

Parte 3

1. Ejercicio1

Las pruebas de integración se realizan para validar la interacción entre diferentes módulos, y se realizan pruebas unitarias para comprobar si son correctas las piezas de código individuales.

Las pruebas de integración del sistema se realizan principalmente para probar los requisitos de interfaz de un sistema. Mientras que las pruebas de aceptación del usuario se realizan para verificar la funcionalidad del sistema en su conjunto por parte de un usuario final.

2. Ejercicio2

De acuerdo a lo leído sobre la pirámide de cohn se puede decir que:

inicialmente realizamos pruebas unitarias para comprobar que los elementos como código, funciones estén funcionando de la manera adecuada y como se espera, al comprobar que todo está funcionando como se espera pasamos a realizar las pruebas de Aceptación/Integración/Componente para ver cómo nos va en nuestras pruebas unitarias al integrarlas entres si pasamos a realizar las pruebas de GUI, UI, o interfaz de usuario buscan validar el comportamiento de la interfaz. Es decir, las pruebas de GUI se orientan a detectar errores propios de la interfaz gráfica tales como problemas en el layout, lógica de presentación, estilos, responsividad, recursos, ortografía, etc. finalmente pasamos a Las estrategias pruebas exploratorias las cuales son una de pruebas de software que suelen describirse como aprendizaje simultáneo, diseño de prueba y ejecución. Se centran en la detección y dependen de que el tester descubra defectos que no es fácil cubrir con otras pruebas.

3. Ejercicio 3

- Prueba de caja negra
 - Los casos de prueba tendrán más detalles sobre las condiciones de entrada, los pasos de prueba, los resultados esperados y los datos de prueba.
 - El principal objetivo de esta prueba es probar la funcionalidad / comportamiento de la aplicación
 - Este tipo de pruebas puede ser realizado por un probador sin ningún conocimiento de codificación de la Aplicación bajo prueba.
- Prueba de caja blanca
 - Los casos de prueba serán simples con los detalles de los conceptos técnicos como declaraciones, cobertura de código.
 - El principal objetivo es probar la infraestructura de la aplicación.
 - El evaluador debe tener conocimiento de la estructura interna y cómo funciona.

4. Ejercicio 4

¿Qué son las pruebas exploratorias?

Es un proceso de aprendizaje, diseño de pruebas y ejecución de pruebas simultáneos. Podemos decir que, en esta prueba, la planificación, el análisis, el diseño y la ejecución de las pruebas se realizan de forma conjunta e instantánea.

Las pruebas automatizadas tienen como objetivo detectar fallas en el software evitando que una persona tenga que ejecutar las pruebas manualmente. En este caso, el experto en pruebas genera un caso a probar, utilizando una herramienta para construir un robot de software que realizará automáticamente las pruebas, así que pueden realizarse 24 horas, los 7 días de la semana, multiplicando la cantidad y el alcance de las pruebas. No requiere la intervención del individuo en cada nueva ejecución, ya que la prueba simula la interacción humana con el software.

es útil utilizarla cuando:

- Hay procesos repetitivos
- Hay flujos extensos que requieren actualizaciones y es necesario probar en regresión
- Hay casos complejos que consumen mucho tiempo
- Es necesario correr múltiples pruebas
- Es fundamental minimizar los errores humanos
- Se requiere probar componentes críticos
- Se precisa realizar pruebas en múltiples plataformas

5. Ejercicio 5 -> Técnicas de diseño de pruebas

- Análisis de valor límite (BVA)

El análisis del valor límite se basa en la prueba en los límites entre las particiones.

Incluye límites máximos, mínimos, internos o externos, valores típicos y valores de error.

- Partición de clases de equivalencia

El particionamiento de clase equivalente le permite dividir un conjunto de condiciones de prueba en una partición que debe considerarse igual.

Este método de prueba de software divide el dominio de entrada de un programa en clases de datos a partir de los cuales se deben diseñar los casos de prueba.

- Pruebas basadas en tablas de decisión

Una tabla de decisiones también se conoce como tabla Causa-Efecto.

Esta técnica de prueba de software se utiliza para funciones que responden a una combinación de entradas o eventos. Por ejemplo, se debe habilitar un botón de envío si el usuario ha ingresado todos los campos obligatorios.

- Transición de estado

En la técnica de transición de estado, los cambios en las condiciones de entrada cambian el estado de la aplicación bajo prueba (AUT). Esta técnica de prueba permite al probador probar el comportamiento de un AUT. El probador puede realizar esta acción ingresando varias condiciones de entrada en una secuencia. En la técnica de transición de estado, el equipo de prueba proporciona valores de prueba de entrada positivos y negativos para evaluar el comportamiento del sistema.

- Adivinando errores

Adivinando errores es una técnica de prueba de software basada en adivinar el error que puede prevalecer en el código. La técnica se basa en gran medida en la experiencia en la que los analistas de pruebas utilizan su experiencia para adivinar la parte problemática de la aplicación de prueba. Por lo tanto, los analistas de pruebas deben ser hábiles y experimentados para adivinar mejor los errores.