



Testing WIS Knowledge report

C1.034

<https://github.com/juanlo017/c1.034>

Apellidos, Nombre	Email
Blanco Mora, David	davblamor@alum.us.es
Castro Vázquez, Jose Ignacio	joscasvaz@alum.us.es
Junco Obregón, Juan del	juajunobr@alum.us.es
López Quirós, Juan	jualopqui1@alum.us.es
Mejías Buitrago, Pablo	pabmejbui@alum.us.es

Tabla de contenidos

Resumen ejecutivo	3
Tabla de revisiones	4
Introducción	5
Contenidos	6
Conclusiones	7
Bibliografía	8

Resumen ejecutivo

Este documento versa sobre nuestros conocimientos de las pruebas de un WIS previos a la asignatura Diseño y Pruebas II.

Tabla de revisiones

Versión	Descripción	Fecha
v1.0	Creación del documento, recapitulación de información de la arquitectura WIS.	14/02/2024
v1.1	Primera versión del documento.	15/02/2024

Introducción

En este documento se revisarán los conceptos sobre las pruebas de un WIS adquiridos a lo largo de nuestra trayectoria en el mundo de la ingeniería del software. Se tratarán diversos temas como los tipos de pruebas que se pueden realizar sobre un sistema, los distintos tipos de escenario para cada prueba y las características que ha de seguir una prueba.

Contenidos

Con los conocimientos adquiridos en las asignaturas anteriormente cursadas, disponemos de las bases que rodean a las pruebas de un WIS. En la asignatura Diseño y Pruebas I, estudiamos los fundamentos relativos al testing de sistemas de información: conceptos principales, tipos de testing, frameworks utilizados, buenas prácticas...

El testing en sistemas de información es una actividad crucial que tiene como objetivo garantizar que el software desarrollado cumpla con los requisitos establecidos, funcione correctamente y sea confiable. A continuación, se presentan los conceptos principales del testing en sistemas de información:

1. Tipos de testing: existen varios tipos de testing, como pruebas unitarias, pruebas de integración, pruebas de sistema, pruebas de aceptación, pruebas de regresión, entre otras. Cada tipo de prueba tiene un propósito específico y se realiza en diferentes momentos del ciclo de vida del software.
2. Estrategias de testing: la estrategia de testing define cómo se llevarán a cabo las pruebas y cuál será el enfoque para detectar los defectos en el software. Las estrategias pueden incluir técnicas como el testing funcional, el testing de rendimiento, el testing de seguridad, entre otros.
3. Plan de pruebas: un plan de pruebas es un documento que describe las actividades de testing que se llevarán a cabo, los recursos necesarios y el cronograma de las pruebas. El plan de pruebas es esencial para asegurar que se realicen las pruebas adecuadas y se cumpla con los plazos establecidos.
4. Casos de prueba: los casos de prueba son una descripción detallada de los escenarios de prueba que se utilizarán para probar el software. Los casos de prueba deben ser claros, precisos y completos, para garantizar que se prueben todas las funcionalidades del software y se detecten los posibles errores.
5. Reporte de errores: los errores detectados durante las pruebas deben ser reportados de manera clara y precisa, indicando el tipo de error, su gravedad y cómo reproducirlo. Esto permitirá que el equipo de desarrollo corrija los errores y realice las pruebas necesarias para asegurarse de que el problema haya sido solucionado.

En resumen, el testing en sistemas de información es una actividad crítica para garantizar la calidad del software y su correcto funcionamiento. Los tipos de pruebas, estrategias de testing, plan de pruebas, casos de prueba y reporte de errores son algunos de los conceptos fundamentales que deben considerarse en la realización de pruebas de software efectivas.

Conclusiones

En este documento hemos repasado los conceptos aprendidos en la carrera sobre el testing en sistemas de información. Esperamos poder afianzar en esta asignatura lo aprendido y obtener un conocimiento más sólido sobre este tema tan importante en el contexto de la ingeniería del software.

Bibliografía

Intencionalmente en blanco.