Dumb monkey, they are important for basic testing. They help in finding those bugs which are having high severity. Dumb monkey are less expensive as compare to Smart monkeys.

Example: In phone number filed Symbols are entered.

**Q7. What is Negative Testing?**

Negative Testing is performed to find the situation when the software crashed. It is a negative approach, in this tester try to put efforts to find the negative aspects of the application. Negative testing ensures that application can handle the invalid input, incorrect data and incorrect user response. For example, when user enters the alphabetical data in a numeric field, then error message should be display saying “Incorrect data type, please enter a number”

**Q8. What is Path Testing?**

Path testing is a testing in which tester ensure that every path of the application should be executed at least once. In this testing, all paths in the program source code are tested at least once. Tester can use the control flow graph to perform this type of testing.

**Q9. What is Performance Testing?**

Performance Testing is focused on verifying the system performance requirements like response time, Transactional throughput and number of concurrent users. It is used to accurately measure the End-to-End performance of a system. It identifies the loop holes in Architectural Design which helps to tune the application.

It includes the following:

* Emulating ‘n’ number of users interacting with the system using minimal hardware.
* Measuring End-User’s Response time.
* Repeating the load consistently
* Monitoring the system components under controlled load.
* Providing robust analysis and reporting engines

**Q10. What is the difference between baseline and benchmark testing?**

The difference between baseline and benchmark testing are:

* Baseline testing is the process of running a set of tests to capture performance information whereas Benchmarking is the process of comparing application performance with respect to industry standard that is given by some other organization.
* Baseline testing use the information collected to made the change in the application to improve performance and capabilities of the application whereas benchmark information where our application stands with respect to others.
* Baseline compares present performance of application with its own previous performance where as benchmark compares our application performance with other companies application’s performance.

**Q11. What is test driver and test stub?**

* The Stub is called from the software component to be tested. It is used in top down approach.
* The driver calls a component to be tested. It is used in bottom up approach.
* Both test stub and test driver are dummy software components.

We need test stub and test driver because of following reason:

* Suppose we want to test the interface between modules A and B and we have developed only module A. So we cannot test module A but if a dummy module is prepare, using that we can test module A.
* Now module B cannot send or receive data from module A directly so, in these cases we have to transfer data from one module to another module by some external features. This external feature used is called Driver.

**Q12. What is Agile Testing?**

Agile Testing means to quickly validation of the client’s requirements and make the application of high quality user interface. When the build is released to the testing team, testing of the application is started to find the bugs. As a Tester, we need to focus on the customer or end user requirements. We put the efforts to deliver the quality product in spite of short time frame which will further help in reducing the cost of development and test feedbacks will be implemented in the code which will avoid the defects coming from the end user.

**Q13. Explain bug life cycle.**

Bug Life Cycle:

* When a tester finds a bug .The bug is assigned NEW or OPEN with status,
* The bug is assigned to development project manager who will analyze the bug .He will check whether it is a valid defect. If not valid bus is rejected, now status is REJECTED.
* If not, next the defect is checked whether it is in scope. When bug is not part of the current release .Such defects are POSTPONED
* Now, Tester checks whether similar defect was raised earlier. If yes defect is assigned a status DUPLICATE
* When bug is assigned to developer. During this stage bug is assigned a status IN-PROGRESS
* Once bug is fixed. Defect is assigned a status FIXED
* Next the tester will re-test the code. In case the test case passes the defect is CLOSED
* If test case fails again the bug is RE-OPENED and assigned to the developer. That’s all to Bug Life Cycle.

**Q14. What is Matching Defects?**

Matching Defects helps us to remove the locking of same defect in the bug in the application. While using QC, every time we lock a bug, QC saves the list of keywords from the Summary and Description Fields of the bug. When we search for similar defects in QC, keywords in these fields are matched with other defects which are locked previously. Keywords are more than two characters and they are not case sensitive. We have two methods to conduct search of similar defects.

* **Finding similar Defects:** compare a selected defect with all other existing defects in project.
* **Finding similar Text:** compare a specific test string against all other existing defects in project.

**Q15. What is Recovery Testing?**

Recovery testing is done to check how fast and better the application can recover against any type of crash or hardware failure. Type or extent of recovery is specified in the requirement specifications. Recovery testing will enable customer to avoid any inconvenience that are generally associated with the loss of data and performance of the application. We can perform regular recovery testing in order to take backup of all necessary and important data.

**Q17. What is Test Case?**

A test case is a set of conditions which is used by tester to perform the testing of application to make sure that application is working as per the requirement of the user.

* A Test Case contains information like test steps, verification steps, prerequisites, outputs, test environment, etc
* The process of developing test cases can also enable us to determine the issues related to the requirement and designing process of the application.

**Q18. In Test First Design what step you will follow to add new functionality into the project?**

When we have to add new functionality our project, we perform the following steps:

* Quickly add a developer test: we need to create a test that ensures that new added functionality will not crash our project.
* Run your tests. Execute that test, to ensure that new add functionality does not crash our application.
* Update your production code. In this we update our code with few more functionality so that the code passes the new test. Like adding of error message in field where field can take only numeric data.
* Run your test suite again. If test fails, we have to do change in the code and perform retesting of the application.

**Q18. What is Validation and Verification?**

Verification: process of evaluating work-products of a development phase to determine whether they fulfill the specified requirements for that phase.  
Validation: process of evaluating software during or at the end of the development process to determine whether it specified requirements.

Difference between Verification and Validation:

* Verification is Static testing where as Validations is Dynamic Testing.
* Verification takes place before validation.
* Verification evaluates plans, document, requirements and specification, where as Validation evaluates product.
* Verification inputs are checklist, issues list, walkthroughs and inspection where as in Validation testing of actual product.

Verification output is set of document, plans, specification and requirement documents where as in Validation actual product is output.

**Q19. What are different approaches to do Integration Testing?**

Integration testing is black box testing. Integration testing focuses on the interfaces between units, to ensure that units work together to complete a specify task. The purpose of integration testing is to confirm that different components of the application interact with each other. Integration testing is considered complete, when actual results and expected results are same. There are mainly three approaches to do integration testing.

* **Top-down Approach:** Tests the components by integrating from top to bottom.
* **Bottom-up approach:** It takes place from the bottom of the control flow to the higher level components
* **Big bang approach:** In this are different module are joined together to form a complete system and then testing is performed on it.

**Q20. Can you explain the elementary process?**

Software applications are made up by the help of several elementary processes. There are two types of elementary processes:

* **Dynamic elementary Process:** The dynamic elementary involves process of moving data from one location to another location. The location can be within the application and outside the application.
* **Static elementary Process:** Static elementary involves maintaining the data of the application.

**Q21. Explain the PDCA cycle.**

Software testing is an important part of the software development process. In normal software development there are four important steps PDCA (Plan, Do, Check, Act) cycle. The four steps are discussed below:

* **Plan:** Define the goal and the plan for achieving that goal.
* **Do:** execute those plan strategy which is planned in plan phase
* **Check:** Check to make sure that everything is going according to the plan and gets the expected results.
* **Act:** Act according to that issue.

**Q22. What are the categories of defects?**

There are three main categories of defects:

* **Wrong:** The requirements are implemented incorrectly in the application.
* **Missing:** When requirement given by the customer and application is unable to meet those application.
* **Extra:** A requirement incorporated into the product that was not given by the end customer. This is always a variance from the specification, but may be an attribute desired by the user of the product.

**Q23. What are different types of verifications?**

Verification is static type of software testing which is started in earlier phase of development of software. In this approach we don’t execute the software that the reason it comes in static testing. The product is evaluated by going through the code. Types of verification are:

* **Walkthrough:** Walkthroughs are informal technique. Where the Developer leader organizing a meeting with team member to take feedback regarding the software. This can be used for the improvement of the software quality. Walkthrough are unplanned in the SDLC cycle.
* **Inspection:** Inspection is a done by checking of a software product thoroughly with the intention to find out defect and ensuring that software is meeting the user requirements.

**Q24. Which test cases are written first: white boxes or black boxes?**

Generally, black box test cases are written first and white box test cases later. To write black box test cases we need the requirement documents and design or project plan. All these documents are easily available in the earlier phase of the development. A black box test case does not need functional design of the application but white box testing needs. Structural design of the application is clearer in the later part of project, mostly while executing or designing. For black box testing you need to only analyze from the functional perspective which is easily available from a simple requirement document.

**Q25. What is difference between latent and masked defect?**

The difference between latent and masked defect are:

* A latent defect is an existing defect that has not yet caused a failure because the conditions that are required to invoke the defect is not meet.
* A masked defect is an existing defect that has not yet caused a failure just because another defect hides that part of the code from being executed where it is present.

**Q26. What is coverage and what are the different types of coverage techniques?**

Coverage is a measurement used in software testing to describe the degree to which the source code is tested. There are three basic types of coverage techniques as shown in the following figure:

* **Statement coverage:** This coverage ensures that each line of source code has been executed and tested.
* **Decision coverage:** This coverage ensures that every decision (true/false) in the source code has been executed and tested.
* **Path coverage:** In this coverage we ensure that every possible route through a given part of code is executed and tested.

**Q27. Explain the concept of defect cascading?**

Defect cascading is a defect which is caused by another defect. In this one defect invokes the other defect in the application. When a defect is present in any stage but is not identified, hide to other phases without getting noticed. This will result in increase in number of defects.

**Q28. What are the basic elements of defect report format?**

The basic elements of Defect Report Format are:

1. Project name  
2. Module name  
3. Defect detected on  
4. Defect detected by  
5. Defect id  
6. Defect name  
7. Snapshot of the defect(if the defect is in the non reproducible environment)  
8. Priority, severity, status  
9. Defect resolved by  
10. Defect resolved on.

**Q29. What is destructive testing, and what are its benefits?**

Destructive testing includes methods where material is broken down in to evaluate the mechanical properties, such as strength, toughness and hardness. For example, finding the quality of a weld is good enough to withstand extreme pressure and also to verify the properties of a material.

Benefits of Destructive Testing (DT)

* Verifies properties of a material
* Determines quality of welds
* Helps you to reduce failures, accidents and costs
* Ensures compliance with regulations

**Q30. What is Use Case Testing?**

**Use Case:** A use case is a description of the process which is performed by the end user for a particular task. Use case contains a sequence of step which is performed by the end user to complete a specific task or a step by step process that describe how the application and end user interact with each other. Use case is written by the user point of view.

**Use case Testing:** the use case testing uses this use case to evaluate the application. So that, the tester can examines all the functionalities of the application. Use case testing cover whole application.