Deliverable D05: Testing learning report

-

Group E8.02

Github Repository: https://github.com/acme-recipes/Acme-Recipes

Members:

- Gregorio Ortega Soldado (greortsol@alum.us.es) 30271286C
- Alejandro Manuel Gestoso Torres (alegestor@alum.us.es) 30260633Q
- Jaime Stockwell Mendoza (jaistomen@alum.us.es) 30696480J
- Pablo Aurelio Sánchez Valenzuela (pabsanval1@alum.us.es) 30246142S
- Manuel Cabra Morón (mancabmor1@alum.us.es) 47561328L
- Fernando Claros Barrero (ferclabar@alum.us.es) 77868841H

Fecha: 28/08/2022

Table of Contents

Executive Summary

The following document details an explanation of what we know about testing a Web Information System after taking the subject of Design and Testing II, from this degree of Software Engineering.

Revision Table

Revision	Description	Date
v1.0	Sprint retrospective discussion	28/08/2022
v1.1	Revision of our base of knowledge	28/08/2022
v1.2	Final version	29/08/2022

Introduction

It has already been taught to us in this course the importance of testing a Web Information System, not only because of making sure that our code is right, but for keeping it simple, eliminating defects or guaranteeing quality.

We have learned that we can appreciate 2 different ways of testing: a functional and performance way.

The functional tests detect failures and bad behaviors in the system by comparing the actual results of a request to the system with the value we expect the system to return.

On the other hand, we have performance testing, that checks if the syste performance accomplishes the expectations. This approach requires performing statistical-based investigation at a given confidence level.

We must take into account that despite not doing any actual testing or measurement on this particular project, we have already coursed the subject and done the testing tasks as part of the previous project. As a result, we have learned on the previous project some key topics that we will feature now.

Contents

Functional Testing

When testing a Web Information System we had to face different concepts:

- **SUT** (System under test). In the general theory of testing, the SUT ranges from a method in a class to a full system. Our application, Acme Toolkits was the SUT, and we tested it feature by feature carefully.

- **Fixture**. It consists of a web of artifacts that will be needed to set up for the SUT in order to be tested. The fixture ranges from an object to a complete system. Our database is the fixture.

When implementing the tests we had to make sure that we had fulfilled the 3 kinds of test that we were taught in classes:

1. **Positive test cases**. It is a script that checks that the SUT works well in a context in which it is expected to work well. Working well means that the SUT does not issue any errors or exceptions and produces the result expected.

   For instance, we implemented a positive test case from the feature any.chirp in our last project, in which any kind of user (authenticated or not), could create a chirp successfully. This kind of test has been the most habitual for us to implement in previous subjects.

2. **Negative test cases**. It is a script that checks that the SUT works well in a context in which it is expected to report an error. Furthermore, in order to implement a complete negative test, we have learned how to use invalid data (trying to submit it) and we have taken into account some pieces of advice that we will use in the future, such as, empty forms or binding errors. However, the most interesting ones for us were errors that involved patterns, ranges of dates and complex validations, given that we had never implemented a negative test that was similar to reality.

3. **Hacking test cases.** It can also be considered as a negative test case. In this context, it is a script that checks that the SUT works well in a context in which it is expected to crash. For example, the case that would happen when a user requests a feature with the wrong user role.

   Regarding these kinds of tests, for most of us, we had never paid much attention to them in previous subjects, so they were really interesting for us here.

Regarding the organization of the tests, we applied the techniques used on the Acme Jobs project. We created Test classes, which are collections of related test cases that are encapsulated in a single class. Our test classes encapsulated positive, negative and hacking test cases (if any) regarding each particular feature of the semester project.

Our approach to functional testing is called E2E (end-to-end). It attempts to test software “naturally” since the goal is to simulate a user interacting with the system, while checking that the system performs as expected in every step of the user simulated experience.

Performance Testing

We believe that the performance testing has been one of the highlights of the testing part, given that this was the first test of that kind that we all were making.

In our previous experience we first ran our test suites and collected performance data, thanks to the excel provided by the generated logs. Then, we computed the confidence interval for the mean of the wall time (we used the standard of 95%).

However, the goal was to prove that the performance could be improved, by means of a reforization of code (using the computer of a different teammate). Then, we repeated the collection of performance data with the new information.

Finally, we launched the Z test to analyze and compare both examples (before and after refactoring code). Thanks to the p-value, resulting from this analysis, we could conclude if the performance has been improved.

Conclusions

After having taken this subject we believe that we have acquired knowledge about testing a Web Information System, that we did not have before. Most of us had never implemented a negative or a hacking test case, and now we consider them a very useful approach.

We hope to have accomplished the expected goals about testing, however we seek learning more on this topic in the future. Therefore, we hope to become real experts in the field of testing and learn everything about it, not only the basic knowledge acquired in this subject, in our future.