22_desarrollo_dirigido_por_pruebas

October 15, 2020

1 Desarrollo Dirigido por Pruebas (TDD).

Es una práctica cuyo objetivos es lograr un código limpio, seguro y que funcione.

La idea es que los requerimientos sean traducidos en pruebas, de este modo, cuando las pruebas pasen se garantizará que el software cumple con los requisitos que se han establecido.

1.1 Conceptos relacionados.

1.1.1 Deuda técnica.

Es el costo que implica retrabajar un artefacto por causas de su pobre desarrollo en etapas tempranas del ciclo de vida.

1.1.2 Escribir las pruebas primero (Test First Development).

Consiste en crear pruebas para cada funcionalidad que debe cumplir el producto de software.

1.1.3 Refactorización.

Es la acción de reescribir el código de un componente de software sin afectar su comportamiento.

1.2 Tipos de pruebas.

1.2.1 Pruebas funcionales.

Fuente: https://hackr.io/blog/types-of-software-testing

- **Pruebas unitarias.** Evalúan el correcto funcionamiento de cada componente de código
- **Pruebas de integración.** Evalúan el funcionamiento del sistema a partir de la inclusión de nuevos componentes.
- **Pruebas de extremo a extremo**. Evalúa que cada componente de software de un proceso funciones correctamente de principio a fin.
- **Pruebas de humo**. Son pruebas no exhaustivas que se le hacen a un producto de software previo a su liberación a producción para validar que las funcionalodades primordiales funcionan correctamente.
- **Pruebas de sanidad**. Son pruebas superficiales de los puntos más relevantes de un producto de software.
- **Pruebas de regresión**. Son pruebas exhaustivas que se hacen para validar que el sistema funciona correctamente después de cierto tiempo de estar en producción.

- **Pruebas de aceptación**. Son pruebas que validan al funcionalidad del producto ante el cliente.
- **Pruebas de caja blanca**. Son pruebas que inciden en la lógica interna de los componentes.
- **Pruebas de caja negra**. Son pruebas que inciden en el comportamiento de los componentes.
- **Pruebas de interfaz** Son pruebas sobre los puntos de acceso de los componentes de un sistema.

1.2.2 Pruebas no funcionales.

- **Pruebas de rendimiento**. Evalúan el uso de recursos de un sistema.
- **Pruebas de seguridad**. Evalúan las posibles vulnerabildiades de un sistema.
- **Pruebas de carga**. Evalúan la capacidad de un sistema ante cierto tipo de cargas de uso.
- **Pruebas de fallo**. Evalúan la resiliencia de un sistema.
- **Pruebas de compatibilidad**. Evalúan si un sistema puede interactuar con otros.
- **Pruebas de usabilidad**. Evalúan la experiencia general del uso de un sistema.
- **Pruebas de escalabilidad**. Evalúan la capacidad de crecer en cuestión de recursos de un sistema.
- **Pruebas de volumen**. Evalúan la capacidad de un sistema de gestionar grandes cantidades de datos.
- **Pruebas de estrés**. Evalúan el comportamiento de un sistema cuando se encuentra al límite de sus capacidades.
- **Pruebas de mantenibilidad**. Evalúan la facilidad de mantenimiento de un sistema.
- **Pruebas de cumplimiento (compliance)**. Evalúan si un sistema cumple con la normatividad aplicable.
- **Pruebas de eficiencia**. Evalúan si el sistema cumple con los objetivos funcionales, en tiempo razonable.
- **Pruebas de confiabilidad** Evalúan la disponibilidad y madurez de los procesos dentro del sistema.
- **Pruebas de resistencia.** Evalúa la operación del sistema con carga en tiempos largos.
- **Pruebas de recuperación ante desastres**. Evalúa la capacidad de un sistema para restablecer sus funcionalidades en caso de una emergencia crítica, así comod e mitgar riesgos.
- **Pruebas de localización**. Evalúa la capacidad de un sistema de poder ofercer servicios homogeneos a usuarios de diversas zonas geográficas.
- **Pruebas de internacionalización**. Evalúa la capacidad de un sistema de poder ofercer servicios homogeneos a usuarios de diversas zonas geográficas a novel internacional.