DP2 2021-2022

Knowledge of WIS' testing

Repositorio:

https://github.com/mpadillatabuenca/Acme-Toolkit.git

Miembros:

- José Manuel Bejarano Pozo (josbezpoz@alum.us.es)
- Mario Espinosa Rodríguez (maresprod5@alum.us.es)
- Andrea Meca Sánchez (andmecsan@alum.us.es)
- Manuel Padilla Tabuenca (maresprod5@alum.us.es)
- Ezequiel Pérez Sosa (ezepersos@alum.us.es)
- Javier Terroba Orozco (javteroro@alum.us.es)

GRUPO G1-E2-05

Versión 1.0.0

25/02/2022

Tabla de contenidos

Historial de versiones

Fecha	Versión	Descripción de los cambios
25/02/2022	1.0.0	Creación y finalización del documento.

Introducción

En este documento, realizado por Manuel Padilla Tabuenca hablaré de forma individual sobre los conocimientos previos a la asignatura de Diseño y Pruebas II en cuanto a la realización de pruebas de sistemas de información.

La estructura de los contenidos será de la forma:

- **Motivo de las pruebas y definiciones:** Motivaciones detrás de la utilización de pruebas como validación y términos recurrentes.
- **Trabajo con el framework JUnit:** Desarrollo de pruebas con el framework JUNIT en varios proyectos a lo largo de mi carrera.
- **Estrategias de cobertura y buenas prácticas:** Cómo generar una suite de pruebas rica, además de buenas prácticas para conseguirla.

Motivo de las pruebas y vocabulario

En cuanto a porqué se realizan pruebas en el desarrollo software, y las medidas que se pueden tomar para realizarlos de una forma eficiente, ordenada y rigurosa, puedo hablar de: alcance de las pruebas, tipos de pruebas, y términos utilizados.

Alcance de las pruebas

Se puede definir una prueba software como un fragmento de código que configura, ejecuta y comprueba la correcta, o incorrecta ejecución de una o varias partes de una aplicación software.
En cuanto al alcance de estas pruebas, existen varios tipos, siendo cada uno de ellos menos costoso que el siguiente, pero a su vez más focalizado, destacan los tests unitarios, que son en los que se ha centrado mi conocimiento.
Estos tests son pruebas, relativamente pequeñas, encargadas de probar, convenientemente una función de nuestra aplicación.

Tipos de pruebas

En cuanto a teoría sobre tipos de pruebas, en otras asignaturas han sido nombrados las distintas opciones, pero principalmente se han focalizado en tests unitarios.
Sin embargo, conozco la existencia de: tests de integración, tests end-to-end, tests de aceptación y tests exploratorios.

Términos utilizados

Como una forma de pequeña enumeración de principales términos a lo largo del testing de sistemas de información, tenemos:

- **Bug:** Ejecución errónea de forma inesperada en nuestro código, encontrar bugs es el principal objetivo de las pruebas.
- **Fallo:** Error en el código que produce estos bugs, puede no manifestarse o se considera como error humano.
- **Debugging:** Búsqueda exhaustiva por parte de desarrolladores de estos fallos.
- **Fixture/arrange:** Fases en las que se divide una prueba unitaria, son, el establecimiento de los datos, ejecución y comprobación.
- **Test case:** Método test que ejecuta trozos de código de nuestra aplicación.
- **Test suite:** Conjunto de pruebas de nuestra aplicación.

Trabajo con el framework JUnit

En cuanto al framework JUnit diría que estoy bastante familiarizado, ya que he realizado 2 proyectos utilizándolo, y considero que los conocimientos obtenidos en Diseño y Pruebas 1 son considerables.
He aprendido principalmente a realizar pruebas unitarias con él, parametrizar estas pruebas, hacer debugging sobre ellas, y probar capas web de sistemas de información.
Esto último, consiste en aislar, en el caso de una aplicación utilizando MVC, los controladores, de forma que los repositorios y servicios siempre devuelven respuestas esperadas, para centrarnos en el funcionamiento de la capa web.
Para realizarlo aprendí conocimiento sobre stubs y mocks.

Estrategias de cobertura y buenas prácticas

Diría que las estrategias de cobertura es el conjunto de conceptos del que me gustaría obtener más conocimiento, ya que en Diseño y Pruebas 1, lo que aprendí fue a probar todas las ramas de ejecución del código.
En cuanto a buenas prácticas tenemos: parametrizar los tests, mantener los principios DRY y KISS y centrar los tests unitarios en trozos concretos de código.

Conclusión

Las conclusiones obtenidas gracias a la realización de de este documento son principalmente, el pequeño resumen de las destrezas adquiridas a lo largo de mi carrera, pienso que en cuanto a las pruebas de sistemas web poseo un buen nivel de conocimiento, sin embargo, me gustaría ahondar en otros temas, como los mencionados anteriormente (estrategias de cobertura y buenas prácticas).

Además, me gustaría obtener más información sobre los distintos tipos de pruebas, ya que solo conozco las unitarias.

Espero que asignaturas de este segundo cuatrimestre como Diseño y Pruebas 2, o Proceso y Gestión Software 2 me ayuden a adquirir estos conocimientos nombrados.

Bibliografía

1. Apuntes de la asignatura de Diseño y Pruebas 1 (ev.us.es)