

Testing WIS Knowledge

Group: C1.031

Repository: <https://github.com/DP2-C1-031/Acme-One-C1.031>

Student #1

Name: Cuenca Pérez, Pablo
Email: pabcueper@alum.us.es

Student #2

Name: Núñez Sánchez, Juan
Email: juanunsan2@alum.us.es

Student #3

Name: Peña Núñez, Felipe
Email: felpennun@alum.us.es

Student #4

Name: Muñoz Romero, Javier
Email: javmunrom@us.es

Student #5

Name: Paradas Borrego, Álvaro
Email: alvparbor@alum.us.es

Date: Seville February 10, 2024

Contenido

Resumen ejecutivo	3
Registro de cambios	4
Introducción	5
Contenido	6
Conclusión	7
Bibliografía	8

Resumen ejecutivo

A lo largo de estos años anteriores en los que hemos estudiado en la ETSII se nos ha repetido flagrantemente la importancia de la realización de pruebas al código que realizamos en los proyectos orientados hacia los sistemas de información pues en ellos se va a basar en gran medida el éxito o fracaso del proyecto en sí mismo.

La detección de errores y su corrección es clave y para ello realizamos una serie de pruebas distintas, entre las cuales destacaremos las siguientes:

Pruebas de regresión, pruebas unitarias, pruebas de rendimiento, pruebas de integración y pruebas de aceptación.

Registro de cambios

Versión	Causa del Cambio	Responsable del Cambio	Fecha del Cambio
1.0	Primera versión	Pablo Cuenca Pérez	10/02/2024

Introducción

El testing de software permite tener procesos, métodos de trabajo y herramienta para identificar defectos en el software buscando alcanzar un punto de estabilidad del mismo.

Durante la realización del testing en un proyecto que trata de realizar un sistema de información podemos encontrar una amplia variedad en los tipos de testing, entre los cuales podemos destacar 5.

Las pruebas de regresión, las pruebas unitarias, las pruebas de integración, las pruebas de rendimiento y las pruebas de aceptación.

Cada una de estas pruebas se aplica de distinta manera para buscar debilidades o posibles mejoras a lo largo de todo el software que estamos implementando.

Contenido

Las pruebas son una de las partes más importantes que debemos tener en cuenta a la hora de realizar y planificar un sistema de información. El uso de estas pruebas está basado en la necesidad de introducir un control de calidad a la hora de desarrollar un software y, aunque las pruebas cuestan dinero, permiten un gran ahorro de dinero a las empresas que las realizan.

Este dinero se debe a la detección de fallos previos al lanzamiento del producto al mercado para que se pueda corregir, por ejemplo, puede detectar tanto fallos arquitectónicos, malas decisiones de diseño o vulnerabilidades del sistema.

Cuando se deja un espacio amplio para las pruebas el software se vuelve mas confiable y su calidad aumenta. Un sistema de estas características es más propenso a superar las expectativas del cliente y obtener un mejor rendimiento cuando sale al mercado.

Las pruebas en un sistema software son tan comunes en todos los sistemas como variadas.

Entre todos los tipos de pruebas destacaremos las siguientes:

1. Pruebas de regresión: verifica si alguna de las nuevas características añadidas al sistema rompe o degrada la funcionalidad del mismos. Es un paso crucial para permitir que los desarrolladores mejoren el software sin afectar negativamente su funcionalidad existente.
2. Pruebas unitarias: son las pruebas mínimas que se pueden realizar en un sistema. Valida que cada unidad de software funcione según lo esperado. Consisten en aislar una parte del código y comprobar que funciona a la perfección y permiten detectar errores que de otra manera sería imposible detectar.
3. Pruebas de integración: busca asegurar que todos los elementos del software operen juntos de manera correcta. Son aplicadas tras las pruebas unitarias pues ponen a prueba las interfaces entre los subsistemas.
4. Pruebas de rendimiento: comprueba el funcionamiento del software bajo distintas cargas de trabajo evaluando las capacidades del sistema con cargas reales de trabajo.
5. Pruebas de aceptación: verifica si todo el sistema en su conjunto funciona según está previsto. En el desarrollo ágil son pruebas recurrentes para asegurar que el cliente está de acuerdo con el rumbo que tiene el desarrollo del software.

Conclusión

Como conclusión debemos recalcar de nuevo la importancia de las pruebas pues de estas depende en gran medida el éxito o fracaso del producto final puesto que la detección de los errores y posibles mejoras recae completamente en estas. Saber de la importancia de la realización de estas pruebas y ser capaz de aplicarlas correctamente durante el desarrollo es uno de los puntos mas importantes a la hora del desarrollo y planificación de un proceso software.

Bibliografía

<https://www.ibm.com/es-es/topics/software-testing>

<https://www.verity.cl/5-pruebas-funcionales-que-todo-software-necesita/>

<https://geekflare.com/es/regression-testing-tools/>

<https://www.yeeply.com/blog/que-son-pruebas-unitarias/>

<https://keepcoding.io/blog/que-son-las-pruebas-de-integracion/>

<https://keepcoding.io/blog/que-son-las-pruebas-de-rendimiento/>

<https://pacifitic.org/que-es-el-testing-de-software-y-por-que-es-tan-importante-en-el-desarrollo-de-software/#:~:text=El%20Testing%20de%20Software%20es,proceso%20de%20estabilidad%20del%20mismo.>

|