Manual testing interview questions uk

In this article you will be learning some tricky manual testing questions and answers along with examples that will help you crack the interview easily. The interview questions below will be very useful for beginners, intermediate and experienced testers. These questions are quite complex, so we need to answer them carefully. Manual Testing Tricky Interview Questions Followed or List Some Questions and Answers: Q #1) Define the following along with examples? A. Analysis of the limit value b. Equivalence testing c. Guess error d. Table check e. Control flow analysis response:1.a) Limit value analysis: test case/data selection process, defining the limits separating valid and invalid conditions. The tests are designed to check the inner and outer edges of these boundaries. In addition to the actual cut-out points or selection technique in which the test data is chosen to be found along the input domain [or output range] classes, the data structure, the procedural parameter, etc. The choice often includes maximum, minimum and trivial values or parameters. For example , input data from 1 to 10 (border value) Test input data from 0, 1, 2 to 9, 10, 111b. Equivalence testing: the system input domain is divided into classes of representative values so that the test cases cannot be limited to one class that meets the minimum no. the number of inspections to be completed. Example - valid data range: 1-10 Test set: 2; 5; 14=> click here to learn more about data analysis and equivalency testing.1.c) Guessing errors: Test data selection technique. The selection criterion is to choose values that appear to cause errors. The mention of errors is based mainly on experience, with some help from other methods, such as the analysis of border values. Based on experience, the test designer guesses the types of errors that might occur in certain types of software and develop test cases to detect them. For example, if any type of resource is assigned dynamically, a good place to look for errors is resource separation. Are all resources properly engagedlocated, or are some lost because the software executes?1.d) Desk Check: Desk check is performed by the developer system or programs. The process shall include a review of the whole product to ensure that it is structurally sound and that standards and requirements are met. This is the most traditional tool for analysing the system or program.1.e) Control flow analysis: It is based on the graphical representation of the program process. Control flow analysis; program charts have nodes that represent a statement or segment that may end with an unresolved node. The graph shows the program's control flow from one segment to another, as shown in the nodes. The purpose of control flow analysis is to identify possible problems in logical branches that may cause loop state or incorrect processing. We hope that questions will help you ace Manual Testing interview. Lominger interview questions are usually asked by job applicants to discuss the obstacles they have to overcome or tell stories in which they made business decisions and took special steps. Candidates in the workplace should then discuss the results of their actions, what they have learned from their choice and how they would apply their experience in other business scenarios. Lominger interview questions are designed to reveal the job candidates' competencies. Although it is not possible to know in advance what specific questions Lominger might ask, the job applicant may think about the situations to discuss in the interview. When preparing stories or answers to these questions, the candidate should think about what steps she took in her role, why she chose these actions, which she would do the same or otherwise in the future, the results of her actions, and what she learned from her experience. Lominger's questions are intended to assess the applicant's competence in certain business-related areas. Some of these areas include approach, ambition, team skills, conflict management, customer attention and delegation skills. Other competencies measured by the Lominger process are technical skills, listening, other motivation, organizing, political savvy, problem solving, effective team building and work-life balance. Lominger's interview process explores a total of 67 competencies. Frequently asked automation testing interview questions for beginners and top-level candidates: test automation plays a very important role throughout the software life cycle. Most of the time, when we want to prepare for an automation testing interview, we focus only on issues related to tools. However, we also need to consider the fact that learning and knowing the tool is just average, and that is not the main goal. Thus, when we are going to an automation tester interview, we need to consider Automation in general and focus on the system and the steps involved. We all know that software testing is a very important part of software development. But, with the fast-growing software development methodology and environment, it becomes difficult to manually check the entire application for a limited time along with cost limits. Thus, automation testing is growing rapidly in the market to accelerate the pace of development. This tutorial includes top interview questions about automation testing. I have tried to quote down short and quick issues that are very specific to automation in general and are not specific to any tool. Top 39 Automation testing Interview QuestionsWe are on the basic test of automation issues as well as some additional questions about intermediate to expert level candidates up to 2 to 5 years of experience. Q #1) What is automation? Answer: Automation is any action that can reduce people's Q #2) What is automation automation The process of using specific software tools or scripts to perform test tasks, such as entering data, performing test steps and comparing results, etc., is called automation testing. Q #3) What all things can you automate? Answer: Regression test kitSmoke/Sanity test suiteBuild placementTest data creationAutomating behind GUI, such as testing APIs and methods. Q #4) When is automation testing useful? Answer: Automation testing is useful for the following scenarios:a) Regression testing: In case of error correction or introduction of a new module, we must ensure that the functionality already in place or unchanged is not affected. In this case, we end up running the regression test case several times. For example: After each change request or bug fix, after each iteration incremental development approach, etc.b) Non-functional testing: testing of non-functional aspects of the application. For example, load testing or performance testing, etc. The automation of test cases in the above scenarios helps to achieve test speed and reduce human error. Q #5) How do you identify test cases that are suitable for automation? Answer: The most important step in identifying automation test cases is the automatic step. Q #6) Can you achieve 100% automation? Answer: 100% automation would be difficult to achieve because there would be many edge test cases and in some cases that are met rarely. Automating these instances that are not executed, which often bore will add value to the automated set. Q #7) How to decide what to use for automation testing in your projects? Answer: To define the automation testing tool in your project:a) Understand the project requirements carefully and identify the test scenarios that you want to automate.b) Look for a list of tools that support your project requirements.c) Identify your budget for the Automation tool. Select the tools in your budget.d) Specify whether you already have skillful tools. If you do not have the necessary qualified resources, then determine the cost of teaching existing resources or hiring new resources.e) Now compare each tool to key criteria such as:How easy is it to design and maintain scripts for the tool? Can a person who is not a technical person also perform test cases with little training? Does the tool support different types of platforms, such as web, mobile, desktop, etc. based on your project requirements? Does the tool have test report functionality? If not, is it an easy-to-configure tool? How is the cross-border support feature for web applications? How different types of testing can support this tool? How many languages does the tool support?f) Once you've compared the tools, choose the tool that's in your budget and support your project requirements, and give you more benefits based on the above key criteria. Q #8) There is currently no automation in my project, but now I want to introduce automation, what would be my steps? Answer: First, specify what kind of test/test cases you want to automate. Identify the Design Structure ToolCreate utility files and environment files. Start scriptingIdentify and work on reporting. Assigning time to improve and maintain scripts. The steps required to test automation for a project include:Understand the advantages and disadvantages of automation testing and identify appropriate test scenarios for automation. Select the automation tool that works best for automation of identified scenariosO open an expert tool to help you set up the tool and the necessary environment for executing test applications using the tool. Train a team so they can write scripts in the programming language that the tool supports. Create a test structure or identify an existing one that meets your requirements. Write a performance plan for OS, browsers, mobile devices, and so on. Write programming scripts for manual test cases to convert them to automated test cases. Report the status of a test case using the tool reporting tool. Keep scripts for changes or new features. Q #9) How do you decide which tool you should use? Answer: Concluding which tool is most suitable for a project requires a lot of ideas and discussion. Q #10) When you identify the tool, what would be your next steps? Answer: When we complete the tool, our next step would be to develop a system. Q #11) What is a system? Answer: The system is a set of structures for the entire automation suite. It is also a guideline which, if observed, can create a structure that is easy to maintain and improve. These guidelines include:Encoding standardsStatu processingSusususus and processing elements (object repository QTP)Processing environment files and property filesReporting dataHandling logsQ #12) What are the attributes of a good system? Answer: Features include: Modular: The system should be adaptable to change. Testers must be able to modify scripts according to the environment or changes to login information. Reusable: Commonly used methods or utilities must be written to a common file that is available to all scripts. Consistent: the kit must be written consistently, following all accepted coding practices. Whatever: Scripts must be written so that they are independent of each other. In case one test fails, it should not hold back the remaining test cases (unless it is a login page)Logger: It's good to have introduced logging in accordance with the framework. This would help in case our scripts run longer (say night mode) if the script fails at any time, having a log file will help us determine the location along with the type of error. Reporting: It's good if the reporting feature is automatically embedded in the system. Once the script is done, we can have the results and messages sent by email. Integration: The automation structure is such that it is easy to integrate with other applications, such as continuous integration or triggering an automated script as soon as the build is deployed. Q #13) Can you do without a system? Answer: Systems are guidelines, not mandatory rules, so we can do without a system, but if we create it and follow it, it would be easy to implement improvement and maintenance. Q #14) What are the different types of automation tool you know about? Answer: Open source tool like Selenium, JMeter, etc. Paid tools like QTP, Load Runner, Ranorex, RFT, and Rational Robot.Q #15) What is usually a structure within? Answer: Usually the structure must be - (It differs from the design) src (source) folder that has actual test scripts. Lib (library) folder that contains all libraries and common methods. A class folder that has all the file classes (in-case via java). Log folder that has log file(s). A file/folder that contains all The Ids.A Web Part that contains the URL, environment, and logon information. Q #16) Where do you keep information like URL, login, password? Answer: You should always save this information in a separate file. Q #17) Why would you want to bring this type of

information into a separate file instead of direct code? Answer: URLs, Login, and passwords are the kind of fields that are used very often, and these changes as for the environment and authorization. In case we hardcode it into our code, we must change it to each file that has its reference. In the case of more than 100 files, then it becomes very difficult to change all 100 files, and this, in turn, can cause errors. So this type of information is maintained in a separate file to make the update simple. Q #18) What are the different types of systems? Answer: Different types of systems include: Keyword-driven frameworkData-driven frameworkHybrid FrameworkLinear ScriptingQ #19) Can you tell some good coding practices while automating? Answer: Some good coding practices include: Add relevant comments. Identify the methods you want to reuse and type them in a separate file. Observe language coding conventions. Maintain test data in a separate file. Run scripts regularly. Q #20) Any kind of test that you think should not be automated? Answer: Tests that are rarely executed. Research testingUability testingTest, which is performed quickly when done manually. Q #21) Do you think testing can only be done at UI level? Answer: Today, as we move to neat mode, not limited to the UI layer. Early testimonials are imperial for a neat project. If we focus only on the UI layer, we are actually waiting until the UI is designed and available to test. Rather, we can check even before the UI is actually worked out. We can directly test API or methods using tools like cucumber and FitNesse.In this way, we give feedback a lot early and testing even before the UI is developed. Following this approach will help us test only the GUI aspect of small cosmetic changes or some about the UI and will help developers by giving more time to fix bugs. Q #22) How do you choose which automation tool is best for you? Answer: Choosing an automation tool depends on a variety of factors, such as:the scope of the application we want to automate. Management overheads, such as costs and budget. Time to learn and implement the tool. Type of support available for the tool. Limitation toolQ #23) What do you think there testers back to do automation? Is there a way to overcome it? Answer: The main obstacle for testers is to learn programming/coding if they want to automate. Since the testers do not encode, adaptation to coding is somewhat complicated for testers. We can overcome it: Collaborating with developers when automation. Given that automation is the responsibility of the whole team, not just the testers. Giving special time and focus on automation. Get the right management support. You can save these automation testing interview questions as pdf and print for further reading. Q #24) What is an automation testing system? Answer: The system, in general, is a set of guidelines. A set of guidelines, assumptions, concepts and coding practices for the creation of a performance environment in which tests will be automated is referred to as an automation testing system. The automation testing system is responsible for testing the harness with the mechanism to connect to the test application, to input from the file, to execute the test cases and to generate reports for the test. The automation testing system must be independent of the application and be easy to use, modify or extend. Q #25) What are the important modules for automation testing systems? Answer: The important modules in the automation testing system is:Test assertion tool: This tool will provide defending statements testing the expected values of the application in this test. Such as. Testng, Junit, etc. Data setup: Each test case must take user data either from the database or from the file, or embedded in the test script. The system data module should take care of the uptake of test script data and global variables. Build Management Tool: The framework must be created and despaced before you can create test scripts. Continuous integration tool: with CICD (continuous integration and continuous development) there are tool is needed to integrate and changes made to the system in each iteration. Reporting tool: A reporting tool is required to generate a readable report after you run test cases to better view the steps, results, and failures. Logging tool: The Logging tool helps you better debug errors and errors in your structure. Q #26) Explain some automation testing tools. Answer: Some of the famous automation testing tools are explained below:(i) Selenium: Selenium is a test system for testing web application automation. It supports multiple browsers and is OS independent. Selenium also supports a variety of programming languages such as Java, C#, PHP, Ruby, and Perl, etc. Selenium is a set of open source libraries that can be used to develop additional test systems or test scripts for web application testing. (ii) UFT: Unified functional testing is a licensed tool for functional testing. It provides a wide range of features such as API, web services, etc. and also supports multiple platforms such as desktops, internet and mobile. UFT scripts are written in a visual basic scripting language. (iii) Appium: Appium is an open source mobile app testing tool. It is used to automate testing for cross-platform, native, hybrid, and web-based mobile applications. Appium automates any mobile app from any language with full access to the API and smi from the test code. Appium is based on client-server architecture and has evolved from selenium. (iv) Cucumber: Cucumber is an open source behavior-driven development tool. It is used for web application automation testing and supports languages such as ruby, java, scala, groovy, etc. Cucumber reads the executable specification, written in plain text, and checks this specification test. In order for cucumber to understand plain text scenarios, we need to follow some basic syntax rules that are known as Gherkin. v) TestComplete: TestComplete is a licensed automated UI testing tool to test the application on various platforms such as desktops, the web, mobile devices, etc. It provides flexibility to record a test case in one browser and run it on multiple browsers and thus supports cross-browser testing. TestComplete is a built-in object recognition algorithm that uniquely identifies an object and stores it in a repository. Q #27) What are the different types of testing system methods? Answer: There are four types of automation testing system methods. These are: (i) Modular testing system: This system is based on the concept of abstraction. In this regard, the tester creates scripts for each module of the application being tested separately, and then combines these scripts in a hierarchical order to create large test cases. This creates an abstraction layer between modules, so any changes to the test scripts for one module do not affect any other modules. Advantages of this system: easier testing maintenance and scalability. Creating test cases using already scripted modules is and faster. Disadvantages: data embedded in test cases. Thus, to execute the same test script with different data, there are large changes to the script level. (ii) Data-based testing system: a data-based test system for input data and input data corresponding to input data is stored in a file or database, and the automated script uses the same set of test phases for multiple data sets. With this system, we can run several test cases where only the input data varies and the execution stages are the same. Advantages: Reduces the number of test scripts to run. We run the same script multiple times with different data. Less coding for automation testing. More flexibility to maintain and detect bugs or improve functionality. Test data may be created even before the automated test system is prepared. Disadvantages: Only similar checks with the same set of enforcement actions can be combined for multiple datases. A different test case is required for a different set of execution actions. (iii) Keyword-driven testing system: it is an application-independent testing system that uses data tables and self-arcing keywords. Keywords explain the steps to be taken for the test application, and the data table provides input and expected output data. Keyword-based testing is data-based testing. Advantages: Less encoding and the same script can be used for multiple datases. You don't need to know automation to create a test using existing activity keywords. One keyword can be used in several test cases. Disadvantages: This system is more complex because it has to take care of keyword actions as well as data entry. Test cases become longer and complex, thus affecting the same diet. (iv) Hybrid Testing System: This system is a combination of all of the above-mentioned testing systems (Modular, data-driven, and keyword-driven). In this context, test cases are developed from modular scripts, combining them into modular testing systems. Each test uses a driver script that uses a data file, as is the case with a data-based structure and a keyword-based action file. Advantages: Modular and easy to maintain. Less coding can take care of more test cases. One test instance can be executed with multiple datases. Disadvantages: Complex read, maintain and improve. Q #28) When Do You Want Manual Testing Over Automation Testing? Answer: We prefer manual testing over manual testing in the following cases: the project is short scripts and writing scripts will be time consuming and costly. Flexibility is needed. Automated test cases are programmed and run in a certain configuration. A usability test must be carried out. Applications/module is newly created and there are no cases of previous tests. A hoc or exploratory testing. Q #29) Ir Ir testing neat methodology useful isn't it? Answer: Automation testing is useful for regression, smoke or intelligence testing. All these types of testing in the traditional waterfall model take place at the end of the cycle, and sometimes, if there are not many application improvements, we probably won't even have to perform regression testing. Since using a flexible methodology, each iteration requires the execution of a regression test in the case of some new features being added. Also, the regression kit itself continues to grow after each sprint, because in functional test cases the current sprint module should be added to the regression suite for the next sprint. Thus, automation testing neat methodology is very useful and helps achieve maximum test coverage in less time sprint. Q #30) List some of the advantages and disadvantages of automation testing. Answer: Advantages: Less human resourcesReusabilityMore test coverage less timeReliabilityParallel execution test casesFastDisadvantages: Development and maintenance time is more. Costkilled resources are needed. Debugging of setupTest scripts is a problem. Q #31) List some of the advantages and disadvantages of manual testing. Answer:Advantages:No environment setup required. Programming knowledge is not required. Recommended for dynamically changing requirements. Allows human observation power to detect more bugs. For short-term projects, the costs are lower. Flexibilitycomplex: Difficult to perform complex calculations. ReuseTime, takingGaugsu risk of human error or error. More human resources are needed. Q #32) Can we do Automation testing without a system? If so, why do we need a system? Answer: Yes, we can perform automation testing without even using the system. We can only understand the tools that we use for automation and program steps in programming language that tools support. If we automate test cases without a system, then there will be no consistency in programming scripts for test cases. The system is necessary to provide a set of guidelines that everyone must follow in order to have retained the readability, reusable and consistency of test scripts. The system also provides a common basis for reporting and recording functionality. Q #33) How do you automate basic login functionality in test cases application? Answer: Assuming that the automation tool and system are already testing the environment in place. To check the basic Login functionality: Understand project requirements: Login functionality will be username text box, password text box, and login button. Identification of test scenarios: The possible test scenarios for logging on functionality are:Empty user name and passwordUnqualified user name and passwordA useful user name and invalid passwordValid user name and passwordPrepare a data entry file with data that matches each scenario. Run the tool from the program. Specify the user name field, passwords and the login button. For each test, the test extract data from the data file and enter it into the appropriate fields. The program clicks the logon button after you enter the data. Check the error message for negative scenarios and positive scenarios in the success report test script using assertions. Run the test kit and generate the report. Q #34) Is automation testing Black box testing or White-box testing? Answer: Automation testing is primarily black box testing, as we only program the steps that the manual tester performs the application under the test without knowing the low level design or code of the application. Sometimes automated test scripts require access to database data used in the application being tested or some additional encoding data, and thus they can be a way to test the white box. Thus, automated testing can be both black and white box-type testing, depending on the scenarios in which automation is performed. Q #35) How many test cases are you automated per day? Answer: Well, the number depends on the complexity of the test cases. When the complexity was limited, I was able to automate 5-6 test cases per day. Sometimes, I was able to automate only one test case in a complex scenario. I have also broken my test cases into different components such as make input, do calculations, check output, etc. when there are very complex scenarios and have taken 2 or more days. Q #36) What factors determine the effectiveness of automation testing? Answer: Some of the factors that determine the effectiveness of automation testing are:time saved when running scripts over manual execution of test cases. Defects foundTest Coverage or code overlaySuing time or development timeSpraition scriptTest ReusabilityQuality software testQ #37) What test scenarios can be automated? Answer: The types of test cases that can be automated are as follows:(i) smoke test cases: smoke testing is also known as construction test testing. Smoke test cases are run every time a new build is released to test build-making health to perform testing. (ii) Regression test cases: regression testing is testing to ensure that previously developed modules function as expected after the addition of a new module or correction of an error. Regression test cases are a very important incremental software approach when new functionality is added at each incremental stage. In this case, regression testing is performed in each incremental phase. (iii) Complex calculation test cases: this category includes test cases which include some complex calculations to test the application field. Complex calculation results are more prone to human error, so if automated they give accurate results. (iv) Data-based test cases: test cases with the same set of steps and operating more than once when changing data are referred to as data test cases. Automated testing in cases of this type of test is quick and cost-effective. (v) Non-functioning or functional test cases: test cases such as load load and performance tests require a simulated environment with multiple users and multiple hardware or software combinations. Setting multiple environments manually is not possible for each user combination or number. Automated tools can easily create this environment for easy non-functional testing. Q #38) What are the stages of the automation testing life cycle? Answer: The life cycle stages of automation testing are as follows: decision to perform automation testing. Identify and learn about the automation tool. Determine the scope of automation testing. Design and design a test suite. Test executionTest script maintenance. Q #39) What is an automated test script? Answer: An automated test script is a short program that is written in the programming language to perform a set of instructions for the application to be tested to verify that the application meets the requirements. This program, when run, gives test results as to go or does not depend on whether the application is in accordance with expectations. ConclusionSure These are key issues that do not depend on the automation tool or programming language. Automation testing interviews also include tool and programming language issues depending on the tool you've worked with. Most test automation interview questions are centered on your developed system, so it is advisable to carefully create and understand your test system. When I am interviewing, and the candidate has answered my question about the system, I would also like to ask a language question (the core of the java in my case). Questions start from the basics of java typing logic in some basic scenarios, such as: How do you extract a text set from a specific line? How do you extract the URL? Any website, any frame, number of links and its content change dynamically, how do you handle it? How do you handle images and flash objects? How do you find the word in the queue? The answers to all these test automation interview questions are very specific to the tool/language you use for automation. So before you go for an interview, brush your programming skills. In case you don't get a chance to set up your own system and someone else has created it, then take some time to understand it thoroughly before sitting for an interview. Some tips for automation testing interviews would be:Thoroughly know your tool. Learn the locator techniques used by your tool. Practice programming using the language you use to test automation. Learn about your system and its components. It is always beneficial if you have been involved in the development of your system. So, carefully with the modules in the system that you have worked on. I hope these questions would be helpful for you to prepare for a test automation interview. Interview.