withdrive

Test strategy

Test strategy

1.1 Why do we test?

The goal of software testing is to find errors, gaps, or missing requirements in comparison to the actual requirements. We do this so that we can find and squash bugs before delivering a software product to the client. Testing is helpful when we incorporate new features, as we may run older tests that worked before, and ensure that the implementation of new features did not disrupt the application. This is also an important topic that is discussed in the DOT framework. As discussed later.

1.2 What should we test?

It is important to test the C4 part of the implementation. At the C4 level, the whole program is separated into units. These units (pieces of code) may be tested, and this action is called unit-testing. The point of unit testing is to ensure that each unit of the program is working correctly as per defined assertions.

After you make sufficient unit test and ensure that the code coverage of said tests encompasses all units of the application, then you must check if all the components within the scope of the application are working correctly and as intended. This is called integration testing, and as the name suggests we test whether all the components when put together are working correctly and as described.

Next, we must test how the entire system functions. Again, we zoom out and now we are testing the behaviour of all the elements working together. This is called system testing. We do this to evaluate the system's compliance with requirements. System testing takes, as its input, all the integrated components that gone through and passed the integration testing phase.

Finally, to ensure that the end-product meets all the requirements set out by our clients, we do acceptance testing. These tests involve testing the software based on the needs set out by the client/s. The goal is to test and successfully pass the requirements and criteria that was set out by the client.

1.3 How do we test?

This section will look at how the aforementioned testing phases are applied in the real world with a real project. This will include an outline on the tools that will be used for each of the test stages.

Unit testing – For unit testing of the withdrive project, I will be using Junit5 for the testing, as it is a popular framework for Java testing. To mock the database, I am using Mockito, this allows me to test the service layer efficiently. Unit tests for each 'atomic part' of the code (e.g. a method or function). The unit should be tested in isolation. (*Methods - ICT research methods,* 2021)

Integration testing – Once all unit tests are made and pass, I then can start integration testing all the components of the backend. There are frameworks available to do help do this, my choice being Selenium, it is one of the most popular integration testing frameworks for Java.

System testing – Once all unit tests and integration tests are made and pass, I then can start system testing the entire application. This is done by testing the test cases that can be found test-plan document. For this part I will be testing the application and checking whether the desired outcome is met or not. System testing evaluates the complete, integrated system against its requirements. (*Methods - ICT research methods*, 2021)

Acceptance testing – Once all of the aforementioned tests are made and pass, then I will be able to try the acceptance tests. Here we test the requirements set out by the client/s and see whether the application does everything that there was set out for it to do. This is done to ensure the end client is satisfied with the experience.

Once all of these test's pass, then you may consider your application thoroughly tested and ready for deployment to the client and subsequent end users.