17 DE FEBRERO DE 2024

TESTING OF WIS REPORT

GROUP C-1.047

Acme SF, Inc.

Repository link: https://github.com/Cargarmar18/Acme-SF-D01

Castillejo Vela, Manuel
Email: mancasvel@alum.us.es
García Martínez, Carlos
Email: cargarmar18@alum.us.es
Rodríguez Millán, María C.
Email: marrodmil@alum.us.es

Table of contents

1. Executive summary 1
2. Revision table 1
3. Introduction 2
4. Contents 2
4.1 Some key concepts 2
4.2 Structure of a test 2
4.3 Positive/Negative testing 2
4.4 Test doubles 3
4 Conclusions 3
5 Bibliography 3

1. Executive summary

This document presents a comprehensive report on our understanding of Web Information System (WIS) testing, derived from an analysis of key concepts. We have established clear definitions for fundamental terms such as Subject Under Test (SUT) and granularity. Additionally, we have outlined the phases of testing and explored testing methodologies, providing insights into alternatives for achieving better development practices.

2. Revision table

Revision number	Date	Description
1.0.0	17/02/2024	Instantiation and development of document

3. Introduction

Software testing is a crucial aspect of the software development lifecycle aimed at ensuring the quality, reliability, and functionality of software applications. It involves the systematic examination of software components and systems to identify defects, errors, or bugs that could potentially impact the user experience or the performance of the software.

Effective software testing not only helps in identifying and rectifying defects but also enhances user satisfaction, reduces maintenance costs, and minimizes the risks associated with software deployment.

4. Contents

4.1 Some key concepts

A test is defined by its granularity, which dictates the level of complexity and specificity of the Subject Under Test (SUT), representing each portion of code tested by the respective test. By specifying the granularity size, we can gauge both the complexity of the test and the extent to which our system is tested. Generally, greater granularity results in increased complexity and broader coverage. At its simplest form, the unit test is employed to expedite and streamline testing processes. The "unit" size varies and remains undefined, but typically encompasses each public interface within an object-oriented programming language.

4.2 Structure of a test

A test typically consists of three phases: arrange, act, and assert. In the arrange phase, the necessary data and conditions are set up for the test scenario. This involves defining the initial state of the system and preparing any input data required for the test.

The act phase involves executing the specific functionality or operation being tested, also known as the "subject under test" (SUT). This phase triggers the system to perform the intended action based on the arranged data and conditions.

Finally, in the assert phase, the outcome of the action performed in the act phase is evaluated against expected results or conditions. Assertions are used to verify that the actual outcome matches the anticipated behavior, ensuring that the test is successful. This phase validates whether the data arranged and processed during the act phase meets the predefined conditions or expectations, providing confidence in the correctness of the system's behavior under test conditions.

4.3 Positive/Negative testing

Regarding the kind of tests, we can divide the testing into 2 different types positive and negative. Positive testing is a software testing technique where the system is tested with valid and expected inputs. The main objective of positive testing is to verify that the software behaves as intended when provided with the correct input data.

On the other hand, negative testing is used to test negative scenarios, providing unexpected data to verify the behavior and the how the system responds and handle erroneous inputs.

4.4 Test doubles

Test doubles are used in software testing to replace components or dependencies of the system under test (SUT) with simplified versions that facilitate testing. These simplified versions mimic the behavior of the real components but are designed specifically for testing purposes. Test doubles help isolate the SUT from its dependencies, enabling more focused and reliable testing.

One common type of test double is the stub. They are used to simulate specific behaviors or return values that the SUT expects from its dependencies. Stubs are particularly useful when testing components that rely on external services or databases, allowing testers to control the input data and focus on testing specific scenarios.

Another type of test double is the mock. Mocks are used to simulate the behavior of real objects and verify that the SUT correctly interacts with its dependencies. Mocks are valuable for testing interactions between components and ensuring that the SUT behaves as expected in various scenarios.

4 Conclusions

In conclusion, the testing of a Web Information System (WIS) plays a pivotal role in ensuring its functionality, reliability, and usability align with user expectations. While our previous emphasis has primarily been on unit testing, which offers valuable insights into individual components, we recognize the need to broaden our testing approach. Moving forward, we are eager to incorporate end-to-end testing methodologies to comprehensively evaluate the WIS across various user scenarios and interactions.

5 Bibliography

- Bibliography: if there's no relevant bibliography, write "intentionally blank".