**SENIOR QA AUTOMATION**

1. *What QA methodology would you establish?*

**Agil Methodology**: Traditional software development methodologies work on the premise that software requirements remain constant throughout the project. But with an increase in complexity, the requirements undergo numerous changes and continuously evolve. In Agile methodology, software is developed in incremental, rapid cycles. As soon as iteration is completed, the entire system is subjected to testing. Feedback from testing is immediately available and is incorporated in the next cycle. The testing time required in successive iteration can be reduced based on the experience gained from past iterations.

Interactions amongst customers, developers and client are emphasized rather than processes and tools. The agile methodology focuses on responding to change rather than extensive planning.

1. *What testing types would you have your team perform?*

**Unit tests:** are very low level, close to the source of application. They consist in testing individual methods and functions of the classes, components or modules used by software.

**Functional tests:** Focus on the business requirements of application. They only verify the output of an action.

* **Integration Test:** Verify that different modules or services used by application work well together. For example, the interaction with the database or making sure that microservices work together as expected.
* **End to end:** Testing replicates a user behavior with the software in a complete application environment. It verifies that various user flows work as expected and can be as simple as loading a web page or logging in or much more complex scenarios verifying email notifications, online payments, etc.
* **Acceptance tests:** They require the entire application to be up and running and focus on replicating user behaviors. Reject changes if certain goals are not met.
* **Regression test:** It is done when the software or its environment have been modified.

**Performance tests:** Check the behaviors of the system when it is under significant load. These tests are non functional and can have the various form to understand the reliability, stability, and availability of the platform. For instance, it can be observing response times when executing a high number of requests, or seeing how the system behaves with a significant of data.

1. *Create a list of test cases according to the software specifications*

***Annex test cases in the "List Test Cases" folder.***

1. *Define the test plan for this project.*

***Annex test plan in the "Test plan" folder.***

1. *What tools/software would you recommend using for testing?*

**Test Management Tool**

* Xray
* Jira
* TFS
* Azure DevOps

**Defect Tracking Tools**

* Mantis Bug Tracker
* Trac
* Jira
* TFS

**Automation Testing Tools**

* Selenium
* Sahi
* RedwoodHQ
* Katalon
* Ranorex
* RoboFramework
* Serenity
* Cucumber
* Tricentis Tosca
* Sikuli
* Appium
* Robotium

**Performance Testing Tools**

* JMeter
* WAPT
* BlazeMeter

**Version Control System**

* Git
* SVN

**CI/CD**

* Jenkins

**Code Quality**

* SonarQube

1. *How would you address automation in this project taking into account the complexity of the platform and that there are 5 years worth of functionality that has not been automated?*

Initially, I would create a planning which consists of several stages: Context, Evaluation, Scope, Training / Investment, Execution.

**Context:** At this stage, I would discuss with all the stakeholders of the project the feasibility, resources, opinions and proposed time to carry out the automation in the project.

**Evaluation:** In the company of the team, create a matrix, in order to establish which are the flows of greatest importance, complexity, use and maturity, giving punctuation for each one.

To this matrix, add a weighting variable, which will be a basis to indicate which flows should start to be automated. It should be borne in mind that these flows must generate a return on investment.

**Scope:** Through a refinement of the evaluation stage, establish the goal to which we want to reach taking into account the established business rules (Time, Cost, etc.). Also capture the methodology to use, the tools.

**Training / Investment:** Having already defined the scope, verify that it is more beneficial for the company in economic terms and human capital, whether to hire new resources or training the personal existing one or both.

**Execution:** According to the methodology to be used and the priorities defined, initiate the automation of the flows, in a transversal way the progress or setbacks and contingency plan must be indicated.

1. *How would you transition the current QA team from performing functional testing to automation without sacrificing quality and speed?*

Automated tests do not replace manual tests, it is a complement:

* Test automation is used to run test scenarios that are performed manually, but quickly and iteratively.
* In addition to functional testing, other processes can also be automated in the application, such as load, performance, and stress.
* This increases test coverage, improves accuracy, and saves time and money compared to manual tests.

The choice of technology plays an important role in the transition from functional tests to automated tests, in my opinion a filter should be made to the equipment in order to verify that people have notions of development and of course automation.

If knowledge is basic, a good tactic is to start using record and playback tools. This could be implemented in the daily use of the personnel, who will be able to verify which is the real operation of said tools when performing the scripts reproduction. At this point there must be some technical guidelines that guide or give notions that it is an object, that it is adherence and terminology that can help the team evolve.

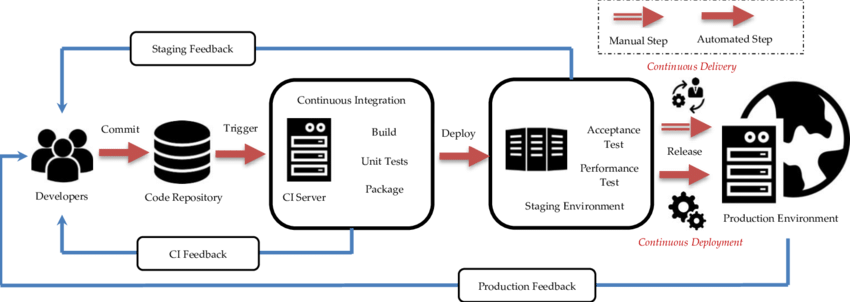
Another important point is that the team is already familiar with user stories, so involving gherkin language is a great help, since it basically describes the behavior of the business(multiple frameworks that use this technique, generate practicality and make it easy to start in the world of automated tests), in a cross sectional way, either in non work times already agreed in the team or of your own interest, you must evolve in the part of syntax, algorithm of the programming language with which the automation must be performed and this is where a merge should be made among all the guidelines mentioned above (Type of tool, language, frameworks).

It is clear that the team must have a person in charge of verifying learning and reinforcing the transition.

1. *How would you implement a Continuous Delivery flow in production ?*

When devising a software CI/CD strategy its best to keep the overall product, user, and business strategies in mind. Considerations will need to be made on what the most high value test coverage targets are. At the core of continuous deployment is a great CI culture. The quality of test suite will determine the level of risk for release, and the team will need to make automating testing a priority during development. This means implementing tests for every new feature, as well as adding tests for any regressions discovered after release.

So, I should have my environment of continuous integration, in which I can establish compilation, testing and reporting tasks. Later, if the results are correct, I apply continuous delivery, automating(Pipelines) the release of the verified code to a repository where the operations area can intervene or we can simply perform continuous deployments, fulfilling all the stages.



1. *Which processes, techniques and tools would you use to do QA automation for cross browser and cross device testing the most effective way possible?*

For processes, first really define what the target population is, that is, what types of devices must be compatible, like with browsers, this in order to narrow down time and scope.

For techniques, I would use test frameworks like TestNG which provides different annotations that would allow to give a good handle to automated tests, including parallelism, and BDD with which I can verify the real behavior between browsers.

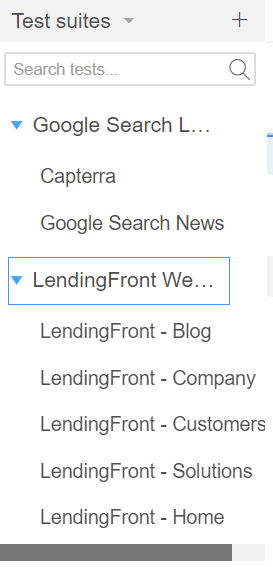
As automation tool, I quite like selenium in its webdriver version since it is not 100% javascript based. Remember that some browsers have compatibility problems with javascript versions.In addition, this version features adherence for automated testing of mobile devices.

Another very useful tool is sikuli, which is image-based automation. Its limitation is that it depends on the use of the screen, which must be dedicated, that is, there cannot be another process running.

1. *Create an automated test pointing to* [*www.google.com*](http://www.google.com)
   1. Record and play With Selenium IDE

***Annex script in the "Scripts" folder file name “LendingFront- Test.side”.***

***Content:***



Info: There area two test suites: Google Search and LendingFront WebSite, Google Search is configured to run your test cases in parallel with **selenium-side-runner** command line**.**

* 1. Using selenium and python create a test case with an assert that looks for a specificword in www.google.com page that doesn’t exist so the test would fail.

***Annex script in the "Scripts" folder file name “Test\_ Google.py”.***

Preconditions:

Install dependencies of selenium for python and chrome driver.