Manual Testing Interview Questions and Answers for Freshers

For a person who is looking to attend an interview on manual testing recently, here are some of the most standard interview questions and answers on manual testing fresher interview questions that will surely help you in the right way. At this juncture, we have a built-in list of the tricky qa interview questions and manual testing aptitude questions along with answers to help Fresher and experienced people to crack this interview.

1) . What is Software Testing?

Ans:- If we go by the ANSI/IEEE 1059 standards software testing is a procedure of breaking down the software to distinguish the contrasting characteristics among the existing software conditions and the required conditions (i.e. bugs and defects) and to assess the highlights of the software at hand.

2). Explain the procedure for manual testing

Ans:- The Manual Testing Process Comprises of the Following-

- Planning and Control
- Analysis and Design
- Implementation and Execution
- Evaluating exit criteria and Reporting
- Test Closure activities

3). Explain the tasks involved in planning and control.

Ans:- Test Planning Comprises of the Following Major Tasks:

- To fix the scope and the number of risks and ascertain the goals of testing.
- To govern the test method.
- To execute the test policy and/or test tactics.

4). What is Static Testing?

Ans:- Static Testing includes the process of exploring the records to recognize the imperfections in the very early stages of SDLC.

5). What is Dynamic Testing?

Ans:- Dynamic testing includes the process of execution of code. It validates and approves the output with the expected results.

6). How does quality control differ from quality assurance?

Ans:- Quality control: Quality control is a product-oriented technique to running a program to determine if it has any defects, in addition to making sure that the software program meets all of the requirements positioned forth by the stakeholders.

Quality assurance: Quality assurance is a process-oriented method that specializes in ensuring that the techniques, techniques, and processes used to create pleasant deliverables are carried out efficiently.

7). Why testing is required in software development?

Ans:- "**Software testing**" has become an increasingly popular methodology for software development.

Following are few compelling motives to show checking out is needed:

1. It factors out the defects and errors that have been made at some point of the improvement phases.
2. Reduces the coding cycles by means of identifying issues on the preliminary degree of the development.
3. Guarantees that software application calls for decrease maintenance cost and effects in greater correct, constant and reliable outcomes.
4. Checking out guarantees that the consumer unearths the agency dependable and their satisfaction inside the utility is maintained.
5. Make sure that software program is malicious program-free and the satisfaction of the product meets the marketplace fashionable.

8). What are the different types of testing?

Ans:- Software Testing is normally categorized into primary wide classes: Functional and Non-Functional Testing.

1. Functional Testing: Functional testing out involves the checking out of the functional aspects of a software program application. While you're acting Functional test, you have to check every and every capability. You need to look at whether you're getting the favored outcomes or not.

There are various Functional testing such as:

1. Unit testing
2. Integration testing
3. End-to-end testing
4. Smoke testing
5. Sanity testing
6. Regression testing
7. Acceptance testing
8. White box testing
9. Black box testing

10. Interface testing

Functional tests are carried out both manually and with the use of automation gear. For this kind of trying out, guide testing is straightforward, but you need to use equipment while vital.

Some tools that you could use for functional testing are Micro Focus UFT (previously known as QTP, and UFT stands for Unified functional checking out), Selenium, JUnit, soapUI, Watir, and so forth.

9) . What is Non-Functional Testing?

Ans:-Non-Functional testing is the testing of non-practical aspects of software, inclusive of overall performance, reliability, usability, security, and so forth. Non-Functional assessments are completed after the functional tests.

There are various Non-Functional testing such as:

1. Performance testing
2. Security testing
3. Load testing
4. Failover testing
5. Compatibility testing
6. Usability testing
7. Scalability testing
8. Volume testing
9. Stress testing
10. Maintainability testing
11. Compliance testing
12. Efficiency testing
13. Reliability testing
14. Endurance testing
15. Disaster recovery testing
16. Localization testing
17. Internationalization testing

10) . What is bug report?

Ans:-Throughout testing, a tester statistics their observations, findings, and different records beneficial to the developers or the control. All these facts belong to a check report, also referred to as a bug report.

a detailed bug report is a critical artifact produced at some point of trying out. It helps the team contributors with:

1. Recognize the trouble

2. Steps to reproduce the hassle,
3. The surroundings and the unique conditions below which it takes place.
4. The decision if/when the builders restore the trouble.

11) . What is the difference between Positive and Negative Testing?

Ans:-

Positive Testing	**Negative Testing:**
It is done to figure out what a framework is expected to do. It checks whether the application is defending the necessities it was built for or not.	It is to figure out what framework has b It finds the deformities from the produc

12). What is Use Case Testing?

Ans:- The use case testing uses the use case to assess the application. So that, the tester can inspect all the functionalities of the application. Use case testing can cover a whole application.

Read: Integration Testing Tutorial Guide for Beginner

13). What is the Test Case?

Ans:- A test case is ideally used to test the conformance of a developed application in consonance with its requirement stipulations. It is a set of settings with pre-requisites, input values, and predictable results in a recognized form.

14) . Explain the Tasks of Test Closure Activities.

Ans:- Test Closure Activities are Endowed with the following Major Tasks:

- To see which strategic deliverables are really delivered and to safeguard that all incident reports have been successfully resolved.
- To confirm and document the test ware such as writings, testing environments, etc. for future reuse.
- To deliver the test ware to the maintenance team. They will give sustenance to the software.
- To assess how the testing actually went and acquire lessons for upcoming releases and ventures.

15) . What is non-functional testing?

Ans:- Non-functional testing tests the device's non-functional requirements, which talk to a characteristic or fine of the machine explicitly asked by means of the client. These consist of overall performance, security, scalability, and usability.

Non-functional testing arrives after functional checking out. It tests the known characteristics irrelevant to the user requirements of the software program. Non-functional testing guarantees that the software is relaxed, scalable, has excessive performance, and won't crash under a hefty load.

16) . What different types of manual testing are there?

Ans:- Following are the different types on manual testing are:

1. Black Box Testing
2. White Box Testing
3. Unit Testing
4. System Testing
5. Integration Testing
6. Acceptance Testing

17) . What is Test-Driven-Development?

Ans:- Test-Driven-Development (TDD) is a popular software development approach, first introduced by way of Kent Beck in his ebook with the identical name, posted in 1999.

Test-Driven-Development is a design process where in you write tests before writing the code. You create a test, see if it fails and then write code to get the test to pass. This way you have a cycle where you only have working code that doesn't break any of your previous tests.

18) . What is cross-browser testing?

Ans:- In the world of Web development, cross-browser testing is crucial. By this important technique, you can find out if your website will work as expected on different browsers and devices. Cross-browser testing is performed with the aim of determining which browsers do not support the websites and how to design them better.

19). What is automated testing?

Ans:-Automated testing, which is also known as check automation, is the programmatic execution of the checks. The tester uses an automation tool or software like Selenium to write down code that performs the following obligations.

1. mechanically run the software.
2. Feed the entered statistics to the machine.
3. Look at the output with the expected outcome.
4. Fail the check if the consequences don’t match. In any other case, skip the take a look at.

As soon as a check is automated, you can run it as frequently as you want, to test if any new code has damaged it. It enables you to spend your time on other high-fee exams, such as exploratory trying out that helps locate insects that an automatic take a look at would omit.

Automated checking out is beneficial for repetitive trying out with inputs that don’t exchange frequently. people get worn-out and bored from conducting the same checks again and again

and seeing equal results. It's clean to make mistakes while you are checking out a feature for the twentieth time. software is a whole lot higher at doing repetitive duties without getting tired or making mistakes than a human operator could.

20). Can you List Down a few Characteristics of a Test Case?

Ans:- A Test Case can have the Following Attributes-

- **Test Case Id –** An exceptional identifier for the test case.
- **Test Summary** – Online comments or summary for each of the test cases.
- **Description** – A Comprehensive narrative of the test case.
- **Precondition or pre-requisite** – A set of fundamentals that must be charted before implementing the test steps.
- **Test Steps –** Comprehensive steps for carrying out the test case.
- **Expected result** – The estimated result in the direction to pass the test.
- **Actual result –** The actual result received after having done the execution of the test steps.
- **Test Result –** Pass/Fail to stand on the test execution.
- **Automation Status –** Identifier for automation – whether the given application is mechanized or not.
- **Date –** The date of the test execution.
- **Executed by –** Name of the person performing the test case.

21). How will you Define a Critical Bug?

Ans:- A critical bug is a bug that has got the tendency to affect a majority of the functionality of the given application and the application cannot be distributed to the end client deprived of the procedure of fixing that bug. It is different from a blocker bug as it doesn't essentially disturb or block the testing of other parts of the given application.

22). What is Endurance Testing?

Ans:- In this type of testing, we test the application's behavior in contrast to the load and stress put on over an application for a long period of time.

23). Why do we need Localization Testing?

Ans:-Localization testing generally deals with the functionality of the application and the GUI of the application.

24). What is Path Testing?

Ans:- Path testing is testing in which the tester guarantees that each path of the application should be affected at least once. In this testing, all the paths in the program's source code are tested in any case once for sure.

25. What are Software bugs?

Ans:-Software bugs can be frustrating to deal with but we can't really avoid them. In application there will always be some defects that occur unexpectedly. Bugs can cause your software to crash, or give wrong results or even worse, leave users' personal data vulnerable.

There are numerous causes for the bugs—for example, negative layout, sloppy programming, loss of model control, or miscommunication. at some stage in development, builders introduce hundreds or thousands of bugs inside the device. The purpose of the tester is to uncover all the bugs.

26). Write some common mistakes that lead to major issues.

Ans:- One of the key areas to focus on when you're manual testing is the common mistakes that lead to major issues. These include obvious mistakes such as typos and incorrect formatting, but also silly mistakes such as users re-using an ID or password because they didn't reset their password when they changed it.

To avoid these things following are the common mistakes you don’t have to do:

1. Negative Scheduling
2. Underestimating
3. Ignoring small issues
4. Not following the exact manner
5. wrong resource allocation

27). What is a user story?

Ans:- User story is a simple but powerful tool that a software tester uses while testing. A user story can be considered as 'a problem statement' that provides focus and direction to your testing activities. A user story is a verbal, brief description of what feature/functionality is being developed as per the requirement gathered from the customer/stakeholders.

28). List some of the popular testing tools/frameworks, providing a brief description of each.

Ans:-There are several different testing tools and frameworks available to you. The first thing to be aware of is that there isn't a single testing methodology — it's important to know why you're using a particular tool/framework. If a new technology comes along, try to test it a little before deciding whether it's worth integrating into your current project.

Following are the popular tools:

1. Selenium: an internet browser automation device that automates the test suites you want to run on a web browser.
2. Protractor: An give up-to-quit check framework for Angular and AngularJS applications. Protractor runs checks against your utility running in a real browser, interacting with it as a person might.
3. Cypress: A present day front-stop trying out tool built for the present day web. Although it’s much like Selenium and Protractor, it’s architecturally distinctive from them.

4. Jasmine: this is an open-supply JavaScript testing framework that lets you jot down behavior-pushed tests.
5. JUnit and NUnit: those are unit checking out frameworks for Java and C# programming languages, respectively.

29). What is a testbed in manual testing?

Ans:-Verification is a static analysis approach. Here, checking out is executed without executing the code. Examples: Reviews, Inspection, and walkthrough.

Validation is a dynamic analysis method in which checking out is carried out by means of executing the code. Examples include functional and non-functional checking out strategies.

30). What is the difference between Validation and Verification?

Ans:-

Verification	Validation
Progression of assessing work-products of a growth phase to control whether they fulfill the stated necessities for that stage.	The process of evaluating software du the development process to determine requirements.

31). Why is Software Testing Required?

Ans:- Software testing is a compulsory process that ensures that the software is completely safe. And, it is good enough to be released to the market. These are the compelling reasons to show why testing is needed:

- Testing points out the errors and issues made during the developing phase.
- It reduces the coding cycles by finding issues at the beginning of the development.
- Ensures that the software has no bugs and the product meets the market standard.
- It does not result in any failures.

32). What is a testbed in manual testing?

Ans:- The testbed is basically an environment configured for testing. This is an environment used for testing an application and the hardware and also any software required to run the program to be tested. It includes software, hardware, network configuration, and other related software.

33). Explain the procedure for manual testing?

Ans:- Here is the complete process of manual testing:

- Planning and Control
- Analysis and Design

- Implementation and Execution
- Evaluating exit criteria and Reporting
- Test Closure Activities

34). What is API testing?

Ans:- It is a kind of software testing where application programming interfaces are tested to figure out if they can meet the expectations to check the functionality, performance, reliability, and security. In simple terms, API testing is intended to reveal bugs, inconsistencies, or deviations from the expected behaviorus of an API.

35). What's the difference between a bug and a defect?

Ans:- variance among predicted effects and real results, detected through the developer after the product goes stay.

36). When is RTM (Requirement Traceability Matrix) prepared?

Ans:- RTM is ready before check case designing. Requirements ought to be traceable from overview activities.

37. What are the advantages of manual testing?

Ans:- Following are the advantages of manual testing:

1. Manual testing is the first stage in the software testing life cycle, and gives an overview of how a user actually interacts with a piece of software.
2. It's a cheaper way of checking out as compared to automatic checking out.
3. Analysis of the product from the point of view of the quiet-person is viable most effective with guide testing.
4. GUI trying out may be accomplished more appropriately with the help of a guide trying out as visual accessibility and options are difficult to automate.
5. East to research for brand spanking new people who've just entered into checking out.
6. It's rather suitable for quick-term tasks while test scripts are not going to be repeated and reused heaps of times.
7. Exceptionally desirable whilst the undertaking is at the early stages of its improvement.
8. Exceptionally dependable, considering automated tests can include mistakes and ignored bugs.

38). What is a test bed?

Ans:- Test bed is a set of hardware, software and other test items maintained to support the testing process. The purpose of the test bed is to control and monitor the conditions of the tests and to provide means to perform tests. In manual testing the test bed is made up of several tools and technologies. Examples include languages such as Perl or PHP, frameworks like Joomla or Wordpress, and databases such as PostgreSQL or MySQL.

39). What is a traceability matrix?

Ans:- Traceability matrix, also called traceability table, refers to the relationship between a requirement, its test cases and their results.

40). What is a Test script?

Ans:- Test script is a piece of software written in a programming language to test the various logical and functional parts of a software product under development, or finished product. Test scripts do not execute on a computer- they are executed by manual testers who follow the instructions provided by the test scripts.

41). What is defect density?

Ans:- defect density is the measure of the density of the defects in the machine. It can be calculated by way of dividing the variety of defects identified by way of the entire number of traces of code(or methods or instructions) inside the application or program.

42). What is defect priority?

Ans:- Defect Priority is the urgency of solving the illness. Normally the defect priority is about on a scale of P0 to P3 with the P0 defect having the most urgency to repair.

43). What is defect severity?

Ans:- Defect severity is the severity of the illness impacting the capability. primarily based on the organization, we will have distinct levels of disorder severity starting from minor to crucial or show stopper.

44). What is a blocker?

Ans:- blocker is a worm of excessive priority and excessive severity. It prevents or blocks trying out of some other essential part of the utility as nicely.

45). Explain use-case testing.

Ans:- Use case testing — (also known as use scenario testing) not only gives us a way to test the functionality of a particular piece of software, but it also helps us understand why we should or shouldn't be using software in the first place.

46). What are the differences between manual and automated testing?

Ans:-

Manual Testing	**Automated Testing**
A human tester checks the software program by way of manually running the take a look at cases and watching and comparing the real and expected outputs.	A tester or a programmer makes use o execute the software program and cor anticipated outputs.

Manual checking out isn't reproducible and repeatable.	Due to the fact it's miles programmed, continually reproducible and repeatabl as normal because the tester wishes.
For new capabilities, a tester can quickly test the characteristic manually, without plenty of configuration and setup.	To set up automated testing, there's th required to write down the assessmen surroundings to run those tests on.
Manual testing is useful for finding bugs in the person interface or accessibility problems.	Automatic testing is more appropriate that a human tester would leave out, a programming insects, business good j
Manual testing is vulnerable to human errors and is slow.	As there's no human participation wor checks), automated testing is extra de faster than manual checking out.

47). What is alpha testing?

Ans:- Alpha testing is conducted to check whether the software program is working fine. Alpha testing process is also known as developer testing. This process is carried out by the software developer himself. Sometimes alpha testing is also called internal testing. In alpha testing, defects and errors are identified, logged and reported. Alpha testing must be done before moving towards beta testing.

48). What is beta testing?

Ans:- Beta testing is any form of software testing that occurs after alpha testing and before the software is released. Beta tests include usability testing, performance testing, stress testing, time modeling, and other Quality Assurance techniques. Some of the most important goals of beta tests are finding bugs or errors in the software and ensuring that the software works correctly before allowing it to be used by customers.

49). What is exploratory testing?

Ans:- Exploratory testing is a research and development method used in software testing. It takes a test design approach to the process and is performed with the primary objective of finding errors or defects in the software product.

50). What is end-to-end testing?

Ans:- It's a term used to describe testing all aspects of the complete workflow: From Asking a User to Sign Up, Pay for their service, Use their service and even provide a rating (or review) of the service. If a user can't do all four of these things in sequence then your E2E tests have failed.

There's a crucial difference between end-to-end testing vs. different sorts of testing which can be greater remoted, which include unit trying out. In cease-to-cease trying out, the software is tested along with all its dependencies and integrations, consisting of databases, networks, document structures, and different outside services.

51). Explain the difference between alpha testing and beta testing.

Ans:- Alpha testing is a type of software testing performance to find bugs before releasing the product to real users or the public. This is a type of user acceptance testing. Beta testing is performed by real users of the software application in a real environment.

52). What's the difference between verification and validation in testing?

Ans:- Verification is a static analysis technique, here testing is done without executing the code. Validation is a dynamic analysis technique where testing is done through code execution.

53). What's the role of documentation in Manual Testing?

Ans:- Documentation is a key player in effective software testing. In this testing, all the details are documented including requirement specifications, designs, inspection reports, test plans, bug reports, etc.

Documentation helps you to estimate the testing effort you will need apart from the test coverage and tracking and tracing needs.

54). What are the different types of testing?

Ans:- Here is the list of different types of testing:

- System Testing
- Integration Testing
- Black Box Testing
- White Box Testing
- Unit Testing
- Acceptance Testing

55). What is the software testing life cycle?

Ans:- ***Here are the different phases involved in the software testing life cycle:***

- Research for Requirement: In the first step, the QA team understands the need in terms of what is tested and figured out the testable needs.
- Test Planning: This is the next step, where the test strategy is defined. Also, the ultimate project goal and scope are finalized.
- Test Case Development: Detailed test cases are defined and developed. The testing team also works on the test data for testing.
- Test Environment Setup: This is a setup of software as well as hardware for the testing teams to take care of test cases.
- Test Execution: This is the process of executing the code and doing a comparison of the expected and actual output.

- Test Cycle Closure: This last phase includes calling out the testing team member meeting and evaluating cycle completion criteria as per the test coverage, cost, quality, critical business goals, and software.

56). What is static software testing?

Ans:- Static software testing is the process by which an individual or a team tests a program without actually executing the program. The test cases are created from specifications or from a previous round of testing, or obtained from users. Flaws detected during static testing can be due to the requirements, design, or any earlier stage in the development cycle. The static procedure can be applied to hardware as well as computer programs.

57). What is dynamic software testing?

Ans:- Dynamic software testing is a method of testing with which the software is evaluated to determine its behavior under various controlled conditions that produce changes to the software's knowledge of its state and external world and which can be applied normally or abnormally.

58). What is meant by Code Coverage?

Ans:- In automation testing, “Code Coverage” is a measure of how much of the application source code is executed when the automated test script runs. It is useful in measuring progress in software testing. A well-designed test plan would have an objective or target Code Coverage number (for example, achieving 75% Code coverage). A business goal may also be set to define the percentage of coverage required in the test process for a release to go live.

59). What is Selenium and its benefits?

Ans:- Selenium is a browser automation framework that enhances the developer’s experience. It provides an extremely fast and simple testing engine to write automated functional tests for websites.

Here are some facts about Selenium and their benefits:

1. Their technology enables developers to write automated functional tests for their software, in a very short time frame, and in any programming language (e.g., Javascript, Java, Python, Ruby).
2. Most web browsers come with corresponding driver libraries that they have included in the distribution packages etc.
3. They can automate non-GUI web applications at the command line using shell-based tools.
4. They allow image comparison by running different images on multiple browsers.
5. Alerts can be generated when tests fail, this helps in identifying even minor bugs quickly.
6. Automated tests (implemented as Selenium WebDriver scripts) are processed by Cucumber under the command of Protractor and can notify users via email or cell phone apps.

7. Helpful features such as screenshot capturing from all your browser's help you understand what results are actually happening inside your browser.

60). What is boundary value analysis?

Ans:- Boundary value analysis is a technique used when making a mathematical determination of the boundary set between two variables. It is also known as end point or endpoint value.

For example when the programmer makes use of the greater-than operator (>) in place of the greater-than-or-equal-to (>=) operator, it causes the off-with- aid-of-one indexing errors.

61). What is unit testing?

Ans:- In software engineering, unit testing is a software testing method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage strategies, and operating strategies are tested to determine if they may be matched to be used.

62). What are the different HTTP status codes that a server can return?

Ans:- An HTTP status code is a 3-digit wide variety that shows the status of an incoming HTTP request, this is if the request has been completed or not.

A server can send the following five kinds of responses for an HTTP request.

1. **Information (100 - 199):** An HTTP status to those status codes provides a brief response. The response includes the reputation line and non-compulsory headers and terminates through an empty line.
2. **Success (200 - 299):**Indicate that the incoming HTTP request becomes effectively received, understood, and accepted.
3. **Redirect (300 - 399):** These popularity codes indicate in addition actions the consumer should take to satisfy the HTTP request. It could imply that the requested useful resource may have moved briefly or permanently. It may additionally redirect the customer to some other URL.
4. **A client error (400 - 499):** Imply trouble with the consumer who initiated the HTTP request.
5. **Server error (500 - 599):** The 5XX status code suggests a hassle on the server even as processing the request.

63). What is meant by browser automation?

Ans:- A browser automation tool is a program that automatically loads and navigates (browsing) through tabs of websites in normal ways. This means that the software does not alter or modify any of the data and contents on the website. It is a very common question for interviewers.

64). What is A/B testing?

Ans:- A/B testing is one of the most important techniques used today in software development to build better products and services.

A/B testing is a method of testing two different versions of a webpage (e.g. A 1 and A 2) on the same page to determine which version performs better in conversion rates.

You may choose the part of your customers to use function A. the alternative organization uses feature B. Then individual feedback and responses are evaluated by the use of statistical checking out to determine the final version of the feature.

Here are some more interview questions for manual testing.

65). What is the difference between Retesting and Regression Testing?

Ans:- Regression Testing: Regression testing means retesting that software to see if it still performs the same way. Most development teams need to perform regression testing before code changes are rolled out. This type of testing is an essential part of quality assurance.

Re-testing: Retesting means through regressing you get a new target that you can test against.

Regression	**Retesting**
It is performed to make sure that the adjustments haven't affected the unchanged part.	It is accomplished to ensure that the te filed in the closing execution are passe are fixed with the aid of developers.
It isn't finished for precise come across fixes.	It is also carried out based totally on d
It is most effective the previous model functionality-centric.	It is cutting-edge or previous model fu
It may be accomplished parallel with retesting.	It is had to perform before regression t
It does not include the verification of bugs.	It includes the verification of bugs.
In this type of testing, take a look at instances that may be automatic and the testing style is popular.	In this type of testing, take a look at ca automatic and the testing is accomplis
It is only used for handed take a look at cases.	It is best used for failed check cases.

66). What are the types of Integration Testing?

Ans: Following are the 3 types of Integration Testing:

1. Big bang testing
2. Bottom-Up Testing

3. Top-Down Testing

67). Name some of the most popular integration testing tools.

Ans:- Following are the most popular Integration Testing tools:

- DBUnit
- Greenmail
- Mockito
- REST-Assured
- JUnit 5
- H2 Database, etc.

68). What is functional testing?

Ans:- This is a form of black-box testing focused on the software's functional needs instead of its internal implementation. A functional need refers to required behaviour in the system, in terms of its output and input.

It validates the software against the functional needs, ignoring the non-functional attributes like usability, reliability, and performance.

69). Name two parameters that can be useful to check the quality of test execution.

Ans:- Following are the two parameters required to check the quality of test execution includes:

- Defect reject ratio: Ratio of total rejection to total production.
- Defect leakage ratio: Ratio of the total possibility of rejection occurrence to the total production

Manual Testing Interview Questions for Experienced

If you have some experience then this manual testing interview questions for 1 year experience will be best for you.

70). What is a Test Harness?

Ans:- A test harness is the gathering of software along with the test information arranged to test a program unit by running it under changing conditions which include checking the input values with the expected yield.

Read: Unit Testing Interview Questions and Answers

71). What is a Test Closure?

Ans:- Test Closure is the note arranged before the test group formally finishes the testing procedure. This note contains the aggregate no. of experiments, total no. of experiments

executed, total no. of imperfections discovered, add total no. of imperfections settled, total no. of bugs not settled, total no of bugs rejected, and so forth.

72). What is Top-Down Approach?

Ans:- Testing happens from top-to-bottom. High-level state modules are tested first and after that low-level modules and lastly incorporating the low-level modules to a high-level state to guarantee the framework is working as it is expected to. Stubs are utilized as an impermanent module if a module isn't prepared for integration testing.

73). What is the Bottom-Up Approach?

Ans:- It is the opposite of the Top-Down Approach. Testing happens from base levels to high-up levels. The lowest level modules are tried first and afterward high-level state modules and lastly coordinating the high-level state modules to a low level to guarantee the framework is filling in as it has been proposed to. Drivers are utilized as a transitory module for incorporation testing.

74). Is it True That We Can do System Testing at any Stage?

Ans:- No. The system testing must start only if all units are in place and are working properly. Though, it ought to happen before the UAT (User Acceptance testing).

75). What are the Experience-Based Testing Techniques?

Ans:- Inexperienced-based methods, individuals' information, abilities, and foundation knowledge are prime supporters of the test conditions and experiments. The experience of both technical, as well as business, is vital, as they convey alternate points of view to the test examination and configuration process. Because of past involvement with comparable frameworks, they may have bits of knowledge into what could turn out badly, which is exceptionally valuable for testing purposes.

76). Explain Configuration Testing.

Ans:- Configuration testing is a type of software testing that determines whether the product is configured properly for the user's environment. To answer this question, we have to understand what configuration management is. Configuration management is the monitoring and control of changes to a product or system. This can include assessing risk, identifying the changes and ensuring that an update meets quality standards. You may also hear it referred to as change management, especially in an enterprise environment.

77) . Explain Test Scenario.

Ans:- Test scenario also called a scenario test, is described as high-level particular documentation of test cases or use cases. In this, the tester checks the software application from an end-user perspective. It typically can serve as the idea for lower-level test cases or use case creation. Taking a look at a scenario is also known as a test condition or test possibility. It gives you an idea of what we need to test.

78). What is the defect life cycle?

Ans:- One of the most important aspects of software testing is the defect life cycle. There are four stages of a typical product life cycle. The first stage is when a product is new and is labeled as "ideal." The second stage is during the maturity phase, when products are evaluated for the purpose to improve them or re-design them. The next stage is when products have been

improved but have problems. This phase is called troubleshooting. Then, a phase called revalidation, where it's important to test out whether a given product meets all of its end-user needs.

79). When is it Ideal that the Testing is Stopped?

Ans:- It depends on the level of risks associated with the system being tested. There are some criteria based on which it is ok to stop testing.

- Closing date (Testing, Release)
- The testing budget has been exhausted
- Bug rate fall below the definite level
- Test cases finished with assured percentage passed
- Alpha or beta periods for testing ends
- Reporting of code, functionality, or necessities are met to a stated point

80). Explain the Concept of Semi-Random Test Cases?

Ans:- Semi-random test cases are those test cases that we get when we perform arbitrary experiments and do proportionality parceling to those experiments; it evacuates repetitive experiments, along these lines giving us semi-random test cases.

81). Why Do We Use Decision Tables?

Ans:- The techniques of equivalence dividing and boundary value analysis are regularly connected to the particular circumstances or sources of info. Nonetheless, if distinctive combinations of sources of info result in various actions being taken, this can be more difficult to indicate utilizing comparability apportioning and limiting esteem investigation, which has got a tendency to be more centered around the UI.

Read: Manual Testing Tutorials For Beginners

The other two determinations-based methods, choice tables, and state change testing are more centered around business rationale or business rules. A choice table is a decent method to manage blends of things (e.g. inputs). This procedure is once in a while additionally alluded to as a 'cause impact table. The purpose behind this is there is a related rationale charting system called 'cause-impact diagramming' which was some of the time used to help determine the decision table

82). Write the difference between smoke testing and sanity testing

Ans:- Smoke Testing: It is a type of testing executed to make sure that the extreme functionalities of this system are running nicely. It acts as an affirmation of whether the quality assurance team can in addition proceed with checking out or not.

Sanity Testing: It is an unscripted form of checking out achieved to make sure that the code modifications which can be made are running properly. It is achieved with the aid of the test team for a few basic checks. This testing makes a specialty of one or a few regions of capability and is usually narrow and deep.

83). What is meant by latent defect?

Ans:- A defect that is not obvious to the users. For example, in a web page which contains links, there are many possible values of the javascript option we could send with each link:We can create a list of these values and see which one turns out to be the optimal one and then send that value. If it happens often, it's recommended to tune this section immediately before deploying the site.

There are more manual testing interview questions for experienced. Check-out every question because all of them are very important.

84). What is the difference between static and dynamic testing?

Ans:- In web application testing, static testing is all about reviewing the code that you and others are writing, whereas with dynamic testing we are testing actual end user interaction with the website or a software application.

85). Why is it That the Boundary Value Analysis Provides Good Test Cases?

Ans:- This is for the reason that errors are often made during the program design of the different cases near the 'edges of the array of values.

86). What is Test Coverage?

Ans:- Test coverage assesses in some specific way the quantity of testing completed by a regular set of tests (derived in some other way, e.g. using requirements-based methods). Everywhere we can tally things and can tell whether or not each of those things has been verified by some test, then we can measure coverage.

87). Explain the Concept of Defect Cascading?

Ans:- Defect cascading is a defect that is triggered by a different defect in the same application. In this one, the defect beseeches the other defect in the application. When a defect is extant in any stage but is not recognized, hide to other stages without getting noted. This will affect an upsurge in the number of defects.

88). What is Regression Testing?

Ans:- Regression testing checks that alteration in code has not affected the operational functionality.

If you find this blog relevant, check out Manual Testing training by Janbask Training, JanBask Training is a dynamic, highly professional, and global online training provider committed to propelling the next generation of technology learners with a whole new way of training experience.

89). What are the valuable steps to resolve issues while testing?

Ans:- *Following are the valuable steps to resolve issues while testing:*

Record: Log and deal with any issues which have occurred

Report: document the issues to a higher stage supervisor

Control: outline the issue management method

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

Ans:- Following are the two parameters:

- Defect reject ratio
- Defect leakage ratio

91). What is mutation testing?

Ans: Mutation testing is a loosely defined software testing technique, which allows one to add small program mutations to a program and then execute the program with a test harness, evaluating the actual output of this system with the predicted output.

92). What are the categories of debugging?

Ans:- Categories for debugging

1. Brute force debugging
2. Backtracking
3. Cause elimination
4. Program Slicing
5. Fault tree analysis

93). What is the DFD (Data Flow Diagram)?

Ans:-The Data Flow Diagram (DFD) is a visual tool used to represent all the items (Entities, Processes, Events and Actions) in your design in order. In other words, the DFD is a simplified picture of our entire software application. If we can see what data flows through an FDD then we will be able to answer a lot of questions and test our logic.

94). Explain what the Test Plan is?

Ans:- Test plan is a document that mainly tells about the approach, test case and testing evidence of the testing process.

95). What is Meant by Baseline Testing?

Ans:- A baseline is an indicator of a particular benchmark that serves as a foundation of new creation.

96). What is Integration Testing?

Ans:- Integration testing is black-box testing. It focuses on the interface between the units to ensure that units work together to complete a specific task.

97. What is the difference between a bug, a defect, and an error?

Ans:- A bug is basically a fault in the software. It is generally found during testing time. Bugs occur due to some coding issues and lead a program to malfunction. They may also be the reason for functional issues in the product. These are fatal errors that can block the product functionality, results in a crash, or cause performance-related issues.

An error is a mistake on the software development part. Errors generally arise in software leading to a change in the overall functionality of the program.

A defect is a variance between the expected outputs and actual outputs. It is found by the developer after the product goes live. It is found after the application goes into production. A defect is also referred to as other issues with the software products, with its external behaviour, or with its internal features.

98). What is regression testing?

Ans:- Testing a previous test program to make sure that defects have not been introduced in unchanged areas of the software, the output of these changes made is referred to as Regression Testing.

99). Explain what LCSAJ is?

Ans:-LCSAJ stands for 'linear code sequence and jump.' It consists of the subsequent 3 items

- a) start of the linear collection of executable statements
- b) end of the linear collection
- c) The target line to which control go with the flow is transferred on the end of the linear sequence

100). What is the pesticide paradox? How to overcome it?

Ans:- When the same tests are done again and again, eventually the same tests cases will no longer find new bugs. Developers have to become extra careful in the places where testers found more defects and might not look into other areas. Here are the major methods to prevent pesticide paradox:

- To write a new set of test cases to exercise different parts of the software:
- To prepare new test cases and add them to the existing test cases.
- These are effective methods to find more defects in the area where defect numbers dropped.

101). What is black box testing, and what are the various techniques?

Ans:- Also known as specification-based testing, Black-Box Testing analyses the functionality of the software without knowing much about the internal structure of the item. The purpose of black-box testing is to check the functionality of the system as a whole to ensure that it is performing correctly and exceeding the user demands. Some of the techniques used in the black box testing:

- Equivalence Partitioning
- Decision Table Based Technique
- Boundary Value Analysis
- Cause-effect Graphing
- Use Case Testing

102). What is the difference between static testing and dynamic testing?

Ans:- Here is the list of major differences between static testing and dynamic testing

- **Static Testing:** This is a white box technique that includes the process of exploring the records to recognize the imperfections.
- **Dynamic Testing**: Dynamic testing refers to the process of executing code at the later stage of the software development lifecycle. It approves the results with the expected outputs.
- **Static Testing:** It is performed before the code deployment.
- **Dynamic Testing:** It is performed after the code deployment.
- **Static Testing:** It is implemented at the verification stage.
- **Dynamic Testing:** It begins during the validation stage.

103). What is risk-based testing?

Ans:- Risk-based testing is the check-point where you decide the tests to be performed during testing. There are many ways in which the risk factors can determine the risk-based testing. Hence, it is completely up to the person who chooses this strategy and how he implements it.

104). What is the KEY difference between preventative and reactive approaches to testing?

Ans:- Preventative tests are designed early; reactive assessments are designed after the software program has been produced.

105). What is the purpose of exit criteria?

Ans:- When the test level is completed then the purpose of exit criteria is defined.

106). What is ‘configuration management’?

Ans:- Generally, every high-functioning organization has a master plan on the detailed description of how they are supposed to operate and accomplish tasks. Software configuration management includes the processes, policies, and tools that help in organizing, controlling, coordinating, and tracking the code, documentation, issues, designs & tools, compilers, and libraries.

107). Explain STLC.

Ans:- STLC stands for Software Testing Life Cycle, it is a fundamental part of SDLC used to test software as well as make sure that the quality standards are met. Generally, it includes

verification activities and validation activities. In STLC, numerous activities are done in a specific order.

These are the six different phases in STLC Model:

- Requirement Analysis
- Test Planning
- Test Case Development
- Test Environment Setup
- Test Execution
- Test Cycle Closure

108). Explain Endurance Testing or Soak Testing?

Ans:- Endurance testing is a type of performance testing that is generally performed to check the performance of the system under constant use. The main purpose here is to determine whether a system can sustain a continuous high load or not.

109). What is the function of the software testing tool "phantom"?

Ans:- This freeware is used for windows, GUI automation scripting language. It allows you to take full control of windows and functions automatically. It is able to simulate any combination of keystrokes and mouse clicks and also menus, lists, and more.

110). Why developers shouldn't test the software they wrote?

Ans:-Here are major reasons to show why developers are poor testers:

- Developers test the code to ensure that it works instead of testing all the ways in which it does not work.
- Because they are the ones who wrote it themselves, developers tend to be optimistic about the software and do not get the right attitude required for testing to break software.

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

Ans:- The structure-based testing needs a profound knowledge of the code, it needs a profound knowledge of the code as it covers testing of some structural parts of the application. This testing is focused on enhancing security, checking the flow of inputs/outputs through the application, and improving the design and usability. Some of the major white-box techniques include:

- Condition Coverage
- Statement Coverage
- Multiple Coverage
- Decision Coverage

112). What determines the level of risk?

Ans:- The probability of an unfavorable event and the effect of the event decide the level of risk.

That's it for manual testing interview questions for experienced.

Bottom Line

Software testing is a key part of the development process, it ensures quality and provides the confidence to release the software to customers. This Q&A format guide and Manual Testing Interview Questions guide includes all important points, it is focused on helping you to prepare for your next interview as we; master the field of manual testing.

Because testing is the only element of a good software development process, the developing need to use high coding standards, best practices, as well as patterns to decrease the number of bugs. In the long-term strategy, the most efficient way to improve the testing process is to test frequently, measure the outputs, collect feedback and use it to get better results.

To sharpen up your skills and be industry-ready consider going for a Manual Testing Training Program that will get you the best manual-testing-tutorial, introduce you to a comprehensive curriculum, and gets you the right exposure.

That's it for now! Hopefully, you found this post helpful. If you have queries regarding the post, feel free to comment below. We will be glad to hear from you.