

## Software Testing Interview Questions-Latest Software Testing Questions with Answers updated on Jan 2025

### Latest Software Testing Questions with Answers

#### 1. 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.

#### 2. 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.

#### 3. Does automation replace manual testing?

Automation is the integration of testing tools into the test environment in such a manner that the test execution, logging, and comparison of results are done with little human intervention. A testing tool is a software application which helps automate the testing process. But the testing tool is not the complete answer for automation. One of the huge mistakes done in testing automation is automating the wrong things during development. Many testers learn the hard way that everything cannot be automated. The best components to automate are repetitive tasks. So some companies first start with manual testing and then see which tests are the most repetitive ones and only those are then automated.

As a rule of thumb do not try to automate:

1. Unstable software: If the software is still under development and undergoing many changes automation testing will not be that effective.
2. Once in a blue moon test scripts: Do not automate test scripts which will be run once in a while.
3. Code and document review: Do not try to automate code and document reviews; they will just cause trouble.

The following figure shows what should not be automated.

All repetitive tasks which are frequently used should be automated. For instance, regression tests are prime candidates for automation because they're typically executed many times. Smoke, load, and performance tests are other examples of repetitive tasks that are suitable for automation. White box testing can also be automated using various unit testing tools. Code coverage can also be a good candidate for automation.

#### 4. 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

### 5. How do you define a testing policy?

The following are the important steps used to define a testing policy in general. But it can change according to your organization. Let's discuss in detail the steps of implementing a testing policy in an organization.

**Definition:** The first step any organization needs to do is define one unique definition for testing within the organization so that everyone is of the same mindset.

**How to achieve:** How are we going to achieve our objective? Is there going to be a testing committee, will there be compulsory test plans which need to be executed, etc?.

**Evaluate:** After testing is implemented in a project how do we evaluate it? Are we going to derive metrics of defects per phase, per programmer, etc. Finally, it's important to let everyone know how testing has added value to the project?.

**Standards:** Finally, what are the standards we want to achieve by testing? For instance, we can say that more than 20 defects per KLOC will be considered below standard and code review should be done for it.

### 6. What is the MAIN benefit of designing tests early in the life cycle?

It helps prevent defects from being introduced into the code.

### 7. What is risk-based testing?

Risk-based testing 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.

### 8. 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.

### 9. 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

### 10. What is the difference between static and dynamic testing?

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

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

### 11. 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

### 12. 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.

### 13. What are the tables in test plans?

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

### 14. What is configuration management?

Configuration management is the detailed recording and updating of information for hardware and software components. When we say components we not only mean source code. It can be tracking of changes for

software documents such as requirement, design, test cases, etc.

When changes are done in adhoc and in an uncontrolled manner chaotic situations can arise and more defects injected. So whenever changes are done it should be done in a controlled fashion and with proper versioning. At any moment of time we should be able to revert back to the old version. The main intention of configuration management is to track our changes if we have issues with the current system. Configuration management is done using baselines.

#### **15. 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.

#### **16. How does a coverage tool work?**

While doing testing on the actual product, the code coverage testing tool is run simultaneously. While the testing is going on, the code coverage tool monitors the executed statements of the source code. When the final testing is completed we get a complete report of the pending statements and also get the coverage percentage.

#### **17. What is Fault Masking?**

Error condition hiding another error condition.

#### **18. What does COTS represent?**

COTS - Commercial off The Shelf.

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**

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

#### **Phase Test Plan**

#### **19. Should testing be done only after the build and execution phases are complete?**

In traditional testing methodology testing is always done after the build and execution phases. But that's a wrong way of thinking because the earlier we catch a defect, the more cost effective it is. For instance, fixing a defect in maintenance is ten times more costly than fixing it during execution.

In the requirement phase we can verify if the requirements are met according to the customer needs. During

design we can check whether the design document covers all the requirements. In this stage we can also generate rough functional data. We can also review the design document from the architecture and the correctness perspectives. In the build and execution phase we can execute unit test cases and generate structural and functional data. And finally comes the testing phase done in the traditional way. i.e., run the system test cases and see if the system works according to the requirements. During installation we need to see if the system is compatible with the software. Finally, during the maintenance phase when any fixes are made we can retest the fixes and follow the regression testing. Therefore, Testing should occur in conjunction with each phase of the software development.

## 20. 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

## 21. 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.

## 22. 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. 1 test for statement coverage, 2 for branch coverage

## 23. 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

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

Technical Review.

**25. 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

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

By testers.

**27. Are there more defects in the design phase or in the coding phase?**

The design phase is more error prone than the execution phase. One of the most frequent defects which occur during design is that the product does not cover the complete requirements of the customer. Second is wrong or bad architecture and technical decisions make the next phase, execution, more prone to defects. Because the design phase drives the execution phase it's the most critical phase to test. The testing of the design phase can be done by good review. On average, 60% of defects occur during design and 40% during the execution phase.

**28. 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.

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

Inspection

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

Yes, because both help detect faults and improve quality. To test a function, what has to write a programmer, which calls the function to be tested and passes it test data.

### **31. What is a test log?**

The IEEE Std. 829-1998 defines a test log as a chronological record of relevant details about the execution of test cases. It's a detailed view of activity and events given in chronological manner.

### **32. What does entry and exit criteria mean in a project?**

Entry and exit criteria are a must for the success of any project. If you do not know where to start and where to finish then your goals are not clear. By defining exit and entry criteria you define your boundaries. For instance, you can define entry criteria that the customer should provide the requirement document or acceptance plan. If this entry criteria is not met then you will not start the project. On the other end, you can also define exit criteria for your project. For instance, one of the common exit criteria in projects is that the customer has successfully executed the acceptance test plan.

### **33. What is the difference between verification and validation?**

Verification is a review without actually executing the process while validation is checking the product with actual execution. For instance, code review and syntax check is verification while actually running the product and checking the results is validation.

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

#### **a) Security Testing**

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?

#### **b) Load Testing**

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

#### **c) Integration Level Testing**

### **35. Can you explain process areas in CMMI?**

A process area is the area of improvement defined by CMMI. Every maturity level consists of process areas. A process area is a group of practices or activities performed collectively to achieve a specific objective. For instance, you can see from the following figure we have process areas such as project planning, configuration management, and requirement gathering.

### **36. 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.

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

Provide developers and other parties with feedback about the problem to enable identification, isolation and correction as necessary.

1. Provide ideas for test process improvement.
2. Provide a vehicle for assessing tester competence.
3. Provide testers with a means of tracking the quality of the system under test.

### **38. How does load testing work for websites?**

Websites have software called a web server installed on the server. The user sends a request to the web server and receives a response. So, for instance, when you type [www.google.com](http://www.google.com) the web server senses it and sends you the home page as a response. This happens each time you click on a link, do a submit, etc. So if we want to do load testing you need to just multiply these requests and responses "N" times. This is what an automation tool does. It first captures the request and response and then just multiplies it by "N" times and sends it to the web server, which results in load simulation.

So once the tool captures the request and response, we just need to multiply the request and response with the virtual user. Virtual users are logical users which actually simulate the actual physical user by sending in the same request and response. If you want to do load testing with 10,000 users on an application it's practically impossible. But by using the load testing tool you only need to create 1000 virtual users.

### **39. What is functional system testing?**

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

### **40. What kind of input do we need from the end user to begin proper testing?**

The product has to be used by the user. He is the most important person as he has more interest than anyone else in the project.

From the user we need the following data:

The first thing we need is the acceptance test plan from the end user. The acceptance test defines the entire test which the product has to pass so that it can go into production. We also need the requirement document from the customer. In normal scenarios the customer never writes a formal document until he is really sure of his requirements. But at some point the customer should sign saying yes this is what he wants.

The customer should also define the risky sections of the project. For instance, in a normal accounting project if a voucher entry screen does not work that will stop the accounting functionality completely. But if reports are not derived the accounting department can use it for some time. The customer is the right person to say



which section will affect him the most. With this feedback the testers can prepare a proper test plan for those areas and test it thoroughly.

The customer should also provide proper data for testing. Feeding proper data during testing is very important. In many scenarios testers key in wrong data and expect results which are of no interest to the customer.

#### **41. 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.

#### **42. What is a V-Model?**

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

#### **43. What is maintenance testing?**

Triggered by modifications, migration or retirement of existing software

#### **45. Can you explain the workbench concept?**

In order to understand testing methodology we need to understand the workbench concept. A Workbench is a way of documenting how a specific activity has to be performed. A workbench is referred to as phases, steps, and tasks as shown in the following figure.

There are five tasks for every workbench:

**Input:** Every task needs some defined input and entrance criteria. So for every workbench we need defined inputs. Input forms the first steps of the workbench.

**Execute:** This is the main task of the workbench which will transform the input into the expected

**Output.**

**Check:** Check steps assure that the output after execution meets the desired result.

**Production output:** If the check is right the production output forms the exit criteria of the workbench.

**Rework:** During the check step if the output is not as desired then we need to again start from the execute step.

#### **46. Can you explain the concept of defect cascading?**

Defect cascading is a defect which is caused by another defect. One defect triggers the other defect. For instance, in the accounting application shown here there is a defect which leads to negative taxation. So the negative taxation defect affects the ledger which in turn affects four other modules.

#### **47. Can you explain cohabiting software?**

When we install the application at the end client it is very possible that on the same PC other applications also exist. It is also very possible that those applications share common DLLs, resources etc., with your application. There is a huge chance in such situations that your changes can affect the cohabiting software. So the best practice is after you install your application or after any changes, tell other application owners to run a test cycle on their application.

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

#### **49. What is the difference between pilot and beta testing?**

The difference between pilot and beta testing is that pilot testing is nothing but actually using the product (limited to some users) and in beta testing we do not input real data, but it's installed at the end customer to validate if the product can be used in production.

#### **50. 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.

#### **51. 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.

#### **52. Can you explain data-driven testing?**

Normally an application has to be tested with multiple sets of data. For instance, a simple login screen, depending on the user type, will give different rights. For example, if the user is an admin he will have full rights, while a user will have limited rights and support if he only has read-only support rights. In this scenario the testing steps are the same but with different user ids and passwords. In data-driven testing, inputs to the system are read from data files such as Excel, CSV (comma separated values), ODBC, etc. So the values are read from these sources and then test steps are executed by automated testing.

#### **53. When should configuration management procedures be implemented?**

During test planning.

#### 54. What are the different strategies for rollout to end users?

There are four major ways of rolling out any project:

**Pilot** : The actual production system is installed at a single or limited number of users. Pilot basically means that the product is actually rolled out to limited users for real work.

**Gradual Implementation** : In this implementation we ship the entire product to the limited users or all users at the customer end. Here, the developers get instant feedback from the recipients which allow them to make changes before the product is available. But the downside is that developers and testers maintain more than one version at one time.

**Phased Implementation**: In this implementation the product is rolled out to all users in incrementally. That means each successive rollout has some added functionality. So as new functionality comes in, new installations occur and the customer tests them progressively. The benefit of this kind of rollout is that customers can start using the functionality and provide valuable feedback progressively. The only issue here is that with each rollout and added functionality the integration becomes more complicated.

**Parallel Implementation** : In these types of rollouts the existing application is run side by side with the new application. If there are any issues with the new application we again move back to the old application. One of the biggest problems with parallel implementation is we need extra hardware, software, and resources.

#### 55. What is the purpose of exit criteria?

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

#### 56. What determines the level of risk?

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

#### 57. 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.

#### 58. Can you explain tailoring?

As the name suggests, tailoring is nothing but changing an action to achieve an objective according to conditions. Whenever tailoring is done there should be adequate reasons for it. Remember when a process is defined in an organization it should be followed properly. So even if tailoring is applied the process is not bypassed or omitted.

#### 59. What is Six Sigma?

Six Sigma is a statistical measure of variation in a process. We say a process has achieved Six Sigma if the quality is 3.4 DPMO (Defect per Million Opportunities). It's a problem-solving methodology that can be applied to a process to eliminate the root cause of defects and costs associated with it.

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

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

**61. 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.

**62. What's the difference between System testing and Acceptance testing?**

Acceptance testing checks the system against the "Requirements." It is similar to System testing in that the whole system is checked but the important difference is the change in focus: System testing checks that the system that was specified has been delivered. Acceptance testing checks that the system will deliver what was requested. The customer should always do Acceptance testing and not the developer.

The customer knows what is required from the system to achieve value in the business and is the only person qualified to make that judgement. This testing is more about ensuring that the software is delivered as defined by the customer. It's like getting a green light from the customer that the software meets expectations and is ready to be used.

**63. 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.

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

It avoids author bias in defining effective tests.

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

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

**66. 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.

### **67. 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

### **68. Can you explain regression testing and confirmation testing?**

Regression testing is used for regression defects. Regression defects are defects occur when the functionality which was once working normally has stopped working. This is probably because of changes made in the program or the environment. To uncover such kind of defect regression testing is conducted.

The following figure shows the difference between regression and confirmation testing.

If we fix a defect in an existing application we use confirmation testing to test if the defect is removed. It's very possible because of this defect or changes to the application that other sections of the application are affected. So to ensure that no other section is affected we can use regression testing to confirm this.

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

There are currently seven different agile methodologies, they are :

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)

### **70. 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.

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

Because testing everything is not feasible.

**72. 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**.

**73. Can you explain requirement traceability and its importance?**

In most organizations testing only starts after the execution/coding phase of the project. But if the organization wants to really benefit from testing, then testers should get involved right from the requirement phase. If the tester gets involved right from the requirement phase then requirement traceability is one of the important reports that can detail what kind of test coverage the test cases have.

**74. 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.

**75. 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.

**76. 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.

**77. When Regression Testing should be performed?**

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

**78. 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.

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

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

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

For example memory leaks.

**81. 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.

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

Lack of Objectivity

**83. How much testing is enough?**

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

**84. 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.

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

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

**86. What are the different kinds of variations used in Six Sigma?**

Variation is the basis of Six Sigma. It defines how many changes are happening in the output of a process. So if a process is improved then this should reduce variations. In Six Sigma we identify variations in the process, control them, and reduce or eliminate defects.

**87. 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.

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

Because incremental integration has better early defects screening and isolation ability

**89. 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.

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

Bottom-up integration

**91. Explain Unit Testing, Integration Tests, System Testing and Acceptance Testing?**

**Unit testing** : Testing performed on a single, stand-alone module or unit of code.

**Integration Tests** : Testing performed on groups of modules to ensure that data and control are passed properly between modules.

**System testing** : Testing a predetermined combination of tests that, when executed successfully meets requirements.

**Acceptance testing** : Testing to ensure that the system meets the needs of the organization and the end user or customer (i.e. validates that the right system was built).

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



Metrics from previous similar projects and discussions with the development team. 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:**

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

#### **93. 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

#### **94. Does an increase in testing always improve the project?**

No an increase in testing does not always mean improvement of the product, company, or project. In real test scenarios only 20% of test plans are critical from a business angle. Running those critical test plans will assure that the testing is properly done. The following graph explains the impact of under testing and over testing. If you under test a system the number of defects will increase, but if you over test a system your cost of testing will increase. Even if your defects come down your cost of testing has gone up.

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

Normally black box test cases are written first and white box test cases later. In order to write black box test cases we need the requirement document and, design or project plan. All these documents are easily available at the initial start of the project. White box test cases cannot be started in the initial phase of the project because they need more architecture clarity which is not available at the start of the project. So normally white box test cases are written after black box test cases are written. Black box test cases do not require system understanding but white box testing needs more structural understanding. And structural understanding is clearer in the later part of project, i.e., while executing or designing. For black box testing you need to only analyze from the functional perspective which is easily available from a simple requirement document.

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**

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**

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.

**96. What is use case testing?**

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

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

The complete Verification and Validation of software is done in SDLC, while STLC only does Validation of the system. SDLC is a part of STLC.

**98. Describe software review and formal technical review (FTR).**

Software reviews works as a filter for the software process. It helps to uncover errors and defects in software. Software reviews enhance the quality of software. Software reviews refine software, including requirements and design models, code, and testing data.

A formal technical review (FTR) is a software quality control activity. In this activity, software developer and other team members are involved. The objectives of an FTR are:

1. Uncover the errors.
2. Verify that the software under technical review meets its requirements.
3. To ensure that the software must follow the predefined standards.
4. To make projects more manageable.

The FTR includes walkthroughs and inspections. Each FTR is conducted as a normal meeting. FTR will be successful only if it is properly planned, and executed.

**99. What are the attributes of good test case?**

The following are the attributes of good test case.

A good test has a high probability of finding an error. To find the maximum error, the tester and developer should have complete understanding of the software and attempt to check all the conditions that how the software might fail.

A good test is not redundant. Every test should have a different purpose from other, otherwise tester will repeat the testing process for same condition.

A good test should be neither too simple nor too complex. In general, each test should be executed separately. If we combine more than one test into one test case, it might be very difficult to execute. Sometimes we can combine tests but it may hide some errors.

**100. Describe cyclomatic complexity with example.**

Cyclomatic complexity is a software metric that measure the logical strength of the program. It was developed by Thomas J. McCabe. Cyclomatic complexity is calculated by using the control flow graph of the program. In the flow graph, nodes are represented by circle. Areas bounded by edges and nodes are called regions. When counting regions, we also include the area outside the graph as a region.

CAMPUSPRO