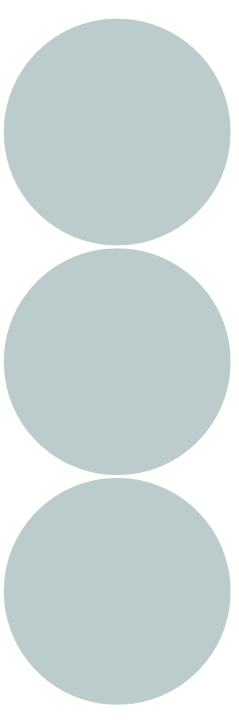


Manual Testing Interview Questions and Answers



Q1. What is the difference between Functional Requirement and Non-Functional Requirement?

Ans: The Functional Requirement specifies how the system or application SHOULD DO where in

Non Functional Requirement it specifies how the system or application SHOULD BE.

Some functional Requirements are

- Authentication
- Business rules
- Historical Data
- Legal and Regulatory Requirements
- External Interfaces

Some Non-Functional Requirements are

- Performance
- Reliability
- Security
- Recovery
- Data Integrity
- Usability

Q2. How Severity and Priority are related to each other?

Ans:

- Severity- tells the seriousness/depth of the bug where as
- Priority- tells which bug should rectify first.
- Severity- Application point of view
- Priority- User point of view

Q3. Explain the different types of Severity?

Ans:

1. User Interface Defect-**Low**
2. Boundary Related Defects-**Medium**
3. Error Handling Defects-**Medium**
4. Calculation Defects-**High**
5. Interpreting Data Defects-**High**
6. Hardware Failures& Problems-**High**
7. Compatibility and Inter system defects-**High**
8. Control Flow defects-**High**
9. Load conditions (Memory leakages under load testing)-**High**

Q4a. What is the difference between Priority and Severity?

Ans: The terms Priority and Severity are used in Bug Tracking to share the importance of a bug among the team and to fix it.

Severity: Is found in the Application point of view

Priority- Is found in the User point of view

Severity- (tells the seriousness/depth of the bug)

1. The Severity status is used to explain how badly the deviation is affecting the build.
2. The severity type is defined by the tester based on the written test cases and functionality.

Example

If an application or a 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 and priority is low.

PRIORITY- (tells which bug should rectify first)

1. The Priority status is set by the tester to the developer mentioning the time frame to fix a defect. If High priority is mentioned then the developer has to fix it at the earliest.
2. The priority status is set based on the customer requirements.

Example

If the company name is misspelled in the home page of a website, then the priority is high and the severity is low to fix it.

Severity: Describes the bug in terms of functionality.

Priority: Describes the bug in terms of customer.

Few examples:

High Severity and Low Priority -> Application doesn't allow customer expected configuration.

High Severity and High Priority -> Application doesn't allow multiple user's.

Low Severity and High Priority -> No error message to prevent wrong operation. **Low**

Severity and low Priority -> Error message is having complex meaning.

Or

Few examples:

High Severity -Low priority

Supposing, you try the wildest or the weirdest of operations in a software (say, to be released the next day) which a normal user would not do and supposing this renders a run -time error in the application, the severity would be high. The priority would be low as the operations or the steps which rendered this error by most chances will not be done by a user.

Low Severity -High priority

An example would be- you find a spelling mistake in the name of the website which you are testing. Say, the name is supposed to be Google and its spelled there as 'Gaogle'. Though, it doesn't affect the basic functionality of the software, it needs to be corrected before the release. Hence, the priority is high.

High severity- High Priority

A bug which is a show stopper. i.e., a bug due to which we are unable to proceed our testing. An example would be a run time error during the normal operation of the software, which would cause the application to quit abruptly.

Low severity - low priority Cosmetic bugs

Q4b. What is Defect Severity?

Ans: A defect is a product anomaly or flaw, which is variance from desired product specification. The classification of defect based on its impact on operation of product is called Defect Severity.

Q5. What is Bucket Testing?

Ans: Bucket testing (also known as A/B Testing) is mostly used to study the impact of various product designs in website metrics, two simultaneous versions were run in a single or set of web pages to measure the difference in click rates, interface and traffic.

Q6. What is Entry and Exit Criteria in Software Testing?

Ans: Entry Criteria is the process that must be present when a system begins, like,

- SRS (Software Requirement Specification)
- FRS (Functional Requirement Specification)
- Usecase
- Test Case
- Test plan

Exit Criteria ensures whether testing is completed and the application is ready for release, like,

- Test Summary Report
- Metrics
- Defect Analysis report

Q7. What is Concurrency Testing?

Ans: Concurrency Testing (also commonly known as Multi User Testing) is used to know the effects of accessing the Application, Code Module or Database by different users at the same time. It helps in identifying and measuring the problems in Response time, levels of locking and deadlocking in the application.

Example

Load runner is widely used for this type of testing, Vugen (Virtual User Generator) is used to add the number of concurrent users and how the users need to be added like Gradual Ramp up or Spike Stepped.

Q8. Explain Statement coverage/Code coverage/Line Coverage?

Ans: Statement Coverage or Code Coverage or Line Coverage is a metric used in White Box Testing where we can identify the statements executed and where the code is not executed cause of blockage. In this process each and every line of the code needs to be checked and executed.

Some advantages of Statement Coverage / Code Coverage / Line Coverage are

- It verifies what the written code is expected to do and not to do.
- It measures the quality of code written.
- It checks the flow of different paths in the program also ensure whether those paths are tested or not.

To Calculate Statement Coverage,

Statement Coverage = Statements Tested / Total No. of Statements.

Q9. Explain Branch Coverage/Decision Coverage?

Ans: Branch Coverage or Decision Coverage metric is used to check the volume of testing done in all components. This process is used to ensure whether all the code is executed by verifying every branch or decision outcome (if and while statements) by executing atleast one time, so that no branches lead to the failure of the application.

To Calculate Branch Coverage,

Branch Coverage = Tested Decision Outcomes / Total Decision Outcomes.

Q10. What is the difference between High level and Low Level test case?

Ans: High level Test cases are those which cover major functionality in the application (i.e. retrieve, update display, cancel (functionality related test cases), database test cases). Low level test cases are those related to User Interface (UI) in the application.

Q11. Explain Localization testing with example?

Ans: Localization is the process of changing or modifying an application to a particular culture or locale. This includes change in user interface, graphical designs or even the initial settings according to their culture and requirements. In terms of Localization Testing it verifies how correctly the application is changed or modified into that target culture and language.

In case of translation required of the application on that local language, testing should be done on each field to check the correct translation. Other formats like

date conversion, hardware and software usage like operating system should also be considered in localization testing.

Examples for Localization Testing are In Islamic Banking all the transactions and product features are based on Shariah Law, some important points to be noted in Islamic Banking are

1. In Islamic Banking, the bank shares the profit and loss with the customer.
2. In Islamic Banking, the bank cannot charge interest on the customer; instead they charge a nominal fee which is termed as "Profit
3. In Islamic Banking, the bank will not deal or invest in business like Gambling, Alcohol, Pork, etc.

In this case, we need to test whether these Islamic banking conditions were modified and applied in the application or product.

In Islamic Lending, they follow both the Gregorian calendar and Hijiri Calendar for calculating the loan repayment schedule. The Hijiri Calendar is commonly called as Islamic Calendar followed in all the Muslim countries according to the lunar cycle. The Hijiri Calendar has 12 months and 354 days which is 11 days shorter than Gregorian calendar. In this case, we need to test the repayment schedule by comparing both the Gregorian calendar and Hijiri Calendar.

Q12. Explain Risk Analysis in Software Testing?

Ans: In Software Testing, Risk Analysis is the process of identifying risks in applications and prioritizing them to test.

In Software testing some unavoidable risk might takes place like

- Change in requirements or Incomplete requirements
- Time allocation for testing.
- Developers delaying to deliver the build for testing.
- Urgency from client for delivery.
- Defect Leakage due to application size or complexity.

To overcome these risks, the following activities can be done

- Conducting Risk Assessment review meeting with the development team.
- Profile for Risk coverage is created by mentioning the importance of each area.
- Using maximum resources to work on High Risk areas like allocating more testers for High risk areas and minimum resources for Medium and Low risk areas. Creation of Risk assessment database for future maintenance and management review.

Q13. What is the difference between Two Tier Architecture and Three Tier Architecture?

Ans:

In Two Tier Architecture or Client/Server Architecture two layers like Client and Server is involved. The Client sends request to Server and the Server responds to the request by fetching the data from it. The problem with the Two Tier Architecture is the server cannot respond to multiple requests at the same time which causes data integrity issues.

The Client/Server Testing involves testing the Two Tier Architecture of user interface in the front end and database as backend with dependencies on Client, Hardware and Servers. **In**

Three Tier Architecture or Multi Tier Architecture three layers like Client, Server and Database are involved. In this the Client sends a request to Server, where the Server sends the request to Database for data, based on that request the Database sends back the data to Server and from Server the data is forwarded to Client.

The Web Application Testing involves testing the Three Tier Architecture including the User interface, Functionality, Performance, Compatibility, Security and Database testing.

Q14. What is the difference between Static testing and dynamic testing ?

Ans: Static Testing (done in Verification stage)

Static Testing is a White Box testing technique where the developers verify or test their code with the help of checklist to find errors in it, this type of testing is done without running the actually developed application or program. Code Reviews, Inspections, Walkthroughs are mostly done in this stage of testing.

Dynamic Testing (done in Validation stage)

Dynamic Testing is done by executing the actual application with valid inputs to check the expected output. Examples of Dynamic Testing methodologies are Unit Testing, Integration Testing, System Testing and Acceptance Testing.

Some differences between Static Testing and Dynamic Testing are,

- Static Testing is more cost effective than Dynamic Testing because Static Testing is done in the initial stage.
- In terms of Statement Coverage, the Static Testing covers more areas than Dynamic Testing in shorter time.
- Static Testing is done before the code deployment where the Dynamic Testing is done after the code deployment.
- Static Testing is done in the Verification stage where the Dynamic Testing is done in the Validation stage.

Q15. Explain Use case diagram. What are the attributes of use cases?

Ans: Use Case Diagrams is an overview graphical representation of the functionality in a system. It is used in the analysis phase of a project to specify the system to be developed. In Use Case Diagrams the whole system is defined as

ACTORS, USE CASES and ASSOCIATIONS, the ACTORS are the external part of the system like users, computer software & hardware, USECASES is the behavior or functionality of the system when these ACTORS perform an action, the ASSOCIATIONS are the line drawn to show the connection between ACTORS and USECASES. One ACTOR can link too many USECASES and one USECASE can link too many ACTORS.

Q16. What is Web Application testing? Explain the different phases in Web Application testing?

Ans: Web Application testing is done on a website to check its load, performance, Security, Functionality, Interface, compatibility and other usability related issues. In Web application testing, three phases of testing is done, they are,

Web Tier Testing

In Web tier testing, the browser compatibility of the application will be tested for IE, Fire Fox and other web browsers.

Middle Tier Testing

In Middle tier testing, the functionality and security issues were tested.

Database Tier Testing

In Database tier testing, the database integrity and the contents of the database were tested and verified.

Q17. Explain Unit testing, Interface Testing and Integration testing. Also explain the types of integration testing in brief?

Unit testing

Ans: Unit Testing is done to check whether the individual modules of the source code are working properly. i.e. testing each and every unit of the application separately by the developer in developer's environment.

Interface Testing

Interface Testing is done to check whether the individual modules are communicating properly one among other as per the specifications.

Interface testing is mostly used in testing the user interface of GUI application.

Integration testing

Integration Testing is done to check the connectivity by combining all the individual modules together and test the functionality. **The types of Integration Testing are**

1.Big Bang Integration Testing

In Big Bang Integration Testing, the individual modules are not integrated until all the modules are ready. Then they will run to check whether it is performing well.

In this type of testing, some disadvantages might occur like,

Defects can be found at the later stage. It would be difficult to find out whether the defect arose in Interface or in module.

2.Top Down Integration Testing

In Top Down Integration Testing, the high level modules are integrated and tested first. i.e Testing from main module to sub module. In this type of testing, Stubs are used as temporary module if a module is not ready for integration testing.

3. Bottom Up Integration Testing

In Bottom Up Integration Testing, the low level modules are integrated and tested first i.e Testing from sub module to main module. Same like Stubs, here drivers are used as a temporary module for integration testing.

Q18. Explain Alpha, Beta, Gamma Testing?

Ans: Alpha Testing:

Alpha Testing is mostly like performing usability testing which is done by the in-house developers who developed the software or testers. Sometimes this Alpha Testing is done by the client or an outsider with the presence of developer and tester. The version release after alpha testing is called Alpha Release.

Beta Testing:

Beta Testing is done by limited number of end users before delivery, the change request would be fixed if the user gives feedback or reports defect. The version release after beta testing is called beta Release.

Gamma Testing:

Gamma Testing is done when the software is ready for release with specified requirements, this testing is done directly by skipping all the in-house testing activities.

Q19. Explain the methods and techniques used for Security Testing?

Ans: Security testing can be performed in many ways like,

1. Black Box Testing
2. White Box Testing
3. Database Testing
4. Black Box Testing
5. Session Hijacking

Session Hijacking commonly called as "IP Spoofing" where a user session will be attacked on a protected network.

1.Session Prediction

Session prediction is a method of obtaining data or a session ID of an authorized user and gets access to the application. In a web application the session ID can be retrieved from cookies or URL.

The session prediction happening can be predicted when a website is not responding normally or stops responding for an unknown reason.

2.Email Spoofing

Email Spoofing is duplicating the email header ("From" address) to look like originated from actual source and if the email is replied it will land in the spammers inbox. By inserting commands in the header the message information can be altered. It is possible to send a spoofed email with information you didn't write.

3.Content Spoofing

Content spoofing is a technique to develop a fake website and make the user believe that the information and website is genuine. When the user enters his Credit Card Number, Password, SSN and other important details the hacker can get the data and use it for fraud purposes.

4. Phishing

Phishing is similar to Email Spoofing where the hacker sends a genuine look like mail attempting to get the personal and financial information of the user. The emails will appear to have come from well known websites.

5. Password Cracking

Password Cracking is used to identify an unknown password or to identify a forgotten password

Password cracking can be done through two ways,

1. **Brute Force** – The hacker tries with a combination of characters within a length and tries until it is getting accepted.
2. **Password Dictionary** – The hacker uses the Password dictionary where it is available on various topics.
3. **White Box level** 4. **Malicious Code Injection**

SQL Injection is most popular in Code Injection Attack, the hacker attach the malicious code into the good code by inserting the field in the application. The motive behind the injection is to steal the secured information which was intended to be used by a set of users.

Apart from SQL Injection, the other types of malicious code injection are XPath Injection, LDAP Injection, and Command Execution Injection. Similar to SQL Injection the XPath Injection deals with XML document.

b. Penetration Testing:

Penetration Testing is used to check the security of a computer or a network. The test process explores all the security aspects of the system and tries to penetrate the system.

c. Input validation:

Input validation is used to defend the applications from hackers. If the input is not validated mostly in web applications it could lead to system crashes, database manipulation and corruption.

d. Variable Manipulation

Variable manipulation is used as a method for specifying or editing the variables in a program. It is mostly used to alter the data sent to web server.

3. Database Level

1. SQL Injection

SQL Injection is used to hack the websites by changing the backend SQL statements, using this technique the hacker can steal the data from database and also delete and modify it.

Q20.Explain IEEE 829 standards and other Software Testing standards?

Ans: An IEEE 829 standard is used for Software Test Documentation, where it specifies format for the set of documents to be used in the different stages software testing. The documents are,

Test Plan- Test Plan is a planning document which has information about the scope, resources, duration, test coverage and other details.

Test Design- Test Design document has information of test pass criteria with test conditions and expected results.

Test Case- Test case document has information about the test data to be used.

Test Procedure- Test Procedure has information about the test steps to be followed and how to execute it.

Test Log- Test log has details about the run test cases, test plans & fail status, order, and the resource information who tested it.

Test Incident Report- Test Incident Report has information about the failed test comparing the actual result with expected result.

Test Summary Report- Test Summary Report has information about the testing done and quality of the software, it also analyses whether the software has met the requirements given by customer.

The other standards related to software testing are,

IEEE 1008 is for Unit Testing

IEEE 1012 is for Software verification and validation

IEEE 1028 is for Software Inspections

IEEE 1061 is for Software metrics and methodology

IEEE 1233 is for guiding the SRS development

IEEE 12207 is for SLC process

Q21. What is Test Harness?

Ans: Test Harness is configuring a set of tools and test data to test an application in various conditions, which involves monitoring the output with expected output for correctness. The benefits of Test Harness are,

- Productivity increase due to process automation.
- Quality in the application.

Q22 .What is the difference between bug log and defect tracking ?

Ans: Bug Log: Bug Log document showing the number of defect such as open, closed, reopen or deferred of a particular module

Defect Tracking- The process of tracking a defect such as symptom, whether reproducible /not, priority, severity and status.

Q23. What are Integration Testing and Regression Testing?

Ans:

Integration Testing:

- Combining the modules together & construct software architecture.
- To test the communication & data flow
- White & Black box testing techniques are used
- It is done by developer & tester

Regression Testing

- It is re-execution of our testing after the bug is fixed to ensure that the build is free from bugs.
- Done after bug is fixed
- It is done by Tester

Q24. Explain Peer Review in Software Testing?

Ans:

It is an alternative form of Testing, where some colleagues were invited to examine your work products for defects and improvement opportunities.

Some Peer review approaches are,

Inspection

It is a more systematic and rigorous type of peer review. Inspections are more effective at finding defects than are informal reviews.

Ex: In Motorola's Iridium project nearly 80% of the defects were detected through inspections where only 60% of the defects were detected through formal reviews.

Team Reviews: It is a planned and structured approach but less formal and less rigorous comparing to Inspections.

Walkthrough: It is an informal review because the work product's author describes it to some colleagues and asks for suggestions. Walkthroughs are informal because they typically do not follow a defined procedure, do not specify exit criteria, require no management reporting, and generate no metrics.

Or

A '**walkthrough**' is an informal meeting for evaluation or informational purposes. Little or no preparation is usually required.

Pair Programming: In Pair Programming, two developers work together on the same program at a single workstation and continuously reviewing their work.

Peer Desk check

In Peer Desk check only one person besides the author examines the work product. It is an informal review, where the reviewer can use defect checklists and some analysis methods to increase the effectiveness.

Passaround: It is a multiple, concurrent peer desk check where several people are invited to provide comments on the product.

Q25. Explain Compatibility testing with an example?

Ans:

Compatibility testing is to evaluate the application compatibility with the computing environment like Operating System, Database, Browser compatibility, backwards compatibility, computing capacity of the Hardware Platform and compatibility of the Peripherals.

Example

If Compatibility testing is done on a Game application, before installing a game on a computer, its compatibility is checked with the computer specification that whether it is compatible with the computer having that much of specification or not.

Q26. What is Traceability Matrix?

Ans:

Traceability Matrix is a document used for tracking the requirement, Test cases and the defect. This document is prepared to make the clients satisfy that the coverage done is complete as end to end, this document consists of Requirement/Base line doc Ref No., Test case/Condition, Defects / Bug id. Using this document the person can track the Requirement based on the Defect id.

Q27. Explain Boundary value testing and Equivalence testing with some examples?

Ans:

Boundary value testing is a technique to find whether the application is accepting the expected range of values and rejecting the values which falls out of range. **Example**

A user ID text box has to accept alphabet characters (a - z) with length of 4 to 10 characters.

BVA is done like this, max value: 10 pass; max-1: 9 pass; max+1=11 fail ;min=4 pass;min+1=5 pass;min-1=3 fail;

Like wise we check the corner values and come out with a conclusion whether the application is accepting correct range of values.

Equivalence testing is normally used to check the type of the object. **Example**

A user ID text box has to accept alphabet characters (a - z) with length of 4 to 10 characters.

In +ve condition we have test the object by giving alphabets. i.e. a-z char only, after that we need to check whether the object accepts the value, it will pass.

In -ve condition we have to test by giving other than alphabets (a-z) i.e. A-Z, 0-9, blank etc, it will fail.

Q28. What is Security testing?

Ans:

Security testing is the process that determines that confidential data stays confidential Or Testing how well the system protects against unauthorized internal or external access, willful damage, etc?

This process involves functional testing, penetration testing and verification.

Q29. What is Installation testing?

Ans: Installation testing is done to verify whether the hardware and software are installed and configured properly. This will ensure that all the system components were used during the testing process. This Installation testing will look out the testing for a high volume data, error messages as well as security testing.

Q30. What is AUT?

Ans: AUT is nothing but "Application Under Test". After the designing and coding phase in Software development life cycle, the application comes for testing then at that time the application is stated as Application Under Test.

Q31. What is Defect Leakage?

Ans: Defect leakage occurs at the Customer or the End user side after the application delivery. After the release of the application to the client, if the end user gets any type of defects by using that application then it is called as Defect leakage. This Defect Leakage is also called as Bug Leakage.

Q32. What are the contents in an effective Bug report?

Ans:

1. Project
2. Subject
3. Description
4. Summary
5. Detected By (Name of the Tester)
6. Assigned To (Name of the Developer who is supposed to the Bug)
7. Test Lead (Name)

8. Detected in Version
9. Closed in Version
10. Date Detected
11. Expected Date of Closure
12. Actual Date of Closure
13. Priority (Medium, Low, High, Urgent)
14. Severity (Ranges from 1 to 5)
15. Status
16. Bug ID
17. Attachment
18. Test Case Failed (Test case that is failed for the Bug)

Q33. What is Error guessing and Error seeding?

Ans: Error Guessing is a test case design technique where the tester has to guess what faults might occur and to design the tests to represent them.

Error Seeding is the process of adding known faults intentionally in a program for the reason of monitoring the rate of detection & removal and also to estimate the number of faults remaining in the program.

Q34. What is Ad-hoc testing?

Ans: Ad hoc testing is concern with the Application Testing without following any rules or test cases.

For Ad hoc testing one should have strong knowledge about the Application.

Q35. What are the basic solutions for the software development problems?

Ans:

1. **Basic requirements-** A clear, detailed, complete, achievable, testable requirement has to be developed. Use some prototypes to help pin down requirements. In nimble environments, continuous and close coordination with customers/end-users is needed.
2. **Schedules should be realistic-** enough time to plan, design, test, bug fix, re-test, change, and document in the given schedule. Adequate
3. **testing-** testing should be started early, it should be re-tested after the bug fixed or changed, enough time should be spend for testing and bug-fixing.
4. **Proper study on initial requirements-** be ready to look after more changes after the development has begun and be ready to explain the changes done to others. Work closely with the customers and end-users to manage expectations. This avoids excessive changes in the later stages.
5. **Communication-** conduct frequent inspections and walkthroughs in appropriate time period; ensure that the information and the documentation is available on up-to-date if possible electronic. More emphasize on promoting teamwork and cooperation inside

the team; use prototypes and proper communication with the end-users to clarify their doubts and expectations.

Q36. What are the common problems in the software development process?

Ans: Inadequate requirements from the Client: if the requirements given by the client is not clear, unfinished and not testable, then problems may come.

Unrealistic schedules: Sometimes too much of work is being given to the developer and ask him to complete in a Short duration, then the problems are unavoidable.

Insufficient testing: The problems can arise when the developed software is not tested properly.

Given another work under the existing process: request from the higher management to work on another project or task will bring some problems when the project is being tested as a team.

Miscommunication: in some cases, the developer was not informed about the Clients requirement and expectations, so there can be deviations.

Q37. What is the difference between Software Testing and Quality Assurance (QA) ?

Ans: Software Testing involves operation of a system or application under controlled conditions and evaluating the result. It is oriented to 'detection'.

Quality Assurance (QA) involves the entire software development PROCESS- monitoring and improving the process, making sure that any agreed-upon standards and procedures are followed, and ensuring that problems are found and dealt with. It is oriented to 'prevention'.

Q38. How to Test the water bottle?

Ans:

Note: Before going to generate some test idea on how to test a water bottle, I would like to ask few questions like:

1. Is it a bottle made up off glass, plastic, rubber, some metal, some kind of disposable materials or any thing else?
2. Is it meant only to hot water or we can use it with other fluids like tea, coffee, soft drinks, hot chocolate, soups, wine, cooking oil, vinegar, gasoline, acids, molten lava (!) etc.?
3. Who is going to use this bottle? A school going kid, a housewife, some beverage manufacturing company, an office-goer, a sports man, a mob protesting in a rally (going to use as missiles), an Eskimo living in an igloo or an astronaut in a space ship?

These kinds of questions may allow a tester to know a product (that he is going to test) in a better way. In our case, I am assuming that the water bottle is in form of a pet bottle and actually made up off either plastic or glass (there are 2 versions of the product) and is

intended to be used mainly with water. About the targeted user, even the manufacturing company is not sure about them! (Sounds familiar! When a software company develops a product without clear idea about the users who are going to use the software!)

Test Ideas

1. Check the dimension of the bottle. See if it actually looks like a water bottle or a cylinder, a bowl, a cup, a flower vase, a pen stand or a dustbin! [Build Verification Testing!]
2. See if the cap fits well with the bottle.[Installability Testing!]
3. Test if the mouth of the bottle is not too small to pour water. [Usability Testing!]
4. Fill the bottle with water and keep it on a smooth dry surface. See if it leaks. [Usability Testing!]
5. Fill the bottle with water, seal it with the cap and see if water leaks when the bottle is tilted, inverted, squeezed (in case of plastic made bottle)! [Usability Testing!]
6. Take water in the bottle and keep it in the refrigerator for cooling. See what happens. [Usability Testing!]
7. Keep a water-filled bottle in the refrigerator for a very long time (say a week). See what happens to the water and/or bottle. [Stress Testing!]
8. Keep a water-filled bottle under freezing condition. See if the bottle expands (if plastic made) or breaks (if glass made). [Stress Testing!]
9. Try to heat (boil!) water by keeping the bottle in a microwave oven! [Stress Testing!]
10. Pour some hot (boiling!) water into the bottle and see the effect. [Stress Testing!]
11. Keep a dry bottle for a very long time. See what happens. See if any physical or chemical deformation occurs to the bottle.
12. Test the water after keeping it in the bottle and see if there is any chemical change. See if it is safe to be consumed as drinking water.
13. Keep water in the bottle for sometime. And see if the smell of water changes.
14. Try using the bottle with different types of water (like hard and soft water). [Compatibility Testing!]
15. Try to drink water directly from the bottle and see if it is comfortable to use. Or water gets spilled while doing so. [Usability Testing!]
16. Test if the bottle is ergonomically designed and if it is comfortable to hold. Also see if the center of gravity of the bottle stays low (both when empty and when filled with water) and it does not topple down easily.
17. Drop the bottle from a reasonable height (may be height of a dining table) and see if it breaks (both with plastic and glass model). If it is a glass bottle then in most cases it may break. See if it breaks into tiny little pieces (which are often difficult to clean) or breaks into nice large pieces (which could be cleaned without much difficulty). [Stress Testing!] [Usability Testing!]
18. Test the above test idea with empty bottles and bottles filled with water. [Stress Testing!]
19. Test if the bottle is made up of material, which is recyclable. In case of plastic made bottle test if it is easily crushable.

20. Test if the bottle can also be used to hold other common household things like honey, fruit juice, fuel, paint, turpentine, liquid wax etc. [Capability Testing!] **Q39. What is**

Portlet Testing ?

Ans:

Following are the features that should be concentrated while testing a portlet

i. Test alignment/size display with multiple style sheets and portal configurations. When you configure a portlet object in the portal, you must choose from the following alignments:

a. Narrow portlets are displayed in a narrow side column on the portal page. Narrow portlets must fit in a column that is fewer than 255 pixels wide.

b. Wide portlets are displayed in the middle or widest side column on the portal page. Wide portlets fit in a column fewer than 500 pixels wide.

ii. Test all links and buttons within the portlet display. (if there are errors, check that all forms and functions are uniquely named, and that the preference and gateway settings are configured correctly in the portlet web service editor.) **iii.** Test setting and changing preferences. (if there are errors, check that the preferences are uniquely named and that the preference and gateway settings are configured correctly in the portlet web service editor.)

iv. Test communication with the backend application. Confirm that actions executed through the portlet are completed correctly. (if there are errors, check the gateway configuration in the portlet web service editor.)

v. Test localized portlets in all supported languages. (if there are errors, make sure that the language files are installed correctly and are accessible to the portlet.) **vi.** If the portlet displays secure information or uses a password, use a tunnel tool to confirm that any secure information is not sent or stored in clear text.

Vii. If backwards compatibility is supported, test portlets in multiple versions of the portal.

Q40. What is Equivalence Partitioning?

Ans: Concepts: Equivalence partitioning is a method for deriving test cases. In this method, classes of input conditions called equivalence classes are identified such that each member of the class causes the same kind of processing and output to occur. In this method, the tester identifies various equivalence classes for partitioning. A class is a set of input conditions that are likely to be handled the same way by the system. If the system were to handle one case in the class erroneously, it would handle all cases erroneously.

Q41. Why Learn Equivalence Partitioning?

Ans: Equivalence partitioning drastically cuts down the number of test cases required to test a system reasonably. It is an attempt to get a good 'hit rate', to find the most errors with the smallest number of test cases.

DESIGNING TEST CASES USING EQUIVALENCE PARTITIONING

Ans: To use equivalence partitioning, you will need to perform two steps.

1. Identify the equivalence classes
2. Design test cases

STEP 1:

IDENTIFY EQUIVALENCE CLASSES Take each input condition described in the specification and derive at least two equivalence classes for it. One class represents the set of cases which satisfy the condition (the valid class) and one represents cases which do not (the invalid class) Following are some general guidelines for identifying equivalence classes: a) If the requirements state that a numeric value is input to the system and must be within a range of values, identify one valid class inputs which are within the valid range and two invalid equivalence classes inputs which are too low and inputs which are too high. For example, if an item in inventory can have a quantity of - 9999 to + 9999, identify the following classes:

1. One valid class: (QTY is greater than or equal to -9999 and is less than or equal to 9999). This is written as $(-9999 \leq QTY \leq 9999)$
2. The invalid class (QTY is less than -9999), also written as $(QTY < -9999)$
3. The invalid class (QTY is greater than 9999) , also written as $(QTY > 9999)$ b) If the requirements state that the number of items input by the system at some point must lie within a certain range, specify one valid class where the number of inputs is within the valid range, one invalid class where there are too few inputs and one invalid class where there are, too many inputs.

Q42. What are two types of Metrics?

Ans:

1. Process metrics: Primary metrics are also called as Process metrics. This is the metric the Six Sigma practitioners care about and can influence. Primary metrics are almost the direct output characteristic of a process. It is a measure of a process and not a measure of a high-level business objective. Primary Process metrics are usually Process Defects, Process cycle time and Process consumption.
2. Product metrics: Product metrics quantitatively characterize some aspect of the structure of a software product, such as a requirements specification, a design, or source code.

Q43. What is the Outcome of Testing?

Ans:

A stable application, performing its task as expected.

Q44. Why do you go for White box testing, when Black box testing is available?

Ans: A benchmark that certifies Commercial (Business) aspects and also functional (technical) aspects is objectives of black box testing. Here loops, structures, arrays,

conditions, files, etc are very micro level but they are Basement for any application, So White box takes these things in Macro level and test these things

Q45. What is Baseline document, Can you say any two?

Ans: A baseline document, which starts the understanding of the application before the tester, starts actual testing. Functional Specification and Business Requirement Document

Q46. Tell names of some testing type which you learnt or experienced?

Ans: Any 5 or 6 types which are related to companies profile is good to say in the interview,

1. Ad - Hoc testing
2. Cookie Testing
3. CET (Customer Experience Test)
4. Depth Test
5. Event-Driven
6. Performance Testing
7. Recovery testing
8. Sanity Test
9. Security Testing
10. Smoke testing
11. Web Testing

Q47. What exactly is Heuristic checklist approach for unit testing?

Ans: It is method of achieving the most appropriate solution of several found by alternative methods is selected at successive stages testing. The checklist Prepared to Proceed is called Heuristic checklist

Q48. What is a Data Guideline?

Ans: Data Guidelines are used to specify the data required to populate the test bed and prepare test scripts. It includes all data parameters that are required to test the conditions derived from the requirement / specification The Document, which supports in preparing test data are called Data guidelines

Q49. Why do you go for Test Bed?

Ans: When Test Condition is executed its result should be compared to Test result (expected result), as Test data is needed for this here comes the role of test Bed where Test data is made ready.

Q50. Why do we prepare test condition, test cases, test script (Before Starting Testing)?

Ans: These are test design document which are used to execute the actual testing Without which execution of testing is impossible, finally this execution is going to find the bugs to be fixed so we have prepare this documents.

Q51. Is it not waste of time in preparing the test condition, test case & Test Script?

Ans: No document prepared in any process is waste of time, That too test design documents which plays vital role in test execution can never be said waste of time as without which proper testing cannot be done.

Q52. How do you go about testing of Web Application?

Ans: To approach a web application testing, the first attack on the application should be on its performance behavior as that is very important for a web application and then transfer of data between web server and .front end server, security server and back end server.

Q53. What kind of Document you need for going for a Functional testing?

Ans: Functional specification is the ultimate document, which expresses all the functionalities of the application and other documents like user manual and BRS are also need for functional testing. Gap analysis document will add value to understand expected and existing system.

Q54. Can the System testing be done at any stage?

Ans: No, .The system as a whole can be tested only if all modules are integrated and all modules work correctly System testing should be done before UAT (User Acceptance testing) and Before Unit Testing.

Q55. What is Mutation testing & when can it be done?

Ans: Mutation testing is a powerful fault-based testing technique for unit level testing. Since it is a fault-based testing technique, it is aimed at testing and uncovering some specific kinds of faults, namely simple syntactic changes to a program. Mutation testing is based on two assumptions: the competent programmer hypothesis and the coupling effect. The competent programmer hypothesis assumes that competent programmers turn to write nearly "correct" programs. The coupling effect stated that a set of test data that can uncover all simple faults in a program is also capable of detecting more complex faults. Mutation testing injects faults into code to determine optimal test inputs.

Q56. Why it is impossible to test a program completely?

Ans: With any software other than the smallest and simplest program, there are too many inputs, too many outputs, and too many path combinations to fully test. Also, software specifications can be subjective and be interpreted in different ways.

Q57. How will you review the test case and how many types are there?

Ans: There are 2 types of review:

Informal Review: technical lead reviewing.

Peer Review: by a peer at the same organization (walkthrough? technical - inspection).

Or Reviews:

1. Management Review
2. Technical Review
3. Code Review
4. Formal Review (Inspections and Audits)
5. Informal Review (Peer Review and Code Review)

and coming to walk through....

objectives of Reviews:

1. To find defects in requirements.
2. To find defects in Design.
3. To identify deviations in any process and also provide valued suggestions to improve the process.

Q58. What do you mean by Pilot Testing?

Ans:

- Pilot testing involves having a group of end users try the system prior to its full deployment in order to give feedback on IIS 5.0 features and functions.
- Or
- Pilot Testing is a Testing Activity which resembles the Production Environment.
- It is Done Exactly between UAT and Production Drop.
 - Few Users who simulate the Production environment to continue the Business Activity with the System.
 - They Will Check the Major Functionality of the System before going into production. This is basically done to avoid the high-level Disasters.
 - Priority of the Pilot Testing Is High and Issues Raised in Pilot Testing has to be Fixed As Soon As Possible.

Q59. What is SRS and BRS in manual testing?

Ans: BRS is Business Requirement Specification which means the client who want to make the application gives the specification to software development organization and then the organization convert it to SRS (Software requirement Specification) as per the need of the software.

Q60. What is Smoke Test and Sanity Testing? When will use the Above Tests?

Ans: Smoke Testing: It is done to make sure if the build we got is testable or not, i.e to check for the testability of the build also called as "day 0" check. Done at the 'build level'

Sanity Testing: It is done during the release phase to check for the main functionalities without going deeper. Sometimes also called as subset of regression testing. When no rigorous regression testing is done to the build, sanity does that part by checking major functionalities. Done at the 'release level'

Q61. What is debugging?

Ans: Debugging is finding and removing "bugs" which cause the program to respond in a way that is not intended.

Q62. What is determination?

Ans: Determination has different meanings in different situations. Determination means a strong intention or a fixed intention to achieve a specific purpose. Determination, as a core value, means to have strong will power in order to achieve a task in life. Determination means a strong sense of self-devotion and self-commitment in order to achieve or perform a given task. The people who are determined to achieve various objectives in life are known to succeed highly in various walks of life.

Another way, it could also mean calculating, ascertaining or even realizing a specific amount, limit, character, etc. It also refers to a certain result of such ascertaining or even defining a certain concept.

It can also mean to reach at a particular decision and firmly achieve its purpose.

Q63. What is exact difference between Debugging & Testing?

Ans: Testing is nothing but finding an error/problem and its done by testers where as debugging is nothing but finding the root cause for the error/problem and that is taken care by developers.

Or

Debugging- is removing the bug and is done by developer.

Testing - is identifying the bug and is done by tester. **Q64.**

What is fish model can you explain?

Ans: Fish model explains the mapping between different stages of development and testing.

Phase 1

Information gathering takes place and here the BRS document is prepared.

Phase 2 Analysis takes place

During this phase, development people prepare SRS document which is a combination of functional requirement specification and system requirement specification. During this phase, testing people are going for reviews.

Phase-3 Design phase

Here HLD and LLD high level design document and low level design documents are prepared by development team. Here, the testing people are going for prototype reviews. **Phase-4**

coding phase

White box testers start coding and white box testing is being conducted by testing team.

Phase-5

testing phase

White box testing takes place by the black box test engineers.

Phase-6 release and maintenance.

Q65. What is Conformance Testing?

Ans: The process of testing that an implementation conforms to the specification on which it is based. Usually applied to testing conformance to a formal standard.

Q66. What is Context Driven Testing?

Ans: The context-driven school of software testing is flavor of Agile Testing that advocates continuous and creative evaluation of testing opportunities in light of the potential information revealed and the value of that information to the organization right now.

Q67. What is End-to-End testing?

Ans: Similar to system testing, the 'macro' end of the test scale involves testing of a complete application environment in a situation that mimics real-world use, such as interacting with a database, using network communications, or interacting with other hardware, applications, or systems if appropriate.

Q68. When the testing should be ended?

Ans: Testing is a never ending process, because of some factors testing May terminates. The factors may be most of the tests are executed, project deadline, test budget depletion, bug rate falls down below the criteria.

Q69. What is Parallel/Audit Testing?

Ans: Testing where the user reconciles the output of the new system to the output of the current system to verify the new system performs the operations correctly.

Q70. What are the roles of glass-box and black-box testing tools?

Ans: Black-box testing

It is not based on knowledge of internal design or code. Tests are based on requirements and functionality. Black box testing is used to find the errors in the following.

1. Interface errors
2. Performance errors
3. Initialization errors
4. Incorrect or missing functionality
5. Errors while accessing external database

Glass-box testing

It is based on internal design of an application code. Tests are based on path coverage, branch coverage, and statement coverage. It is also known as White Box testing.

1. White box test cases can check for;
2. All independent paths with in a module are executed atleast once
3. Execute all loops
4. Exercises all logical decisions

5. Exercise internal data structure to ensure their validity

Q71. What is your experience with change control? Our development team has only 10 members. Do you think managing change is such a big deal for us?

Ans: Whenever the modifications happening to the actual project all the corresponding documents are adapted on the information. So as to keep the documents always in sync with the product at any point of time

Q72. What is GAP ANALYSIS?

Ans: The gap analysis can be done by traceability matrix that means tracking down each individual requirement in SRS to various work products.

Q73. How do you know when your code has met specifications?

Ans: With the help of traceability matrix. All the requirements are tracked to the test cases. When all the test cases are executed and passed is an indication that the code has met the requirements.

Q74. At what stage of the life cycle does testing begin in your opinion?

Ans: Testing is a continuous process and it starts as and when the requirement for the project /product begins to be framed.

Requirements phase: testing is done to check whether the project/product details are reflecting clients ideas or giving an idea of complete project from the clients perspective (as he wished to be) or not.

Q75). What are the properties of a good requirement?

Ans: Requirement specifications are important and one of the most reliable methods of insuring problems in a complex software project. Requirements are the details describing an application's externally perceived functionality and properties. Requirements should be clear, complete, reasonably detailed, cohesive, attainable and testable.

Q76. How do you scope, organize, and execute a test project?

Ans: The Scope can be defined from the BRS, SRS, FRS or from functional points. It may be anything that is provided by the client. And regarding organizing we need to analyze the functionality to be covered and who will testing the modules and pros and cons of the application. Identify the number if test cases, resource allocation, what are the risks that we need mitigate all these come into picture.

Once this is done it is very easy to execute based on the plan what we have chalked out.

Q77. How would you ensure 100% coverage of testing?

Ans: We can not perform 100% testing on any application. but the criteria to ensure test completion on a project are:

1. All the test cases are executed with the certain percentage of pass.
2. Bug falls below a certain level
3. Test budget depleted
4. Dead lines reached (project or test)
5. When all the functionalities are covered in a test cases
6. All critical & high bugs must have a status of CLOSED

Q78. Do you go about testing a web application?

Ans: Ideally to test a web application, the components and functionality on both the client and server side should be tested. But it is practically impossible

The best approach to examine the project's requirements, set priorities based on risk analysis, and then determine where to focus testing efforts within budget and schedule constraints. To test a web application we need to perform testing for both GUI and client-server architecture.

Based on many factors like project requirements, risk analysis, budget and schedule, we can determine that what kind of testing will be appropriate for your project. We can perform unit n integration testing, functionality testing, GUI testing, usability testing, compatibility testing, security testing, performance testing, recovery testing and regression testing.

Q79. What are your strengths?

Ans: I'm well motivated, well-organized, good team player, dedicative to work and I've got a strong desire to succeed, and I'm always ready and willing to learn new information and skills.

Q80. When should you begin testing?

Ans: For any Project, testing activity will be there from starting onwards, After the Requirements gathering, Design Document (High and Low) will be prepared, that will be tested, whether they are confirming to requirements or not, Design then Coding- White box will be done, after the Build or System is ready, Integration followed by functional testing will be done, Till the product or Project was stable. After the product or project is stable, then testing will be stopped.

Q81. When should you begin test planning?

Ans: Test planning is done by test lead. As a test lead test planning begins when TRM is finalized by project manager and handover to the test lead. Here test lead have some responsibilities those are,

1. Testing team formation
2. identifying tactical risks
3. preparing test plan
4. Reviews on test plans

Q82. Would you like to work in a team or alone, why?

Ans: I would like to work in a team. Because the process of software development is like a relay race where many runners have to contribute in their respective laps. It is important because the complexity of work and degree of efforts required is beyond level of an individual.

Q83. When should testing Start in a project? Why?

Ans: Testing is a continuous activity carried out at every stage of the project. You first test everything that you get from the client. As tester (technical tester), my work will start as soon as the project starts.

Q84. Have you ever created a test plan?

Ans: This is just a sample answer - "I have never created any test plan. I developed and executed testcase. But I was involved/ participated actively with my Team Leader while creating Test Plans."

Q85. Define quality for me as you understand it

Ans: It is software that is reasonably bug-free and delivered on time and within the budget, meets the requirements and expectations and is maintainable.

Q86. What is the role of QA in a development project?

Ans: Quality Assurance Group assures the Quality it must monitor the whole development process. they are most concentration on prevention of bugs.

It must set standards, introduce review procedures, and educate people into better ways to design and develop products.

Q87. How involved where you with your Team Lead in writing the Test Plan?

Ans: As per my knowledge Test Member are always out of scope while preparing the Test Plan, Test Plan is a higher level document for Testing Team. Test Plan includes Purpose, scope, Customer/Client scope, schedule, Hardware, Deliverables and Test Cases etc. Test plan derived from PMP (Project Management Plan). Team member scope is just go through TEST PLAN then they come to know what all are their responsibilities, Deliverable of modules. Test Plan is just for input documents for every testing Team as well as Test Lead.

Q88. What processes/methodologies are you familiar with?

Ans:

Methodology

1. Spiral methodology
2. Waterfall methodology. these two are old methods.
3. Rational unified processing. this is from I B M and
4. Rapid application development. this is from Microsoft office.

Q89. What is globalization testing?

Ans: The goal of globalization testing is to detect potential problems in application design that could inhibit globalization. It makes sure that the code can handle all international support without breaking functionality that would cause either data loss or display problems.

Q90. What is base lining?

Ans: Base lining: Process by which the quality and cost effectiveness of a service is assessed, usually in advance of a change to the service. Base lining usually includes comparison of the service before and after the Change or analysis of trend information. The term Benchmarking is normally used if the comparison is made against other enterprises.

For example:

If the company has different projects. For each project there will be separate test plans. This test plans should be accepted by peers in the organization after modifications. That modified test plans are the baseline for the testers to use in different projects. Any further modifications are done in the test plan. Present modified becomes the baseline. Because this test plan becomes the basis for running the testing project.

Q91. Define each of the following and explain how each relates to the other: Unit, System and Integration testing.

Ans: Unit testing

it is a testing on each unit (program)

System testing

This is a bottleneck stage of our project. This testing done after integration of all modules to check whether our build meets all the requirements of customer or not. Unit and integration testing is a white box testing which can be done by programmers. System testing is a black box testing which can be done by people who do not know programming. The hierarchy of this testing is unit testing integration testing system testing

Integration testing: integration of some units called modules. the test on these modules is called integration testing (module testing).

Q92. Who should you hire in a testing group and why?

Ans: Testing is an interesting part of software cycle. and it is responsible for providing an quality product to a customer. It involves finding bugs which is more difficult and challenging. I wanna be part of testing group because of this.

Q93. What do you think the role of test-group manager should be? Relative to senior management? Relative to other technical groups in the company? Relative to your staff?

Ans: ROLES OF test-group manager INCLUDE

- Defect find and close rates by week, normalized against level of effort (are we finding defects, and can developers keep up with the number found and the ones necessary to fix?)
- Number of tests planned, run, passed by week (do we know what we have to test, and are we able to do so?)
- Defects found per activity vs. total defects found (which activities find the most defects?)
- Schedule estimates vs. actual (will we make the dates, and how well do we estimate?)
- People on the project, planned vs. actual by week or month (do we have the people we need when we need them?)
- Major and minor requirements changes (do we know what we have to do, and does it change?)

Q94. What criteria do you use when determining when to automate a test or leave it manual?

Ans: The Time and Budget both are the key factors in determining whether the test goes on Manual or it can be automated. Apart from that the automation is required for areas such as Functional, Regression, Load and User Interface for accurate results.

Q95. How do you analyze your test results? What metrics do you try to provide?

Ans: Test results are analyzed to identify the major causes of defect and which is the phase that has introduced most of the defects. This can be achieved through cause/effect analysis or Pareto analysis. Analysis of test results can provide several test matrices. Where matrices are

measure to quantify s/w, s/w development resources and s/w development process. Few matrices which we can provide are:

Defect density: $\frac{\text{total no of defects reported during testing}}{\text{size of project Test effectiveness} \times (t + uat)}$

where t: total no of defect recorded during testing

and UAT: total no of defect recorded during use acceptance testing

Defect removal efficiency(DRE): $(\frac{\text{total no of defect removed}}{\text{total no of defect injected}}) \times 100$

Q96. How do you perform regression testing?

Ans: Regression Testing is carried out both manually and automation. The automatic tools are mainly used for the Regression Testing as this is mainly focused repeatedly testing the same application for the changes the application gone through for the new functionality, after fixing the previous bugs, any new changes in the design etc. The regression testing involves executing the test cases, which we ran for finding the defects. Whenever any change takes place in the Application we should make sure, the previous functionality is still available without any break. For this reason one should do the regression testing on the application by running/executing the previously written test cases.

Q97. Describe to me when you would consider employing a failure mode and effect analysis

Ans :quality method that enables the identification and prevention of process or product errors before they occur. Failure modes and effects analysis (FMEA) is a disciplined approach used to identify possible failures of a product or service and then determine the frequency and impact of the failure.: FMEA (Failure Mode and Effects Analysis) is a proactive tool, technique and quality method that enables the identification and prevention of process or product errors before they occur. Failure modes and effects analysis (FMEA) is a disciplined approach used to identify possible failures of a product or service and then determine the frequency and impact of the failure.

Q98. What is UML and how to use it for testing?

Ans: The Unified Modeling Language is a third-generation method for specifying, visualizing, and documenting the artifacts of an object-oriented system under development From the inside, the Unified Modeling Language consists of three things:

1. A formal metamodel
2. A graphical notation
3. A set of idioms of usage

Q99. What you will do during the first day of job ?

Ans: In my present company HR introduced me to my colleagues. and i known the following things.

1. What is the organization structure?
2. What is the current project developing, on what domain etc.,

3. I will know to whom i have to report and what r my other responsibilities.

Q100. What is IEEE? Why is it important?

Ans: Organization of engineers Scientists and students involved in electrical, electronics, and related fields. It is important because it functions as a publishing house and standards-making body.

Q101. Define Verification and Validation. Explain the differences between the two.

Ans: Verification - Evaluation done at the end of a phase to determine that requirements are established during the previous phase have been met. Generally Verification refers to the overall s/w evaluation activity, including reviewing, inspecting, checking and auditing.

Validation: - The process of evaluating s/w at the end of the development process to ensure compliance with requirements. Validation typically involves actual testing and takes place after verification is complete.

Or

Verification: Whether we are building the product right?

Validation: Whether we are building the right product/System?

Q102. Describe a past experience with implementing a test harness in the development of software

Ans: Harness: an arrangement of straps for attaching a horse to a cart.

Test Harness: This class of tool supports the processing of tests by working it almost painless to

1. Install a candidate program in a test environment
2. Feed it input data
3. Simulate by stubs the behavior of subsidiary modules.

Q103. What criteria do you use when determining when to automate a test or leave it manual?

Ans: The Time and Budget both are the key factors in determining whether the test goes on Manual or it can be automated. Apart from that the automation is required for areas such as Functional, Regression, Load and User Interface for accurate results.

Q104. What would you like to do five years from now?

Ans: I would like to be in a managerial role, ideally working closely with external clients. I have worked in client-facing roles for more than two years and I enjoy the challenge of

keeping the customer satisfied. I think it's something I'm good at. I would also like to take on additional responsibility within this area, and possibly other areas such as Finally, I'd like to be on the right career path towards eventually becoming a Senior Manager within the company. I'm very aware that these are ambitious goals, however I feel through hard work and dedication they are quite attainable.

Q105. Define each of the following and explain how each relates to the other:

Unit, System, and Integration testing Ans:

- Unit system comes first. Performed by a developer.
- Integration testing comes next. Performed by a tester
- System testing comes last-Performed by a tester.

Q106. What is IEEE? Why is it important?

Ans: "Institute of Electrical & Electronic Engineers" Organization of engineers, scientists and students involved in electrical, electronics, and related fields. It also functions as a publishing house and standards-making body.

Q107. What is the role of QA in a company that produces software?

Ans: The role of the QA in the company is to produce a quality software and to ensure that it meets all the requirements of its customers before delivering the product.

Q108. How would you build a test team?

Ans:

Building a test team needs a number of factors to judge. Firstly, you have to consider the complexity of the application or project that is going to be tested. Next testing, time allotted levels of testing to be performed. With all these parameters in mind you need to decide the skills and experience level of your testers and how many testers.

Q109). In an application currently in production, one module of code is being modified. Is it necessary to re- test the whole application or is it enough to just test functionality associated with that module?

Ans: It depends on the functionality related with that module. We need to check whether that module is inter-related with other modules. If it is related with other modules, we need to test related modules too. Otherwise, if it is an independent module, no need to test other modules.

Q110). What are ISO standards? Why are they important?

Ans: ISO 9000 specifies requirements for a Quality Management System overseeing the production of a product or service. It is not a standard for ensuring a product or service is of quality; rather, it attests to the process of production, and how it will be managed and reviewed. **For ex a few:**

ISO 9000:2000

Quality management systems. Fundamentals and vocabulary

ISO 9000-1:1994

Quality management and quality assurance standards. Guidelines for selection and use ISO 9000-2:1997

Quality management and quality assurance standards. Generic guidelines for the application of ISO 9001, ISO 9002 and ISO 9003

ISO 9000-3:1997

Quality management and quality assurance standards. Guidelines for the application of ISO 9001:1994 to the development, supply, installation and maintenance of computer software

ISO 9001:1994

Quality systems. Model for quality assurance in design, development, production, installation and servicing

ISO 9001:2000

Quality management systems. Requirements

Q111. What is the Waterfall Development Method and do you agree with all the steps?

Ans: Waterfall approach is a traditional approach to the s/w development. This will work out if the project is a small one (Not complex). Real time projects need spiral methodology as SDLC. Some product based development can follow Waterfall, if it is not complex. Production cost is less if we follow waterfall method.

Q112. What is migration testing?

Changing of an application or changing of their versions and conducting testing is migration testing. Testing of programs or procedures used to convert data from existing systems for use in replacement systems.

Q113a. What is terminology? Why testing Necessary fundamental test process psychology of testing Testing Terminologies

Ans: Error: a human action that produces an incorrect result.

Fault: a manifestation of an error in software.

Failure: a deviation of the software from its expected delivery or service.

Reliability: the probability that the software will not cause the failure of the system for a specified time under specified conditions.

Q113b. Why Testing is Necessary

Ans: Testing is necessary because software is likely to have faults in it and it is better (cheaper, quicker and more expedient) to find and remove these faults before it is put into live operation. Failures that occur during live operation are much more expensive to deal with than failures that occur during testing prior to the release of the software. Of course other consequences of a system failing during live operation include the possibility of the software supplier being sued by the customers!

Testing is also necessary so we can learn about the reliability of the software (that is, how likely it is to fail within a specified time under specified conditions).

Q114. What is UAT testing? When it is to be done?

Ans: UAT stands for 'User acceptance Testing' This testing is carried out with the user perspective and it is usually done before a release

UAT stands for User Acceptance Testing. It is done by the end users along with testers to validate the functionality of the application. It is also called as Pre-Production testing.

Q115. How to find that tools work well with your existing system?

Ans: I think we need to do a market research on various tools depending on the type of application we are testing. Say we are testing an application made in VB with an Oracle Database, and then Win runner is going to give good results. But in some cases it may not, say your application uses a lots of 3rd party Grids and modules which have been integrated into the application. So it depends on the type of application u r testing.

Also we need to know what sort of testing will be performed. If u need to test the performance, u cannot use a record and playback tool, u need a performance testing tool such as Load runner.

Q116. What is the difference between a test strategy and a test plan?

TEST PLAN: IT IS PLAN FOR TESTING.IT DEFINES SCOPE, APPROACH, AND ENVIRONEMENT.

TEST STRATEGY: A TEST STRATEGY IS NOT A DOCUMENT.IT IS A FRAMEWORK FOR MAKING DECISIONS ABOUT VALUE.

Q117. What is Scenarios in term of testing?

Ans: Scenario means development. We define scenario by the following definition: Set of test cases that ensure the business process flows are tested from end to end. It may be independent tests or a series of tests that follow each other, each dependant on the output of the previous one. The term test scenario and test case are often used synonymously.

Q118. Explain the differences between White-box, Gray-box, and Black-box testing?

Ans: Black box testing Tests are based on requirements and functionality. Not based on any knowledge of internal design or code.

White box testing Tests are based on coverage of code statements, branches, paths, conditions. Based on knowledge of the internal logic of an application's code.

Gray Box Testing A Combination of Black and White Box testing methodologies, testing a piece of software against its specification but using some knowledge of its internal workings.

Q119. What is structural and behavioral Testing?

Ans: Structural Testing

It is basically the testing of code which is called white box testing.

Behavioral Testing

It is also called functional testing where the functionality of software is being tested. This kind of testing is called black box testing.

Structural Testing

It's a White Box Testing Technique. Since the testing is based on the internal structure of the program/code & hence it is called as Structural Testing.

Behavioral Testing:

It's a Black Box Testing Technique. Since the testing is based on the external behavior/functionality of the system /application & hence it is called as Behavioral Testing.

Q120. How does unit testing play a role in the development / Software lifecycle?

Ans: We can catch simple bugs like GUI, small functional Bugs during unit testing. This reduces testing time. Overall this saves project time. If developer doesn't catch this type of bugs, this will come to integration testing part and if it catches by a tester, this need to go through a Bug life cycle and consumes a lot of time.

Q121. What made you pick testing over another career?

Ans: Testing is one aspect which is very important in the Software Development Life Cycle (SDLC). I like to be part of the team which is responsible for the quality of the application being delivered. Also, QA has broad opportunities and large scope for learning various technologies. And of course it has lot more opportunities than the Development.

