**What is Automation**

Automation Testing uses an automation tool to execute test cases. The main goal of Automation Testing is to reduce the number of test cases to be run manually and not eliminate Manual Testing.

**Why Automated testing**

* Save time, effort, and money on repetitive tests.
* Avoid errors with manual rework
* Automate smoke and Regression test cases

**Automation Vs Manual**

| **Manual Testing** | **Automated Testing** |
| --- | --- |
| A human executes the test cases one by one, without any software assistance. | Tests are executed by a testing tool or framework, without human assistance. |
| Useful for non-repeatable tests that involve human ingenuity, participation, and domain experience. | Useful for repeatable tests where the software feature under test doesn’t change frequently. |
| Good for accessibility and usability testing, as the tester can test the software from an end-user’s perspective. | Good for testing regression issues to make sure that the software didn’t break after introducing new changes. |
| Can be slow and time-consuming, and subject to human errors and misjudgment. | Since it’s run by a computer, automated tests are fast and free from errors, given that we are testing the right thing. |
| It’s possible to test the software in a randomized manner, also known as exploratory testing. | Exploratory testing is not possible in automated testing. |
| UI problems and inconsistencies are easily spotted by a human tester. | Unless it’s programmed for that, the automated testing cannot discover and report the UI problems. |
| It’s very difficult, rather impossible to test the software under extreme load to conduct performance testing. | Performance testing can be easily done with automation testing. |
| The tester doesn’t need prior programming knowledge. | To write automated tests, the tester needs to have prior programming knowledge. |

**Practices to follow when writing automated tests?**

All the software development rules apply when writing automated tests. Here are some of the best practices that one can apply for tests.

* **Don't Repeat Yourself (DRY) (or) Go for reuablse functioons**
* Avoiding duplication in code is crucial.
* The benefit of this strategy is that the change is isolated to a single location, preventing bugs and errors.
* **Keep Functions Small – To manage and maintain**
* **Write Good Documentation – To understand the code**
* It can also help the person who wrote the tests when they try to modify/understand the tests in the future

**When to Automate**

* The Application should be stable and future changes are very minimal

**What to Automate**

* Tests which are required to run on regular basis.
* If it’s a one time or low priority test cases, we should not select for automation

**Points to be noted in Automation structure**

* Separate the test data from main code
* Separate the weblocators from the main code
* Configuration details (env,database cfredentials and urls) should be managed in config file

**Is 100 % Automation is possible ?**

No

Example: Captcha codes ,working on changing requirements

**Is 100% (exhaustive testing possible)**

No. In Real time , We can only check for important or priority test cases because of shortage or time and resources(people).

* We need to check with the requirements which are important for customer/client.
* Avoid any calculation errors where money is involved
* High priority or risk areas need to be tested first.

**Requirement mapping:**

Each requirement from customer has to be mapped to one or more test cases to ensure proper testing

**Defect Management Tool:**

Tool : JIRA

* Open
* Assigned to developer
* In progress
* Assigned to Tester (Retest)
* Fixed or resolved
* Closed

**Attributes for Good Automation Framework:**

* Reusable – Avoid duplicate lines of code
* Modular – Easy to modify and make changes
* Consistent – Format and coding standards are unique across the project
* **Integration – Easy to integrate with third party applications**

**Types of Environment:**

* Development
* Testing
* Production

What is Cross browser testing?

Cross-browser testing is a type of browser automation testing where the tester verifies if the web application will work smoothly on different browsers. Some of the popular browsers include Google Chrome, Mozilla Firefox, Internet Explorer, Safari, etc.

Reference for quick walk before interview:

<https://www.besanttechnologies.com/selenium-with-python-interview-questions-and-answers>