

PROGRAMA DE ESTUDIOS

DESARROLLO DE SISTEMAS DE INFORMACIÓN

**ANÁLISIS Y DISEÑO
DE SISTEMAS**

Tema:

**VALIDACIÓN -
PRUEBAS
FUNCIONALES DEL DISEÑO.**

**DEL DISEÑO -
UNITARIAS Y**

VALIDACIÓN DEL DISEÑO - PRUEBAS UNITARIAS Y FUNCIONALES DEL DISEÑO.

Pruebas funcionales versus pruebas unitarias: comparación con ejemplos

Pruebas funcionales versus pruebas unitarias – Comparación con ejemplos

-La prueba es un proceso o sistema para identificar características y problemas. Se considera la columna vertebral del desarrollo de software. Las pruebas son cruciales, ya que ayudan a identificar problemas y defectos en un producto para que puedan solucionarse antes de entregarlo.

Esto también garantiza que el código se comporte como se espera y cumpla con los requisitos, garantizando en última instancia la calidad del software. Sin embargo, con tantas opciones disponibles, puede resultar difícil determinar qué pruebas utilizar, ya sea una prueba unitaria o una prueba funcional.

Debe leer el blog completo para decidir entre pruebas funcionales y pruebas unitarias. Esto le ayudará a comprender mejor los escenarios específicos en los que cada tipo de prueba es relevante y complementa al otro.

En este blog, encontrará una guía sencilla que le ayudará a comprender los diferentes aspectos de las pruebas unitarias frente a las pruebas funcionales. Entonces, eche un vistazo a la siguiente guía sencilla y rápida.

¿Qué son las pruebas funcionales?

Las pruebas funcionales son un procedimiento de prueba de sistemas de software para confirmar si el software coincide con su diseño. Se centra en todo el sistema y determina la eficacia con la que ejecuta las funciones previstas, confirmando si cumple con las expectativas del usuario.

Servicios de pruebas funcionales. tiene como objetivo validar las características, capacidades e interacción del sistema con diferentes componentes. Le permite verificar con precisión la entrada y salida, la manipulación de datos y la interacción del usuario, permitiendo probar todas las funciones.

Al ejecutar una prueba funcional, puede detectar errores o fallas que los desarrolladores pueden pasar por alto mientras crean la aplicación. Por lo tanto, las pruebas funcionales son información crucial para los controles de calidad del software y previenen posibles fallas en el cumplimiento de los requisitos comerciales y de los usuarios.

¿Qué es la prueba unitaria?

Las pruebas unitarias verifican código pequeño para entregar información temprana y frecuentemente y se enfocan en unidades y componentes individuales. Su objetivo es confirmar que cada unidad de software funciona según lo previsto y cumple con los requisitos.

Los desarrolladores suelen realizar pruebas unitarias al principio del proceso de desarrollo antes de integrar y probar el código como un sistema completo. Esta práctica es muy apreciada y adoptada por muchas organizaciones, y es beneficiosa para eliminar errores en el código.

Las pruebas unitarias tienen como objetivo revisar todas las unidades de código y garantizar que estén libres de errores y funcionen correctamente. Esto permite el crecimiento sustentable del proyecto y su ejecución para asegurar el comportamiento y resultados esperados.

Diferencia entre prueba funcional y prueba unitaria

Las pruebas funcionales versus unitarias tienen como objetivo identificar y prevenir defectos y entregar un producto de calidad. Sin embargo, algunas cualidades los distinguen. A continuación, se muestra la tabla para identificar la diferencia entre pruebas unitarias y pruebas funcionales.

factores	Examen de la unidad	Prueba de función
	Prueba unidades/componentes individuales de código.	Prueba toda la funcionalidad del sistema.

factores	Examen de la unidad	Prueba de función
Focus	Verificar la corrección y confiabilidad del código	Valida las relaciones con los usuarios y la funcionalidad de un extremo a otro.
Nivel de prueba	Realizado a nivel de código.	Realizado a nivel de sistema o aplicación.
Granularidad de la prueba	bien granulado	De grano rugoso
Probadores	Realizado por desarrolladores	Realizado por un probador dedicado
Dependencias	Dependencias simuladas o stubs	Dependencias reales
Aislamiento de prueba	Función o métodos de prueba.	Pruebas escenario completo
Cobertura de prueba	Cubrir una pequeña porción del sistema.	Cubre una gran parte del sistema.
Tiempo de ejecución	Rápido	Lento y bastante complejo
Propósito	Corrección del código	Validar el comportamiento del sistema y la experiencia del usuario.
Detección de errores	Detectar problemas tempranos con el código	Reconoce problemas funcionales con el sistema.

Factores clave para diferenciar las pruebas unitarias de las pruebas funcionales

Las pruebas tienen como objetivo garantizar la entrega de productos de alta calidad, eficientes y confiables. Es un enfoque sistemático y disciplinado que ayuda a identificar y solucionar defectos, errores y problemas durante el ciclo de desarrollo. A continuación se detallan los propósitos de las pruebas funcionales frente a las unitarias.

1. Propósito

Propósito de la prueba unitaria

Las pruebas unitarias son un paso importante en el proceso de desarrollo porque su objetivo principal es aislar el código escrito y probar si funciona según lo planeado. Es fácil de crear y ejecutar y requiere poco costo.

Las pruebas unitarias tienen como objetivo aislar las partes comprobables más pequeñas de una API (interfaz de programación de aplicaciones) y verificar que funcionen correctamente. Desde la perspectiva del desarrollador, las pruebas unitarias consisten en crear una base de código sólida con un costo mínimo.

Otro factor vital es proporcionar documentación para pruebas de alto nivel. Las pruebas de alto nivel pueden confirmar diferentes características de comportamiento del sistema bajo prueba. Sin embargo, verifica principalmente que el SUT (Sistema bajo prueba) produzca resultados correctos.

Propósito de las pruebas funcionales

Las pruebas funcionales ayudan a probar la funcionalidad de todo el software. Se realiza para identificar si cada característica de la aplicación funciona de acuerdo con los requisitos del software. El objetivo de las pruebas es comprobar las funcionalidades del sistema.

Comprueba toda la aplicación, el hardware y la infraestructura de red, desde la interfaz de usuario frontal hasta la base de datos de back-end. Las pruebas funcionales provienen de pruebas de integración, que confirman que las diferentes partes funcionan juntas como se predijo.

A diferencia de las pruebas unitarias, las pruebas funcionales no indican qué está roto y dónde ubicar la falla en el código base. Mientras que los mensajes de texto funcionales solo informan que es necesario arreglar algo.

2. Mejorar la calidad del software

Pruebas unitarias

Los desarrolladores comprenden a fondo el trabajo y luego escriben una prueba para ejecutar el nuevo código. Las pruebas unitarias ofrecen una red de pruebas de seguridad, lo que brinda a los desarrolladores la confianza para refactorizar el código.

Esto hace que sea más fácil y seguro refactorizar el código mediante la realización de pruebas que garanticen que la refactorización se produzca sin problemas ni interrupciones. Elimina los riesgos de cambiar el código fuente anterior.

Las pruebas unitarias obligan a los desarrolladores a escribir código modular, mantenible y fácilmente manejable. También proporciona grandes beneficios, como una mejor calidad del código y una mejor colaboración y productividad en equipo.

Por ejemplo, si un desarrollador descifra el código en 6 meses, puede ejecutar la prueba para proteger el código. Así es como las pruebas unitarias ayudan a prevenir la regresión. Por lo tanto, las pruebas unitarias ayudan a identificar posibles defectos en su sistema a través de un informe de errores, por lo que los desarrolladores no necesitan invertir tiempo y recursos adicionales a largo plazo.

Pruebas funcionales

Las pruebas funcionales garantizan que todo el sistema funcione según lo previsto. Cada función se compara con los requisitos correspondientes, lo que garantiza que las capacidades críticas de la aplicación se comporten como se espera.

Este tipo de prueba incluye varios subconjuntos, cada uno de los cuales ofrece beneficios específicos. Las pruebas del sistema validan que toda la aplicación o software esté en el contexto de un usuario real. Sin embargo, las pruebas de un

extremo a otro aumentan la cobertura y reducen el riesgo de integrar nuevos códigos en una aplicación.

Por ejemplo, las pruebas de integración garantizan el buen funcionamiento de todas las integraciones. Otro tipo de prueba de función es la prueba de humo, que le ayuda a comprobar las funciones esenciales del software, minimizar la regresión y ahorrar tiempo.

3. Cuándo realizar

Pruebas unitarias

Aunque las pruebas unitarias reemplazan las pruebas funcionales, son la base del proceso de prueba sobre la que se debe construir el resto del proceso de prueba. Los desarrolladores generalmente ejecutan pruebas unitarias durante la etapa de desarrollo.

Test Drive Development (TDD) es un método de desarrollo de software frecuente que admite pruebas antes del código. Con el tiempo, las ventajas de escribir pruebas unitarias se han hecho evidentes.

Pruebas funcionales

Las pruebas funcionales se realizan cuando dos módulos interactúan entre sí. Después de esto, los evaluadores prueban las funciones. Se realiza en la fase inicial del proyecto cuando los requisitos del usuario son más recientes.

Las pruebas funcionales están en otro nivel. Este tipo de prueba comprende varios subconjuntos, cada uno de los cuales ofrece beneficios específicos. Garantiza que todas las funcionalidades cruciales del usuario, como registros, inicios de sesión y flujos de trabajo de compras, funcionen como se desea.

4. Tipo de técnica de prueba

Pruebas unitarias

Según el propósito y la definición de prueba unitaria, es una técnica de prueba de caja blanca. La perspectiva interna del sistema y las habilidades de programación se utilizan para diseñar casos de prueba. El criterio para ello es realizar una sensibilización de rutas y estructuras de datos.

Las pruebas unitarias son una técnica de control de calidad en la que los códigos de aplicación se dividen en componentes básicos. Está diseñado para probar unidades individuales de código de forma aislada. Las pruebas unitarias son fáciles y rápidas de crear y ayudan a encontrar y corregir errores en el ciclo de desarrollo inicial.

Pruebas funcionales

Las pruebas funcionales se consideran una técnica de prueba de caja negra. Esta técnica prueba la funcionalidad del software sin notar ninguna estructura de código interno. Las pruebas funcionales se realizan según los requisitos de los usuarios o del negocio, comparando los resultados con los resultados esperados.

5. Propósito de la cobertura de la prueba

Pruebas unitarias

La cobertura de prueba en las pruebas unitarias tiene como objetivo detectar la mayoría de los errores antes de que lleguen a producción. Refleja todas las partes necesarias del proceso de desarrollo de software. La herramienta de cobertura de prueba realiza un seguimiento del código que se ejecuta y del que no.

La alta cobertura de código brinda a los desarrolladores la confianza de que todo el proyecto está bien desarrollado y mantenido. Garantiza que el conjunto de pruebas sea completo y que se hayan cubierto todas las funciones críticas de una aplicación.

Pruebas funcionales

Las pruebas funcionales pueden establecer la trazabilidad entre los requisitos y los casos de prueba. La cobertura de las pruebas es el elemento clave para evaluar la eficacia de las pruebas funcionales.

Garantiza que se prueben todas las funciones necesarias, utilizando varios parámetros de entrada para probar la lógica. Las pruebas funcionales dan una indicación de qué características se implementan y cómo satisfacen los criterios de aceptación.

6. Complejidad al escribir un caso de prueba

Pruebas unitarias

En TDD (Test Drive Development), se ha visto la complejidad de las pruebas unitarias. En TDD, los desarrolladores deben planificar el diseño del código en función de los requisitos de su software. Por lo tanto, si el diseño de su código inicialmente no está claro, evolucionará y le pedirá que repita la misma prueba.

Por ejemplo, si trabaja con algoritmos que cambian con frecuencia, sería mejor retrasar las pruebas unitarias hasta que comprenda mejor el diseño del código.

Pruebas funcionales

En las pruebas funcionales, puedes simular todas las dependencias en diferentes historias. Generalmente, los desarrolladores dividen su tiempo en partes: desarrollando nuevas funciones y solucionando defectos.

Una parte sostenible del lanzamiento se gasta en probar la funcionalidad ya lanzada y utilizada. El desarrollo de cada característica nueva necesita un plan de pruebas y esfuerzos de mantenimiento de las pruebas existentes.

Por ejemplo, probar una pantalla que procesa datos de la red celular no funcionará sin una conexión a Internet, lo que hace que la ejecución de una prueba funcional sea muy desafiante en tales casos.

Conclusión

Espero que esta guía le haya ayudado a comprender las pruebas funcionales frente a las pruebas unitarias. Los desarrolladores y evaluadores deben distinguir entre la intención y el alcance de las pruebas funcionales y unitarias. Ambas pruebas se utilizan para diferentes propósitos y tienen sus limitaciones. Entonces, para familiarizarse con las pruebas funcionales frente a las pruebas unitarias, lea el blog completo.

FUENTE:

- <https://www.uml.org/>
- <https://www.codingdojo.la/2023/06/16/guia-del-ciclo-de-vida-del-desarrollo-de-software/>
- <https://aws.amazon.com/es/what-is/sdlc/>
- BURCH, John; GRUDNISKY, Gary. "Diseño de Sistemas de Información", Grupo Noriega editores.
- SENN, James A. "Análisis y diseño de sistemas de información", 2da. ed., McGraw-Hill.

