1. What is the Best Moment in Your Testing Life ?

2. Do you remember your Manager appreciate you for your work. tell me any short incident ?

3. How you performed as a Trouble shooter in your Last Job..any event you remember ?

4. Which is the Best Module you ever Tested ? The Best ? and Why ?

5. Have you helped your Team in Risk Management ? How ? any Example ?

6. Tell me which one the Most Critical Bug you find in your Life ? What was the Severity ? How it influenced.

7. Which one the Best Work around you have suggest to solve big trouble for few time, that provided some relax to your Project Team (Does not delay the Release) ? Is it happen any Time ? If yes can you share it with Me ?

1.How to handle the low frequency issues during you testing ?  
2. How to coach a newer beginning in testing scope?  
3. How to improve skills designing test cases and make sure high coverage rate?

**Q1. What is the Process for creating a test script?**

**Step 1** – is to get a thorough understanding of the AUT –  
a) This could be by reading the requirement documents thoroughly  
b) In absence of docs, we could try to understand the any point of reference that we have – a previous version of the application or wire-frames or screenshots

**Step 2** – After understanding the requirements, we make a list of what are the areas in this application that will be have to be tested. In other words, we identify the test requirements. The focus in this step is to identify “What” to test. The outcome of this step is a list of [test scenarios](http://www.softwaretestinghelp.com/decision-table-test-case-design-technique/).

**Step 3** – Once we have the test scenarios, we concentrate next on “How” to test them.  This phase involves writing detailed steps about how to test a particular feature, what data to enter ([test data](http://www.softwaretestinghelp.com/tips-to-design-test-data-before-executing-your-test-cases/)) and what is the expected result.

Once these 3 steps are done, we are ready for testing.

**Q2. What are the fields in a bug report?**

**Following important fields should be included in a**[**good bug report**](http://www.softwaretestinghelp.com/sample-bug-report/)**:**

1. A unique ID
2. Defect description – a short describing what the bug is
3. Steps to reproduce – details about how to arrive at the error, exact test data, the time at which defect was found(if applicable) environment – any information that will help re-encounter the issue
4. Module/section of the application (if applicable)
5. Severity
6. Screenshot
7. Responsible QA – in case of any follow up questions regarding this issue

**Q3. How to test a customer facing software?**

With any application that we test, we are trying to see if a certain set of requirements are met by the application or not. But when it comes to a user facing site, apart from concentrating on functionality, we also have to look into few the usability features, may be performance and security aspects also to a certain extent.

**The first level of testing is** – Does the site satisfy its functional requirements. Example: if it is a loan management site, we need to look at – are the new customer able to apply for a loan, are the existing customer able to access their loan info, is the interest % applied to the loan amount correct, etc.

**The next level of testing is** – how easy is it to use the site, do the options make a logical sense and meet the expectations of the user or not. For example, if the user has to be pass 3-4 screens to submit the basic information they are going to be annoyed, so such issues have to be addressed. Another example, after entering username and password, the user might click on tab- which means the control should go to “Sign in” button, instead if it’s going to cancel, the user is going to be really annoyed and the experience of using the site is going to be compromised. Such issues have to be caught.

------------

[**Performance testing**](http://www.softwaretestinghelp.com/introduction-to-performance-testing-loadrunner-training-tutorial-part-1/) to the complete extent might not be in scope but simple situations like, how long does the search results take to be displayed and how much time does it take for the system to retrieve a customer info at the peak hour – these are some example of the kind of things we would want to keep an eye on.

[**Security**](http://www.softwaretestinghelp.com/category/security-testing/) – for sites where there is a secure login to access the site, the minimum functionality around it has to be tested. For example, if I leave the site idle for more than 10 minutes, is it auto logging out or not. Something as basic as that should be focused on.

**Q4. How to overcome the challenge of not having input documentation for testing?**

IF the detailed standard documentation like BRD and FSD are unavailable, the tester will have to depend on some point of reference.  
**a)** Screenshots  
**b)** A previous version of the application  
**c)** Wireframes …etc

Another factor that helps immensely, is to talk to the developers or the business analysts (when available) to get a confirmation on our understanding or clarifications in case of doubts.

When none of these situations work, we can just conceptualize the application based on our previous IT application experience and create the basic set of test scripts. When testing phase comes up, we can set up a portion of test cycle time and do some test case management (make the already created scripts perfect) so we have the doc for the next phases.

**Q5. How to get**[**maximum productivity**](http://www.softwaretestinghelp.com/how-to-improve-tester-performance/)**from offshore team?**

The key is to make sure that all the testers know about all the modules and that there is no knowledge concentration in one place. Involving everyone in test script peer reviews, defect meetings and KT sessions is going to ensure that everyone is aware of the application to the best extent possible.

Also, by encouraging the concept of team work we can have the team members collaborate, help and aid each other for better productivity.

Regular follow up meetings also help the process very much.

**Q6. What are the Roles and Responsibilities of an onsite coordinator? Does he/she test too?**

Onsite coordinator is a point of contact for the offshore team and to the client for any information regarding the testing engagement.

**This job includes:**

* KT from and to offshore and clients
* Getting the environment to test all ready
* Sanity testing, smoke testing
* Testing – the key functionality.
* Bug review – found by the offshore team
* Bug assigning to the respective dev
* Presenting metrics
* Providing sign off

Yes, even an onsite coordinator has to test.

**Q7. Inconsistent bugs – Why onsite can find it but offshore can’t and vice versa – How to handle this situation?**

Every bug has to be noted and analyzed – whether it is encountered at onsite or offshore, whether repeatable or not. A real value add to a tester’s job is when we involve ourselves in the Root Cause Analysis process for a bug rather than simply reporting it.

**Some of the ways we can handle this situation is:**  
**#1.** All the onsite and offshore team members should follow a guideline that screenshots had to be taken for every error that we encounter – repeatable or not

**#2.** If there are logs, system files or anything like that, that might help us find any evidence of the issue- we should try to find it

**#3.** Despite all these steps, if we still can’t tell why and when the problem occurs- we should report it to the developer all the same – with as much information as we can.

**Q8. Video/audio related testing – What does this include?**

How to test an application having video or audio?

**Here are the important points to consider:**  
– Access levels (restricted or not – password controlled)  
– Different kinds of environments  
– Browser compatibility  
– Screen resolutions  
– Internet connection speeds  
– The specific options on a video – like play, stop, mute etc.  
– Video by size  
– Response to the videos – comments (limitations on the comment length and number of comments it can take)  
– Video responses to the videos  
– Interface with social networking sites – interoperability  
– Buffering speed  
– Embedding the video

**Q9. Mobile Application Testing – What does it include briefly?**

**Mobile App Testing Important Test Scenarios:**  
– Check if the app works well with multiple carriers and multiple devices  
– Usability of the features in a mobile screen  
– Testing it in different mobile platforms – like Android and iOS  
– Installations, uninstalling, launching the app with network and without network, testing functionality  
– Network connections –WiFi, 2G, etc.  
– Logs at  iOS iPhone configuration utility  for Android Monitor.bat can be used for debugging

**Q-1. What Is Requirement Traceability Matrix?**

**Answer.**

Requirement Traceability Matrix (RTM) is a document which records the mapping between the high-level requirements and the test cases in the form of a table.

That’s how it ensures that the Test Plan covers all the requirements and links to their latest version.

**Q-2. Explain The Difference Between Pilot And Beta Testing?**

**Answer.** Read the following points to know the difference between Pilot and Beta testing.

**1.** We do the beta test when the product is about to release to the customer whereas pilot testing takes place in the earlier phase of the development cycle.  
**2.** In the beta test, testing application is given to few users to make sure that application meet the customer requirements and does not contain any showstopper bug. Whereas, in the pilot test, few members of the testing team work at the Customer site to set up the product. They give their feedback also to improve the quality of the end product.

**Q-3. Describe How To Perform Risk Analysis During Software Testing?**

**Answer.** Risk analysis is the process of identifying the hidden issues that may derail the successful delivery of the application. It also prioritizes the sequence of resolving the identified risks for testing purpose.

Following are some of the risks that are of concern to the QA.

**1.** New Hardware.  
**2.** New Technology.  
**3.** New Automation Tool.  
**4.** The sequence of code delivery.  
**5.** Availability of test resources for the application.

We prioritize them into three categories which are as follows.

**1.** High magnitude: Impact of the bug on the other functionality of the application.   
**2.** Medium: it is tolerable in the application but not desirable.  
**3.** Low: it is tolerable. This type of risk has no impact on the company business.

**Q-4. What Is Silk Test And Why Should You Use It?**

**Answer.** Here are some facts about the Silk tool.

**1.** It’s a tool developed for performing the regression and functionality testing of the application.   
**2.** It benefits when we are testing Window based, Java, the web, and the traditional client/server applications.  
**3.** Silk Test help in preparing the test plan and managing them to provide the direct accessing of the database and validation of the field.

**Q-5. What Is The Difference Between Master Test Plan And Test Plan?**

**Answer.** The difference between Master Plan and Test Plan can be described using following points.

**1.** Master Test Plan contains all the test scenarios and risks prone areas of the application. Whereas, Test Plan document contains test cases corresponding to test scenarios.  
**2.** Master Test Plan captures each and every test to be run during the overall development of application whereas test plan describes the scope, approach, resources and schedule of performing the test.  
**3.** MTP includes test scenarios to be executed in all the phases of testing that run during the complete life cycle of the application development. Whereas, a separate Test Plan exists for each phase of testing like Unit, Functional, and System which contains the test cases related to that type only.   
**4.** Only for big projects, we need a Master Test Plan which requires execution in all phases of testing. However, preparing a basic Test Plan is enough for small projects.

**Q-6. How Do You Handle A Non-Reproducible Bug?**

**Answer.** Following bugs lie under the non-reproducible category.

**1.** Defects observed due to low memory issue.   
**2.** Bugs raised due to address pointing to a memory location that does not exist.  
**3.** The race condition is an error scenario which occurs when the timing of one event impacts another executing in a sequence.

A tester can take the following actions to handle the non-reproducible bugs.

**1.** Execute test steps that are close to the error description.  
**2.** Evaluate the test environment.  
**3.** Examine and evaluate test execution results.  
**4.** Keep the resources & time constraints under check.

**Q-7. How Do You Perform Automated Testing In Your Environment?**

**Answer.** Automation Testing is a process of executing tests automatically. It reduces the human intervention to a great extent. We use different test automation tools like QTP, Selenium, and WinRunner. These tools help in speeding up the testing tasks.

Using the above tools we can create test scripts to verify the application automatically. After completing the test execution, these tools also generate the test reports.

**Q-8. What Are The Factors That You’ll Consider To Choose Automated Testing Over Manual Testing?**

**Answer.**

The choice of automated testing over manual testing depends on the following factors.

**1.** Tests require periodic execution.  
**2.** Tests include repetitive steps.  
**3.** Tests execute in a standard runtime environment.  
**4.** Automation is expected to take less time.  
**5.** Automation is increasing reusability.  
**6.** Automation reports are available for every execution.  
**7.** Small releases like service packs which include a minor bug fix. In such cases, regression type of cases is sufficient for validation.

**Q-9. What Is The Difference Between A Test Driver And Test Stub?**

**Answer.**

The **test driver** is a piece of code that calls a software component under test. It is useful in testing that follows the bottom-up approach.

**Test stub** is a dummy program that integrates with an application to complete its functionality. These are relevant for testing that uses the top-down approach.

Let’s take an example.

**1.** Let’s say there is a scenario to test the interface between modules A and B. We have developed only module-A. Then we can test module-A only if we have real module-B or a dummy module for it. In this case, we call module-B as the Test Stub.  
**2.** Now, module-B can’t send or receive data directly from module-A. In such scenario, we’ve to move data from one module to another using some external features called Test Driver.

**Q-10. What Are The Essential Qualities Of An Experienced QA Or Test Lead?**

**Answer.** Every QA or Test Lead should have the following qualities.

**1.** Well-versed in Software testing processes.  
**2.** Ability to accelerate teamwork to increase productivity.  
**3.** Improve coordination between QA and Dev engineers.  
**4.** Provide ideas to refine the QA processes.  
**5.** Ability to conduct RCA meetings and draw conclusions.  
**6.** Excellent written and interpersonal communication skills.  
**7.**Quick learner and able to groom the team members.

**Q-11. What Are The Different Types Of Software Testing?**

**Answer.** Following is the list of various testing types used by manual testers.

* Unit testing
* Integration testing
* Regression testing
* Shakeout testing
* Smoke testing
* Functional testing
* Performance testing
  + Load testing
  + stress testing
  + Endurance testing
* White box and Black box testing
* Alpha and Beta testing
* System testing

**Recommended –**[**Must Know Interview Questions for SSE/Test Lead**](http://www.techbeamers.com/must-know-qa-interview-questions-senior-engineer/)**.**

**Q-12. What Are The Key Elements Of A Test Plan?**

**Answer.** A test plan contains the following main points.

* Testing objectives.
* Test scope.
* Testing the frame.
* The environment
* Reason for testing
* The criteria for entrance and exit
* Deliverables
* Risk factors

**Q-13. What Is A Test Case?**

**Answer.**

A test case is a sequence of actions and observations that are used to verify the desired functionality. A good test case helps to identify problems in the requirements or design of an application.

**Q-14. What Is Agile Testing And Why Is It Important?**

**Answer.** Agile testing is a software testing process which evaluates a software from the customer point of view. And it is important because this does not require Dev to complete coding for starting QA. Instead, the coding and testing both goes hand in hand. However, it may require continuous customer interaction.

**Q-15. How Do You Test A Product If The Requirements Are Yet To Freeze?**

**Answer.** If the requirement spec is not available for a product, then a test plan can be created based on the assumptions made about the product. But we should get all assumptions well documented in the test plan.

**Q-16. How Will You Tell If Enough Test Cases Have Been Created To Test A Product?**

**Answer.** First of all, we’ll check if every requirement has at least one test case covered. If yes, then we can say that there are enough test cases to test the product.

**Q-17. What Will You Do When A Bug Turns Up During Testing?**

**Answer.** When a bug shows up, we can follow the below steps.

* Run more tests to make sure that the problem has a clear description.
* Run a few more tests to ensure that the same problem doesn’t exist with different inputs.
* Once we are sure of the full scope of the bug, then we can add details and report it.

**Q-18. If A Product Is In Production And One Of Its Modules Gets Updated, Then Is It Necessary To Retest?**

**Answer.** It is advisable to perform regression testing and run tests for all of the other modules as well. Finally, the QA should carry out the System testing.

**Q-19. What Is The Difference Between Functional Requirement And Non-Functional Requirement?**

**Answer.** The functional requirement specifies how a product should run whereas a non-functional requirement represents how it should be.

**Functional Requirements.**

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

**Non-Functional Requirements.**

* Performance
* Reliability
* Security
* Recovery
* Data Integrity
* Usability

**Q-20. How Comes The Severity And Priority Relate To Each Other?**

**Answer.**

* **Severity –** Represents the gravity/depth of the bug.
* **Priority –** Specifies which bug should get fixed first.
* **Severity –** Describes the application point of view.
* **Priority –** Defines the user’s point of view.

**Q-21. What Are Different Types Of Severity?**

**Answer.**The severity of a bug can be low, medium or high depending on the context.

* User Interface Defect – Low
* Boundary Related Defects – Medium
* Error Handling Defects – Medium
* Calculation Defects – High
* Misinterpreted Data – High
* Hardware Failures – High
* Compatibility Issues – High
* Control Flow Defects – High
* Load Conditions (Memory leakages under load testing) – High

**Q-22. What Is Entry And Exit Criteria In Software Testing?**

**Answer.**

**Entry criteria –** It is a process that should run when a system begins. It includes the following artifacts.

* SRS (Software Requirement Specification)
* FRS (Functional Requirement Specification)
* Use case
* Test-Case
* Test-plan

**Exit Criteria –** It signals when the testing should complete and when should the product be ready to release. It includes the following artifacts.

* Test Summary Report
* Metrics
* Defect Analysis report

**Q-23. What Is Test Strategy?**

**Answer.** Test strategy is an approach to carry out the testing activity. It covers the following.

* Roles and responsibilities for each member.
* Testing scope.
* Test tools.
* Test environment.
* Testing schedule.
* Associated risks.

**Q-24. What Is Smoke Testing And What Is Sanity?**

**Answer.**

**Smoke testing** confirms the basic functionality works for a product. It requires you to identify the most basic test cases for execution.

**Sanity testing,** on the other hand, ensures that the product runs without any logical errors. For example, if we are testing a calculator app; we may multiply a number by 3 and check whether the sum of the digits of the answer is divisible by 3.

**Q-25. What Is The Difference Between A Bug, Defect, And Error?**

**Answer.** A bug is usually same as the defect. Both of them represents an unexpected behavior of the software.

However, an error would also fall in the same category. But in some cases, errors are fixed values. For example – 404/405 errors in HTML pages.

**Also Read –**[**Software Testing Interview Questions and Answers – Part1**](http://www.techbeamers.com/software-testing-interview-questions-manual-testers-part1/)**.**

**Q-26. What Is The Difference Between High Level And Low-Level Test Case?**

**Answer.**

* High-level test cases cover the core functionality of a product like standard business flows.
* Low-level test cases are those related to user interface (UI) in the application.

**Q-27. What Is The Difference Between Static Testing And Dynamic Testing?**

**Answer.**

**Static Testing.**

* It is a white box testing technique which directs the developers to verify their code with the help of checklist to find errors in it.
* Developers can start it done without actually finalizing the application or program.
* Static testing is more cost effective than Dynamic testing.
* It covers more areas than Dynamic testing in a shorter time.

**Dynamic Testing.**

* Dynamic testing involves the execution of an actual application with valid inputs and checking of the expected output.
* Examples of Dynamic testing are Unit Testing, Integration Testing, System Testing and Acceptance Testing.
* Dynamic testing happens after the code deployment.
* It starts during the validation stage.

**Q-28. What Is Test Harness?**

**Answer.**

Test Harness requires configuring a set of tools and input data to test an application under various conditions. It involves monitoring the actual output with expected output for correctness.

Its benefits are as follows.

* Upward push in productivity due to process automation.
* Improve the overall product Quality.

**Q-29. What Is Defect Leakage?**

**Answer.**

Defect leakage occurs at the Customer or the End-user side after the product delivery. If the end user sees any issue in the application, then such bugs lead to Defect leakage. And this process of finding bugs is also called as Bug Leakage.

**Q-30. What Kind Of Document Will You Need To Begin Functional Testing?**

**Answer.**

* It is none other than the Functional specification document. It defines the full functionality of a product.
* Other documents are also useful in testing like user manual and BRS.
* Gap analysis is another document which can help in understanding the expected and existing system.

**Q-31. Beside Test Case & Test Plan, What Documents A Tester Should Produce?**

**Answer.** Here are a few other documents to prepare.

* Testing metrics
* Test design specs
* End-to-end scenarios
* Test summary reports
* Bug reports

**Q-32. What Is Business Requirements Document (BRD)?**

**Answer.** BRD provides a detailed business solution for a project including the documentation of customer needs and expectations.

BRD fulfills the following objectives.

* Gain agreement with stakeholders.
* Provide clarity on the business requirements.
* Describe the solution that meets the customer/business needs.
* Determine the input for the next phase of the project.

**Q-33. What Is Risk Analysis?**

**Answer.** Risk analysis is a technique to identify the things that can go wrong in a software development project. They can negatively impact the scope, quality, timeliness, and cost of a project.

However, everyone involved in the project has a part in minimizing the risk. But it’s the lead who ensures that whole team understands the individual role in managing the risk.

**Q-34. What Is Exploratory Testing?**

**Answer.**

Exploratory testing is a process which lets a tester to concentrate more on execution and less on planning.

* It requires formulating a test charter, a short declaration of the scope, set of objectives and possible approaches to be used.
* The test design and test execution activities may run in parallel without formally documenting the test conditions, test cases or test scripts.
* Testers can use boundary value analysis to concentrate the testing effort on error-prone areas by accurately pinpointing the boundaries.
* Notes should be recorded for the Exploratory Testing sessions as it would help to create a final report of its execution.

**Q-35. Can We Do System Testing At Any Stage?**

**Answer.** No. The system testing should start only if all modules arc in place and work correctly. However, it should happen before the UAT (User Acceptance testing).

**Q-36. Why Is It Impossible To Test A Program Completely?**

**Answer.**

Here are the two principal reasons that make it impossible to test a program entirely.

* Software specifications can be subjective and can lead to different interpretations.
* A software program may require too many inputs, too many outputs, and too many path combinations to test.

**Q-37. What Is The Primary Difference Between Debugging & Testing?**

**Answer.**

* Testing is to find out defects while using a product whereas debugging is to reach the part of the code causing failure.
* Debugging is isolating the problem area in the code done by a developer whereas Testing is identifying the bug in an application and done by a tester.

**Q-38. What Are The Roles Of Glass-Box And Black-Box Testing Tools?**

**Answer.**

**Black-Box Testing.**

It doesn’t require the knowledge of internal design or code. So the tests are based on requirements and functionality. Black box testing focuses on finding the following errors.

* Interface errors
* Performance errors
* Initialization errors
* Incorrect or missing functionality
* Errors in accessing external database

**Glass-Box Testing Or White-Box Testing.**

It requires familiarity with the internal design and application code. So the tests concentrate on path coverage, branch coverage, and statement coverage. It is expected to cover the following.

* All possible code flows of a module.
* Execute all loops.
* Exercise all logical decisions.
* Verify internal data structure to ensure their validity.

**Q-39. What Is GAP Analysis?**

**Answer.** Gap analysis reveals any deviation between the features available for testing and how the customer perceives them to be.

Traceability matrix is a testing tool which testers can use to track down the gaps.

**Q-40. How Do We Know The Code Has Met Specifications?**

**Answer.** Traceability matrix is an intuitive tool which ensures the requirements mapped to the test cases. And when the execution of all test cases finishes with success, it indicates that the code has met the requirements.

**Must Read –**[**Software Testing Interview Questions and Answers – Part2**](http://www.techbeamers.com/interview-questions-for-manual-testers-part2/)**.**

**Q-41. Is It Possible To Achieve 100% Coverage Of Testing? How Would You Ensure It?**

**Answer.** No, it’s not possible to perform 100% testing of any product. But you can follow the below steps to come closer.

* Set a hard limit on the following factors.
  + Percentage of test cases to be passed.
  + The no. of bug found.
* Set a red flag if,
  + Test budget depleted.
  + Deadlines breached.
* Set a green flag if,
  + The entire functionality gets covered in test cases.
  + All critical & high bugs must have a status of CLOSED.

**Q-42. What Are Error Guessing And Error Seeding?**

**Answer.**

**Error Guessing.**

It is a test case design technique in which testers have to guess the defects that might occur and write test cases to represent them.

**Error Seeding.**

It is the process of adding known bugs in a program for the tracking the rate of detection & removal. It also helps to estimate the number of faults remaining in the program.

**Q-43. What Is The Difference Between Coupling And Cohesion?**

**Answer.** The difference between coupling and cohesion is as follows.

* Cohesion is the degree which measures the dependency of the software component that combines related functionality into a single unit whereas coupling represents the binding of related functionality into a different unit.
* Cohesion deals with the functionality that relates to different process within a single module whereas coupling deals with how much one module is dependent on the other modules within the product.
* It is a good practice to increase the cohesion between the software whereas coupling is discouraged.

**Q-44. What Is CMM?**

**Answer.** The Capability Maturity Model for Software (CMM or SW-CMM) is a model for assessing the maturity of the software processes of an organization and for identifying the key practices that increase the maturity of these processes.

**Q-45. What Is Cause Effect Graph?**

**Answer.** It is a graphical representation of inputs and the associated outputs effects which assist in designing test cases.

**Q-46. Why Does Software Have Bugs?**

**Answer.**

* Miscommunication.
* Programming errors.
* Timeline pressures.
* Change in requirements.
* Software complexity.

**Q-47. What Is Ramp Testing?**

**Answer.** It is a testing method which proposes to raise an input signal until the system breaks down.

**Q-48. What Is Recovery Testing?**

**Answer.** It ensures that the program must recover from any expected or unexpected events without loss of data or functionality.

Events could be like shortage of disk space, unexpected loss of communication, or power out conditions.

**Q-49. What Is Inspection?**

**Answer.** It’s a group review quality improvement process for the product documents. It focuses on the following two aspects.

* Product document improvement.
* Process improvement (of both document production and inspection).

**Q-50. What Is Globalization Testing?**

**Answer.** Globalization testing concentrates on detecting the potential problems in the product design that could spoil globalization. It certifies that the code can handle the desired international support without breaking any functionality. And also, it ensures that there would be no data loss and display problems.

**1. What is the MAIN benefit of designing tests early in the life cycle?**   
It helps prevent defects from being introduced into the code.

**2. What is risk-based testing?**

Risk-based[Testing](http://www.guru99.com/software-testing.html)is the term used for an approach to creating a test strategy that is based on prioritizing tests by risk. The basis of the approach is a detailed risk analysis and prioritizing of risks by risk level. Tests to address each risk are then specified, starting with the highest risk first.

**3. A wholesaler sells printer cartridges. The minimum order quantity is 5. There is a 20% discount for orders of 100 or more printer cartridges. You have been asked to prepare test cases using various values for the number of printer cartridges ordered. Which of the following groups contain three test inputs that would be generated using Boundary Value Analysis?**

4, 5, 99

**4. What is the KEY difference between preventative and reactive approaches to testing?**

Preventative tests are designed early; reactive tests are designed after the software has been produced.

**5. What is the purpose of exit criteria?**

The purpose of exit criteria is to define when a test level is completed.

**6. What determines the level of risk?**

 The likelihood of an adverse event and the impact of the event determine the level of risk.

**7. When is used Decision table testing?**

Decision table testing is used for testing systems for which the specification takes the form of rules or cause-effect combinations. In a decision table the inputs are listed in a column, with the outputs in the same column but below the inputs. The remainder of the table explores combinations of inputs to define the outputs produced.

Learn More About Decision Table Testing Technique in the Video Tutorial [here](http://www.guru99.com/software-testing-techniques-1.html)

**8. What is the MAIN objective when reviewing a software deliverable?**

To identify defects in any software work product.

**9. Which of the following defines the expected results of a test? Test case specification or test design specification.**

Test case specification defines the expected results of a test.

**10. What is the benefit of test independence?**

It avoids author bias in defining effective tests.

**11. As part of which test process do you determine the exit criteria?**

The exit criteria is determined on the bases of 'Test Planning'.

**12. What is beta testing?**

Testing performed by potential customers at their own locations.

**13. Given the following fragment of code, how many tests are required for 100% decision coverage?**

**if width > length**

**thenbiggest\_dimension = width**

**if height > width**

**thenbiggest\_dimension = height**

**end\_if**

**elsebiggest\_dimension = length**

**if height > length**

**thenbiggest\_dimension = height**

**end\_if**

**end\_if**

4

**14. You have designed test cases to provide 100% statement and 100% decision coverage for the following fragment of code. if width > length then biggest\_dimension = width else biggest\_dimension = length end\_if The following has been added to the bottom of the code fragment above. print "Biggest dimension is " &biggest\_dimensionprint "Width: " & width print "Length: " & length How many more test cases are required?**

None, existing test cases can be used.

**15. Rapid Application Development?**

Rapid Application Development (RAD) is formally a parallel development of functions and subsequent integration. Components/functions are developed in parallel as if they were mini projects, the developments are time-boxed, delivered, and then assembled into a working prototype. This can very quickly give the customer something to see and use and to provide feedback regarding the delivery and their requirements. Rapid change and development of the product is possible using this methodology. However the product specification will need to be developed for the product at some point, and the project will need to be placed under more formal controls prior to going into production.

**16. What is the difference between Testing Techniques and Testing Tools?**

Testing technique: – Is a process for ensuring that some aspects of the application system or unit functions properly there may be few techniques but many tools.

Testing Tools: – Is a vehicle for performing a test process. The tool is a resource to the tester, but itself is insufficient to conduct testing

Learn More About Testing Tools  [here](http://www.guru99.com/testing-tools.html)

**17. We use the output of the requirement analysis, the requirement specification as the input for writing …**

User Acceptance Test Cases

**18. Repeated Testing of an already tested program, after modification, to discover any defects introduced or uncovered as a result of the changes in the software being tested or in another related or unrelated software component:**

Regression Testing

**19. What is component testing?**

Component testing, also known as unit, module and program testing, searches for defects in, and verifies the functioning of software (e.g. modules, programs, objects, classes, etc.) that are separately testable. Component testing may be done in isolation from the rest of the system depending on the context of the development life cycle and the system. Most often stubs and drivers are used to replace the missing software and simulate the interface between the software components in a simple manner. A stub is called from the software component to be tested; a driver calls a component to be tested.

Here is an awesome video on [Unit Testing](http://www.guru99.com/unit-testing-guide.html)

**20. What is functional system testing?**

Testing the end to end functionality of the system as a whole is defined as a functional system testing.

**21. What are the benefits of Independent Testing?**

Independent testers are unbiased and identify different defects at the same time.

**22. In a REACTIVE approach to testing when would you expect the bulk of the test design work to be begun?**

The bulk of the test design work begun after the software or system has been produced.

**23. What are the different Methodologies in Agile Development Model?**

There are currently seven different agile methodologies that I am aware of:

1. Extreme Programming (XP)
2. Scrum
3. Lean Software Development
4. Feature-Driven Development
5. Agile Unified Process
6. Crystal
7. Dynamic Systems Development Model (DSDM)

**24. Which activity in the fundamental test process includes evaluation of the testability of the requirements and system?**

A 'Test Analysis' and 'Design' includes evaluation of the testability of the requirements and system.

**25. What is typically the MOST important reason to use risk to drive testing efforts?**

Because testing everything is not feasible.

**26. What is random/monkey testing? When it is used?**

Random testing often known as monkey testing. In such type of testing data is generated randomly often using a tool or automated mechanism. With this randomly generated input the system is tested and results are analysed accordingly. These testing are less reliable; hence it is normally used by the beginners and to see whether the system will hold up under adverse effects.

**27. Which of the following are valid objectives for incident reports?**

1. Provide developers and other parties with feedback about the problem to enable identification, isolation and correction as necessary.
2. Provide ideas for test process improvement.
3. Provide a vehicle for assessing tester competence.
4. Provide testers with a means of tracking the quality of the system under test.

**28. Consider the following techniques. Which are static and which are dynamic techniques?**

1. Equivalence Partitioning.
2. Use Case Testing.
3. Data Flow Analysis.
4. Exploratory Testing.
5. Decision Testing.
6. Inspections.

Data Flow Analysis and Inspections are static; Equivalence Partitioning, Use Case Testing, Exploratory Testing and Decision Testing are dynamic.

**29. Why are static testing and dynamic testing described as complementary?**

Because they share the aim of identifying defects but differ in the types of defect they find.

**30. What are the phases of a formal review?**

In contrast to informal reviews, formal reviews follow a formal process. A typical formal review process consists of six main steps:

1. Planning
2. Kick-off
3. Preparation
4. Review meeting
5. Rework
6. Follow-up.

**31. What is the role of moderator in review process?**

The moderator (or review leader) leads the review process. He or she determines, in co-operation with the author, the type of review, approach and the composition of the review team. The moderator performs the entry check and the follow-up on the rework, in order to control the quality of the input and output of the review process. The moderator also schedules the meeting, disseminates documents before the meeting, coaches other team members, paces the meeting, leads possible discussions and stores the data that is collected.

Learn More about Review process in Video Tutorial [here](http://www.guru99.com/testing-review.html)

**32. What is an equivalence partition (also known as an equivalence class)?**

An input or output ranges of values such that only one value in the range becomes a test case.

**33. When should configuration management procedures be implemented?**

During test planning.

**34. A Type of functional Testing, which investigates the functions relating to detection of threats, such as virus from malicious outsiders?**

Security Testing

**35. Testing where in we subject the target of the test , to varying workloads to measure and evaluate the performance behaviours and ability of the target and of the test to continue to function properly under these different workloads?**

Load Testing

**36. Testing activity which is performed to expose defects in the interfaces and in the interaction between integrated components is?**

Integration Level Testing

**37. What are the Structure-based (white-box) testing techniques?**

Structure-based testing techniques (which are also dynamic rather than static) use the internal structure of the software to derive test cases. They are commonly called 'white-box' or 'glass-box' techniques (implying you can see into the system) since they require knowledge of how the software is implemented, that is, how it works. For example, a structural technique may be concerned with exercising loops in the software. Different test cases may be derived to exercise the loop once, twice, and many times. This may be done regardless of the functionality of the software.

**38. When "Regression Testing" should be performed?**

After the software has changed or when the environment has changed Regression testing should be performed.

**39**. **What is negative and positive testing?**

A negative test is when you put in an invalid input and receives errors. While a positive testing, is when you put in a valid input and expect some action to be completed in accordance with the specification.

**40. What is the purpose of a test completion criterion?**

The purpose of test completion criterion is to determine when to stop testing

**41. What can static analysis NOT find?**

For example memory leaks.

**42. What is the difference between re-testing and regression testing?**

Re-testing ensures the original fault has been removed; regression testing looks for unexpected side effects.

**43. What are the Experience-based testing techniques?**

In experience-based techniques, people's knowledge, skills and background are a prime contributor to the test conditions and test cases. The experience of both technical and business people is important, as they bring different perspectives to the test analysis and design process. Due to previous experience with similar systems, they may have insights into what could go wrong, which is very useful for testing.

**44. What type of review requires formal entry and exit criteria, including metrics?**

Inspection

**45. Could reviews or inspections be considered part of testing?**

Yes, because both help detect faults and improve quality.

**46. An input field takes the year of birth between 1900 and 2004 what are the boundary values for testing this field?**

1899,1900,2004,2005

**47. Which of the following tools would be involved in the automation of regression test? a. Data tester b. Boundary tester c. Capture/Playback d. Output comparator.**

d. Output comparator

**48. To test a function, what has to write a programmer, which calls the function to be tested and passes it test data.**

 Driver

**49. What is the one Key reason why developers have difficulty testing their own work?**

Lack of Objectivity

**50."How much testing is enough?"**

The answer depends on the risk for your industry, contract and special requirements.

**51. When should testing be stopped?**

It depends on the risks for the system being tested. There are some criteria bases on which you can stop testing.

1. Deadlines (Testing, Release)
2. Test budget has been depleted
3. Bug rate fall below certain level
4. Test cases completed with certain percentage passed
5. Alpha or beta periods for testing ends
6. Coverage of code, functionality or requirements are met to a specified point

**52. Which of the following is the main purpose of the integration strategy for integration testing in the small?**

The main purpose of the integration strategy is to specify which modules to combine when and how many at once.

**53.What are semi-random test cases?**

Semi-random test cases are nothing but when we perform random test cases and do equivalence partitioning to those test cases, it removes redundant test cases, thus giving us semi-random test cases.

**54. Given the following code, which statement is true about the minimum number of test cases required for full statement and branch coverage?**

**Read p**

**Read q**

**IF p+q> 100**

**THEN Print "Large"**

**ENDIF**

**IF p > 50**

**THEN Print "p Large"**

**ENDIF**

1 test for statement coverage, 2 for branch coverage

**55.  What is black box testing? What are the different black box testing techniques?**

Black box testing is the software testing method which is used to test the software without knowing the internal structure of code or program. This testing is usually done to check the functionality of an application. The different black box testing techniques are

1. Equivalence Partitioning
2. Boundary value analysis
3. Cause effect graphing

**56. Which review is normally used to evaluate a product to determine its suitability for intended use and to identify discrepancies?**

Technical Review.

**57. Why we use decision tables?**

The techniques of equivalence partitioning and boundary value analysis are often applied to specific situations or inputs. However, if different combinations of inputs result in different actions being taken, this can be more difficult to show using equivalence partitioning and boundary value analysis, which tend to be more focused on the user interface. The other two specification-based techniques, decision tables and state transition testing are more focused on business logic or business rules. A decision table is a good way to deal with combinations of things (e.g. inputs). This technique is sometimes also referred to as a 'cause-effect' table. The reason for this is that there is an associated logic diagramming technique called 'cause-effect graphing' which was sometimes used to help derive the decision table

**58. Faults found should be originally documented by whom?**

By testers.

**59. Which is the current formal world-wide recognized documentation standard?**

There isn't one.

**60. Which of the following is the review participant who has created the item to be reviewed?**

Author

**61. A number of critical bugs are fixed in software. All the bugs are in one module, related to reports. The test manager decides to do regression testing only on the reports module.**

Regression testing should be done on other modules as well because fixing one module may affect other modules.

**62. Why does the boundary value analysis provide good test cases?**

Because errors are frequently made during programming of the different cases near the 'edges' of the range of values.

**63. What makes an inspection different from other review types?**

It is led by a trained leader, uses formal entry and exit criteria and checklists.

**64. Why can be tester dependent on configuration management?**

Because configuration management assures that we know the exact version of the testware and the test object.

**65. What is a V-Model?**

A software development model that illustrates how testing activities integrate with software development phases

**66. What is maintenance testing?**

Triggered by modifications, migration or retirement of existing software

**67. What is test coverage?**

Test coverage measures in some specific way the amount of testing performed by a set of tests (derived in some other way, e.g. using specification-based techniques). Wherever we can count things and can tell whether or not each of those things has been tested by some test, then we can measure coverage.

**68. Why is incremental integration preferred over "big bang" integration?**

Because incremental integration has better early defects screening and isolation ability

**69. When do we prepare RTM (Requirement traceability matrix), is it before test case designing or after test case designing?**

It would be before test case designing. Requirements should already be traceable from Review activities since you should have traceability in the Test Plan already. This question also would depend on the organisation. If the organisations do test after development started then requirements must be already traceable to their source. To make life simpler use a tool to manage requirements.

**70. What is called the process starting with the terminal modules?**

Bottom-up integration

**71. During which test activity could faults be found most cost effectively?**

During test planning

**72. The purpose of requirement phase is**

To freeze requirements, to understand user needs, to define the scope of testing

**73. Why we split testing into distinct stages?**

We split testing into distinct stages because of following reasons,

1. Each test stage has a different purpose
2. It is easier to manage testing in stages
3. We can run different test into different environments
4. Performance and quality of the testing is improved using phased testing

**74. What is DRE?**

To measure test effectiveness a powerful metric is used to measure test effectiveness known as DRE (Defect Removal Efficiency) From this metric we would know how many bugs we have found from the set of test cases. Formula for calculating DRE is

DRE=Number of bugs while testing  / number of bugs while testing + number of bugs found by user

**75. Which of the following is likely to benefit most from the use of test tools providing test capture and replay facilities? a) Regression testing b) Integration testing c) System testing d) User acceptance testing**

Regression testing

**76. How would you estimate the amount of re-testing likely to be required?**

Metrics from previous similar projects and discussions with the development team

**77. What studies data flow analysis?**

The use of data on paths through the code.

**78. What is Alpha testing?**

Pre-release testing by end user representatives at the developer's site.

**79. What is a failure?**

Failure is a departure from specified behaviour.

**80. What are Test comparators?**

Is it really a test if you put some inputs into some software, but never look to see whether the software produces the correct result? The essence of testing is to check whether the software produces the correct result, and to do that, we must compare what the software produces to what it should produce. A test comparator helps to automate aspects of that comparison.

**81. Who is responsible for document all the issues, problems and open point that were identified during the review meeting**

Scribe

**82. What is the main purpose of Informal review**

Inexpensive way to get some benefit

**83. What is the purpose of test design technique?**

Identifying test conditions and Identifying test cases

**84. When testing a grade calculation system, a tester determines that all scores from 90 to 100 will yield a grade of A, but scores below 90 will not. This analysis is known as:**

 Equivalence partitioning

**85. A test manager wants to use the resources available for the automated testing of a web application. The best choice is**Tester, test automater, web specialist, DBA

**86. During the testing of a module tester 'X' finds a bug and assigned it to developer. But developer rejects the same, saying that it's not a bug. What 'X' should do?**

Send to the detailed information of the bug encountered and check the reproducibility

**87. A type of integration testing in which software elements, hardware elements, or both are combined all at once into a component or an overall system, rather than in stages.**

Big-Bang Testing

**88. In practice, which Life Cycle model may have more, fewer or different levels of development and testing, depending on the project and the software product. For example, there may be component integration testing after component testing, and system integration testing after system testing.**

V-Model

**89. Which technique can be used to achieve input and output coverage? It can be applied to human input, input via interfaces to a system, or interface parameters in integration testing.**

Equivalence partitioning

**90. "This life cycle model is basically driven by schedule and budget risks" This statement is best suited for…**

V-Model

**91. In which order should tests be run?**

The most important one must tests first

**92. The later in the development life cycle a fault is discovered, the more expensive it is to fix. Why?**

The fault has been built into more documentation, code, tests, etc

**93. What is Coverage measurement?**

It is a partial measure of test thoroughness.

**94. What is Boundary value testing?**

Test boundary conditions on, below and above the edges of input and output equivalence classes. For instance, let say a bank application where you can withdraw maximum Rs.20,000 and a minimum of Rs.100, so in boundary value testing we test only the exact boundaries, rather than hitting in the middle.  That means we test above the maximum limit and below the minimum limit.

**95. What is Fault Masking?**

Error condition hiding another error condition.

**96. What does COTS represent?**

Commercial off The Shelf.

**97.The purpose of which is allow specific tests to be carried out on a system or network that resembles as closely as possible the environment where the item under test will be used upon release?**

Test Environment

**98. What can be thought of as being based on the project plan, but with greater amounts of detail?**

Phase Test Plan

**99. What is exploratory testing?**

 Exploratory testing is a hands-on approach in which testers are involved in minimum planning and maximum test execution. The planning involves the creation of a test charter, a short declaration of the scope of a short (1 to 2 hour) time-boxed test effort, the objectives and possible approaches to be used. The test design and test execution activities are performed in parallel typically without formally documenting the test conditions, test cases or test scripts. This does not mean that other, more formal testing techniques will not be used. For example, the tester may decide to use boundary value analysis but will think through and test the most important boundary values without necessarily writing them down. Some notes will be written during the exploratory-testing session, so that a report can be produced afterwards.

**100. What is "use case testing"?**

In order to identify and execute the functional requirement of an application from start to finish "use case" is used and the techniques used to do this is known as "Use Case Testing"

**Bonus!**

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

SDLC deals with developement/coding of the software while STLC deales with validation and verification of the software

**102. What is traceability matrix?**

The relationship between test cases and requirements is shown with the help of a document. This document is known as traceability matrix.

**103. What is Equivalence partitioning testing?**

Equivalence partitioning testing is a software testing technique which divides the application input test data into each partition at least once of equivalent data from which test cases can be derived.  By this testing method it reduces the time required for software testing.

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

White box testing technique involves selection of test cases based on an analysis of the internal structure (Code coverage, branches coverage, paths coverage, condition coverage etc.)  of a component or system. It is also known as Code-Based testing or Structural testing.  Different types of white box testing are

1. Statement Coverage
2. Decision Coverage

**105.  In white box testing what do you verify?**

In white box testing following steps are verified.

1. Verify the security holes in the code
2. Verify the incomplete or broken paths in the code
3. Verify the flow of structure according to the document specification
4. Verify the expected outputs
5. Verify all conditional loops in the code to check the complete functionality of the application
6. Verify the line by line coding and cover 100% testing

**106. What is the difference between static and dynamic testing?**

Static testing: During Static testing method, the code is not executed and it is performed using the software documentation.

Dynamic testing:  To perform this testing the code is required to be in an executable form.

**107. What is verification and validation?**

Verification is a process of evaluating software  at development phase and to decide whether the product of a given  application satisfies the specified requirements. Validation is the process of evaluating software at the end of the development process and to check whether it meets the customer requirements.

**108. What are different test levels?**

There are four test levels

1. Unit/component/program/module testing
2. Integration testing
3. System testing
4. Acceptance testing

**109. What is Integration testing?**

Integration testing is a level of software testing process, where individual units of an application are combined and tested. It is usually performed after unit and functional testing.

**110. What are the tables in testplans?**

Test design, scope, test strategies , approach are various details that Test plan document consists of.

1. Test case identifier
2. Scope
3. Features to be tested
4. Features not to be tested
5. Test strategy & Test approach
6. Test deliverables
7. Responsibilities
8. Staffing and training
9. Risk and Contingencies

**111.  What is the difference between UAT (User Acceptance Testing) and System testing?**

System Testing: System testing is finding defects when the system under goes testing as a whole, it is also known as end to end testing. In such type of testing, the application undergoes from beginning till the end.

UAT: User Acceptance Testing (UAT) involves running a product through a series of specific  tests  which determines whether the product will meet the needs of its users.

**112. Mention the difference between Data Driven Testing and Retesting?**

**Retesting:**  It is a process of checking bugs that are actioned by development team to verify that they are actually fixed.

**Data Driven Testing (DDT):**In data driven testing process, application is tested with multiple test data. Application is tested with different set of values.

**113. What are the valuable steps to resolve issues while testing?**

* Record : Log and handle any problems which has happened
* Report: Report the issues to higher level manager
* Control: Define the issue management process

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

Difference between test scenarios and test cases is that

**Test Scenarios:**  Test scenario is prepared before the actual testing starts, it includes plans for testing product, number of team members, environmental condition, making test cases, making test plans and all the features that are to be tested for the product.

**Test Cases:**  It is a document that contains the steps that has to be executed, it has been planned earlier.

**Test Script:**It is written in a programming language and it's a short program used to test part of functionality of the software system. In other words a written set of steps that should be performed manually.

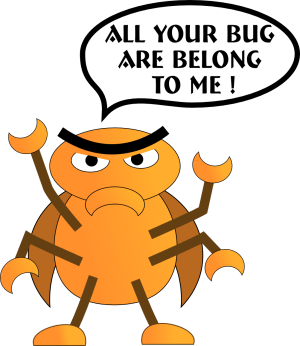
**115. What is Latent defect?**

**Latent defect:**This defect is an existing defect in the system which does not cause any failure as the exact set of conditions has never been met

**116. What are the two parameters which can be useful to know the quality of test execution?**

To know the quality of test execution we can use two parameters

* Defect reject ratio
* Defect leakage ratio

[](http://cdn.guru99.com/images/5-2015/Software_Testing_Interview_Questions.png)

**117. What is the function of software testing tool "phantom"?**

Phantom is a freeware, and is used for windows GUI automation scripting language.  It allows to take control of windows and functions automatically.  It can simulate any combination of key strokes and mouse clicks as well as menus, lists and more.

**118. Explain what is Test Deliverables   ?**

Test Deliverables are set of documents, tools and other components that has to be developed and maintained in support of testing.

There are different test deliverables at every phase of the software development lifecycle

* Before Testing
* During Testing
* After the Testing

**119. What is mutation testing?**

Mutation testing is a technique to identify if a set of test data or test case is useful by intentionally introducing various code changes (bugs) and retesting with original test data/ cases to determine if the bugs are detected.

**120. What all things you should consider before selecting automation tools for the AUT?**

* Technical Feasibility
* Complexity level
* Application stability
* Test data
* Application size
* Re-usability of automated scripts
* Execution across environment

**121. How will you conduct Risk Analysis?**

For the risk analysis following steps need to be implemented

a)      Finding the score of the risk

b)      Making a profile for the risk

c)       Changing the risk properties

d)      Deploy the resources of that test risk

e)      Making a database of risk

**122. What are the categories of debugging?**

Categories for debugging

a)      Brute force debugging

b)      Backtracking

c)       Cause elimination

d)      Program slicing

e)      Fault tree analysis

**123. What is fault masking explain with example?**

When presence of one defect hides the presence of another defect in the system is known as fault masking.

Example : If the "Negative Value" cause a firing of unhandled system exception,  the developer will prevent the negative values inpu. This will resolve the issue and hide the defect of unhandled exception firing.

**124. Explain what is Test Plan ? What are the information that should be covered in Test Plan ?**

A test plan can be defined as a document describing the scope, approach, resources and schedule of testing activities and a test plan should cover the following details.

* Test Strategy
* Test Objective
* Exit/Suspension Criteria
* Resource Planning
* Test Deliverables

**125. How you can eliminate the product risk in your project ?**

To eliminate product risk in your project, there is simple yet crucial step that can reduce the product risk in your project.

* Investigate the specification documents
* Have discussions about the project with all stakeholders including the developer
* As a real user walk around the website

**126. What are the common risk that leads to the project failure?**

The common risk that leads to a project failure are

* Not having enough human resource
* Testing Environment may not be set up properly
* Limited Budget
* Time Limitations

**127.  On what basis you can arrive to an estimation for your project?**

To estimate your project , you have to consider following points

* Divide the whole project into a smallest tasks
* Allocate each task to team members
* Estimate the effort required to complete each task
* Validate the estimation

**128. Explain how you would allocate task to team members ?**

|  |  |
| --- | --- |
| **Task** | **Member** |
| * Analyze software requirement specification | * All the members |
| * Create the test specification | * Tester/Test Analyst |
| * Build up the test environment | * Test administrator |
| * Execute the test cases | * Tester, Test administrator |
| * Report defects | * Tester |

**129. Explain what is testing type and what are the commonly used testing type ?**

To get an expected test outcome a standard procedure is followed which is referred as Testing Type.

Commonly used testing types are

* Unit Testing:  Test the smallest code of an application
* API Testing: Testing API created for the application
* Integration Testing: Individual software modules are combined and tested
* System Testing: Complete testing of system
* Install/UnInstall Testing: Testing done from the point of client/customer view
* Agile Testing: Testing through Agile technique

**130. While monitoring your project what all things you have to consider ?**

The things that has to be taken in considerations are

* Is you project on schedule
* Are you over budget
* Are you working towards the same career goal
* Have you got enough resources
* Are there any warning signs of impending problems
* Is there any pressure from management to complete the project sooner

**131. What are the common mistakes which creates issues ?**

* Matching resources to wrong projects
* Test manager lack of skills
* Not listening to others
* Poor Scheduling
* Underestimating
* Ignoring the small problems
* Not following the process

**132. What does a typical test report contains? What are the benefits of test reports?**

A test report contains following things:

* Project Information
* Test Objective
* Test Summary
* Defect

The benefits of test reports are:

* Current status of project and quality of product are informed
* If required, stake holder and customer can take corrective action
* A final document helps to decide whether the product is ready for release

**133. What is test management review and why it is important?**

Management review is also referred as Software Quality Assurance or SQA. SQA focusses more on the software process rather than the software work products.  It is a set of activities designed to make sure that the project manager follows the standard process.  SQA helps test manager to benchmark the project against the set standards.

**134. What are the best practices for software quality assurance?**

The best practices for an effective SQA implementation is

* Continuous Improvement
* Documentation
* Tool Usage
* Metrics
* Responsibility by team members
* Experienced SQA auditors

**135. When is RTM (Requirement Traceability Matrix) prepared ?**

RTM is prepared before test case designing.  Requirements should be traceable from review activities.

**136. What is difference between Test matrix and Traceability matrix?**

**Test Matrix**:  Test matrix is used to capture actual quality, effort, the plan, resources and time required to capture all phases of software testing

**Traceability Matrix**:Mapping between test cases and customer requirements is known as Traceability Matrix

**137. In manual testing what are stubs and drivers?**

Both stubs and drivers are part of incremental testing.  In incremental testing there are two approaches namely bottom up and top down approach. Drivers are used in bottom up testing and stub is used for top down approach. In order to test the main module, stub is used, whuich is a dummy code or program .

**138. What are the step you would follow once you find the defect?**

Once defect is found you would follow the step

a)      Recreate the defect

b)      Attach the screen shot

c)       Log the defect

**139. Explain what is "Test Plan Driven" or "Key Word Driven" method of testing?**

This technique uses the actual test case document developed by testers using a spread sheet containing special "key Words". The key words control the processing.

**140. What is DFD (Data Flow Diagram) ?**

When a "flow of data" through an information system is graphically represented then it is known as Data Flow Diagram.  It is also used for the visualization of data processing.

**141. Explain what is LCSAJ?**

LCSAJ stands for 'linear code sequence and jump'. It consists of the following three items

a)      Start of the linear sequence of executable statements

b)      End of the linear sequence

c)       The target line to which control flow is transferred at the end of the linear sequence

**142. Explain what is N+1 testing?**

The variation of regression testing is represented as N+1. In this technique the testing is performed in multiple cycles in which errors found in test cycle 'N' are resolved and re-tested in test cycle N+1.  The cycle is repeated unless there are no errors found.

**143. What is Fuzz testing and when it is used?**

Fuzz testing is used to detect security loopholes and coding errors in software.  In this technique random data is added to the system in attempt to crash the system.  If vulnerability persists, a tool called fuzz tester is used to determine potential causes. This technique is more useful for bigger projects but only detects major fault.

**144. Mention what are the main advantages of statement coverage metric of software testing?**

The benefit of statement coverage metric is that

a)      It does not require processing source code and can be applied directly to object code

b)      Bugs are distributed evenly through code, due to which percentage of executable statements covered reflects the percentage of faults discovered

**145. How to generate test cases for replace string method?**

a)      If characters in new string > characters in previous string.  None of the characters should get truncated

b)      If characters in new string< characters in previous string.  Junk characters should not be added

c)       Spaces after and before the string should not be deleted

d)      String should be replaced only for the first occurrence of the string

**146. How will you handle a conflict amogst your team members ?**

* I will talk individually to each person and note their concerns
* I will find solution to the common problems raised by team members
* I will hold a team meeting , reveal the solution and ask people to co-operate

**147. Mention what are the categories of defects?**

Mainly there are three defect categories

* **Wrong**: When requirement is implemented incorrectly
* **Missing**: It is a variance from the specification, an indication that a specification was not implemented or a requirement of the customer is not met
* **Extra**: A requirement incorporated into the product that was not given by the end customer. It is considered as a defect because it is a variance from the existing requirements

**148. Explain how does a test coverage tool works?**

The code coverage testing tool runs parallel while performing testing on the actual product. The code coverage tool monitors the executed statements of the source code. When the final testing is done we get a complete report of the pending statements and also get the coverage percentage.

**149. Mention what is the difference between a "defect" and a "failure" in software testing?**

In simple terms when a defect reaches the end customer it is called a failure while the defect is identified internally and resolved then it is referred as defect.

**150. Explain how to test documents in a project that span across the software development lifecycle?**

The project span across the software development lifecycle in following manner

* Central/Project test plan: It is the main test plan that outlines the complete test strategy of the project. This plan is used till the end of the software development lifecycle
* Acceptance test plan: This document begins during the requirement phase and is completed at final delivery
* System test plan: This plan starts during the design plan and proceeds until the end of the project
* Integration and Unit test plan: Both these test plans start during the execution phase and last until the final delivery

**151. Explain which test cases are written first black boxes or white boxes?**

Black box test cases are written first as to write black box test cases; it requires project plan and requirement document all these documents are easily available at the beginning of the project. While writing white box test cases requires more architectural understanding and is not available at the start of the project.

**152. Explain what is the difference between latent and masked defects?**

* **Latent defect:** A latent defect is an existing defect that has not caused a failure because the sets of conditions were never met
* **Masked defect:** It is an existing defect that has not caused a failure because another defect has prevented that part of the code from being executed

**153. Mention what is bottom up testing?**

Bottom up testing is an approach to integration testing, where the lowest level components are tested first, then used to facilitate the testing of higher level components. The process is repeated until the component at the top of the hierarchy is tested.

**154. Mention what are the different types of test coverage techniques?**

Different types of test coverage techniques include

* **Statement Coverage:** It verifies that each line of source code has been executed and tested
* **Decision Coverage:** It ensures that every decision in the source code is executed and tested
* **Path Coverage:** It ensures that every possible route through a given part of code is executed and tested

**155. Mention what is the meaning of breadth testing?**

Breadth testing is a test suite that exercises the full functionality of a product but does not test features in detail

**156. Mention what is the difference between Pilot and Beta testing?**

The difference between pilot and beta testing is that pilot testing is actually done using the product by the group of user before the final deployment and in beta testing we do not input real data, but it is installed at the end customer to validate if the product can be used in production.

**157. Explain what is the meaning of Code Walk Through?**

Code Walk Through is the informal analysis of the program source code to find defects and verify coding techniques

**158. Mention what are the basic components of defect report format?**

The basic components of defect report format includes

* Project Name
* Module Name
* Defect detected on
* Defect detected by
* Defect ID and Name
* Snapshot of the defect
* Priority and Severity status
* Defect resolved by
* Defect resolved on

**159. Mention what is the purpose behind doing end-to-end testing?**

End-to end testing is done after functional testing. The purpose behind doing end-to-end testing is that

* To validate the software requirements and integration with external interfaces
* Testing application in real world environment scenario
* Testing of interaction between application and database

**160. Explain what it means by test harness?**

A test harness is configuring a set of tools and test data to test an application in various conditions, it involves monitoring the output with expected output for correctness.

**161. Explain in a testing project what testing activities would you automate?**

In a testing project testing activities you would automate are

* Tests that need to be run for every build of the application
* Tests that use multiple data for the same set of actions
* Identical tests that needs to be executed using different browsers
* Mission critical pages
* Transaction with pages that do not change in short time