* What are some best practices you follow for maintaining your

automation suite? How do you ensure your tests are reliable and maintainable?

1.I try to choose correctly which tests have to be automated. I usually try to choose tests that have to be executed several times or they have to be repeated many times.

2.I try to have quality data or data that will be useful in many tests and data that would be so similar to the real one. Have a recreated scenario that is the most similar to the real one.

3.I think that having a test environment where execute the tests is so important. Even, have a clone of the production env with the features to be tested. If this is possible, you can have a better approximation to the prod scenario.

4.Be focused in creating test cases that won’t suffer changes or at least so small changes if there are changes inside the solution

\*To have test cases that are maintainable, the 5th point is so important. Even having good code programming practices to have documented the code with comments and explanations it would help to fix or make changes faster. In terms of reliability, I like to be peer reviewed by another dev or qa mate that can help giving his/her point of view and have test cases linked with the requirements to know exactly the expected result, so assure that the result is well marked as passed or fail correctly.

* Can you describe your approach to developing a test automation strategy? How do you balance between manual and automated

testing?

I think that it is so important to decide which tools fit better with the system under testing, so at first I think that it is so important to analyze the technical solution to decide which is the best tool in terms of automation. Define which scenarios have to be tested and then divide in manual or automated testing taking into account:

1.If the test has to be executed several times or just once to assure the correct execution

2.The knowledge and automation experience in the team. If the team has not so much experience, the effort in terms of knowing the technology and creating the automated test has to be taken into account,because some time has to be used in training.

3. Define good processes with dev teams in terms of integrate automated tests in the CI/CD process

* How do you ensure your automated tests provide sufficient coverage?

What strategies do you use to identify gaps in test coverage?

For finding gaps in test coverage I usually do:

1. Read carefully the requirements and be sure that I understand all and there aren’t ambiguous requirements. This is so important because you need to be sure about which is the result for each action and can design cases with this information

2 Have a or list to mark which requirements are tested in every test, so in this case you can ensure that all requirements are tested

3. As I mentioned before, peers reviews in terms of having another point of view I think that are so helpful

4. Be open to make changes while executing. I mean, that once you start the execution, you can realize that there are scenarios that cannot be reached or maybe you can discover new cases, so have the capacity to adapt your testing to add this scenarios or check with the team how to fix the problem

* Can you explain the Test Pyramid and its significance in test

automation?

It is a test strategy with three levels that helps the dev/qa team to realize if the code during all the cicle until be deployed, works well and if it doesn’t work well, know where it fails quickly.

This strategy consists of dividing the tests in three types layered as a pyramid, with unit tests at the bottom of the pyramid, integration tests in the middle and end-to-end at the top.

The idea is that the bottom of the pyramid should have the highest number of tests, and the top should have the lowest number of tests. That’s because the test execution time increases with each level of testing.