Web application quality assurance involves some functional and non-functional test types, such as unit testing, interface testing, security testing, performance testing, usability testing, and so on. Having a defined testing strategy contributes to software quality by reducing the risks of future problems after project implementation by identifying problems in advance.

For the ToDo APP[1] application (https://qa-test.avenuecode.com/), it is important to perform automated unit tests to identify bugs in the application units (for example, methods, classes). They can be done during the development of the application using JUnit (for example) and applying the Test Driven Development (TDD) in which the unit test is written before the code, this test is executed (which will fail, since it does not have code still), then the code is written just enough to satisfy the test; then the test will run, if it passes, it is necessary to refactor the code, otherwise the developer needs to review the code and run the test again.

If the ToDo APP application contains integration with another application, it is important to test the communication API between the two applications. For this, we can test the request and response (request and response) between systems using the SoapUi test tool (for example). With this tool it would be possible to validate in XML file the Web Service operations validating the requisitions and responses between the systems.

In addition to API testing, with SoapUi it is also possible to perform security tests, but also load and stress tests that are part of the performance tests. For the ToDo APP application, this type of test is very useful, since it is possible, for example, to validate the behavior of the system when several tasks and subtasks are registered by the user. Another tool available to measure the performance of applications is the JMeter with which we can simulate a load in the system and validate the behavior of the system. JMeter creates a test plan (manual or automatic) that will be used to perform the load test and at the end a report will be generated with the test result for analysis.

To validate user empathy with the application, the usability test will help measure how much the system is user-friendly. It is done with one or more end users and a moderator who will give instructions to the user. This test is evaluated with how many steps and how much time it takes to perform actions on the system, how easy the user remembers how to perform these actions, how many mistakes the user makes when performing an action and how the person feels when he or she complies the All these characteristics help to identify the impact that the application has on the end user.

Even though it is not an automated test, the exploratory test is an efficient option in identifying unmapped errors in the automated designed scenarios. This test has no set scenario, only a specific time to use an application functionality and identify the behavior of the system in various situations, which can generate errors and success situations. In situations that generate error, we validate how the system behaves and whether the return to the user of such an action is enough to understand how the system works. During the tests, all possible problems are scored and after completion of that set time, errors are reported and analyzed.

Other types of testing can also be used and to define this, a testing strategy is important to identify the critical points of the application, be it security or usability, and thereby define the tests that are most applicable to each functionality.

[1] Link: https://qa-test.avenuecode.com/