

Trabajo escrito

Por:

Sebastian Montalvo Londoño,

Luis Alfredo Vargas.

SENA

Centro Tecnológico del Mobiliario.

Análisis y desarrollo de software

2453914

Instructor:

Edwar Hernando Velásquez

22/08/2023

Test Driven Development.

Es una técnica de desarrollo la cual permite crear test antes de la implementación del código a producción lo cual es la esencia de esta técnica, el proceso se lleva a cabo antes de implementar para que se pueda ingresar la retroalimentación al código por medio de refactorización la cual se basa en cambiar el código principal sin cambiar su comportamiento en gran medida. Para hacer esta técnica se deben tener en cuenta los requerimientos que se pueden transmitir desde las historias de usuario o criterios de aceptación para posteriormente recrear estos requerimientos en pruebas automatizadas.

El TDD se divide en tres procesos principales que se denominan como: RED, GREEN Y REFACTOR.

RED: El proceso se basa en escribir un test que falle, se denomina RED ya que al hacer que el test falle las letras en la consola se resaltan en color rojo.

GREEN: El proceso se basa en hacer que el test que fallo anteriormente pase, se denomina GREEN debido a que cuando se corrigen los errores en el código, en la consola las letras se resaltan en verde.

REFACTOR: El proceso se basa en eliminar las redundancias del código, más específicamente se basa en mejorar todo lo que se pueda mejorar del código por medio de buenas prácticas.

Explicación más detallada de los pasos del TDD:

RED:

Paso 1: El cliente debe de escribir una historia de usuario la cual puede ser el

Una característica funcional que deba tener el programa.

Paso 2: se deben determinar criterios de aceptación de dicha historia de usuario con el propio cliente, el criterio de evaluación es una condición especial con la cual debe cumplir la historia de usuario.

Paso 3: Se debe seleccionar el criterio de evaluación mas sencillo y convertirlo en una prueba unitaria debido a su característica de probar una parte específica del sistema.

Paso 4: Se verifica que la prueba falla debido a que diferentes características específicas pedidas por el cliente posiblemente no este implementadas en el código aún.

GREEN:

Paso 5: De acuerdo al paso cuatro, se escribe el código el cual haga falta para completar la prueba, en este caso el código que cumpla con los requerimientos establecidos por el cliente.

Paso 6: Se realizan las pruebas automatizadas para comprobar que el código haya quedado en un estado funcional.

REFACTOR:

Paso 7: Se realiza los ajustes necesarios, reducción y optimización del código.

Paso 8: Se refactoriza el código y es sometido de nuevo a las pruebas.

Acceptance Test Driven Development (ATDD)

El Desarrollo guiado por pruebas de aceptación (ATDD por sus siglas en inglés) es una metodología de desarrollo que se basa principalmente en la constante comunicación entre tres entidades, el cliente, los desarrolladores y los testers y se hace por lo general antes de que los desarrolladores comiencen a desarrollar habiendo pruebas de aceptación.

El ATDD sigue los siguientes pasos:

- Se determinan la historia de usuario, teniendo en cuenta que ya se debieron haber priorizado dichas historias para poder seguir un orden de importancia.
- Se hacen las pruebas de aceptación, una vez especificadas las historias de usuario por medio de los colaboradores por parte del cliente.
- Una vez determinadas las pruebas de aceptación se automatizan las pruebas en el lenguaje de dominio del problema para determinar las funcionalidades que posiblemente falten.
- Después se implementa el código en el lenguaje de programación que se haya determinado.
- Finalmente, cuando los requisitos del cliente fueron satisfechos, se hacen los pequeños cambios y refactorizaciones que hagan falta.

Behaviour Driven Development (BDD)

El desarrollo dirigido por comportamiento es un conjunto de refinamiento de técnicas que ayudan a la entrega de un software de mejor calidad en un tiempo menor, además, el BDD se asimila mucho al TDD ya que siguen los siguientes pasos:

- Se describe un escenario.
- Se ejecuta un escenario que falla.
- Se escribe una prueba que corresponda a las especificaciones del escenario.
- Se escribe el código mas simple para pasar la prueba y el escenario.
- Finalmente se refactoriza y se vuelve a ejecutar el código

El BDD tiene el principal objetivo de que las historias de usuario dirijan el desarrollo y permite comprobar que las funcionalidades implementadas sean de acuerdo a lo requerido.

Testing exploratorio:

Este tipo de prueba ha sido de gran interés últimamente por parte de los tester debido a que, aparte de seguir paso a paso con las pruebas enumeradas en el informe de errores, se hacen pruebas adicionales para comprobar que dichos errores enumerados realmente fueron solucionados del todo, teniendo un resultado mejor. Este es un proceso de aprendizaje que hace al aplicativo bajo prueba y es útil para pruebas ágiles ya que no es necesaria una documentación de dichas pruebas exploratorias.

El testign exploratorio sigue las siguientes los siguientes pasos:

- Exploración del producto: Para conocer a fondo el funcionamiento y los objetivos que debe cumplir y por consiguiente hallar las posibles falencias e inestabilidades del producto.
- Diseño de pruebas: para crear las estrategias a seguir para evaluar el producto.
- Ejecución de pruebas: se explora el producto para determinar como funciona y sus puntos débiles.
- Heurística: que son las reglas generales para probar correctamente el producto.
- Resultados revisables: para que el tester pueda explicar los aspectos del programa y el como se cumplen los requisitos.

Automatización de pruebas de regresión:

Estas tienen la funcionalidad de comprobar que los cambios hechos no afectaron alguna funcionalidad anterior del programa, el problema con estas pruebas es que al hacerse de forma manual son muy costosas, tardadas y pueden tener errores debido al factor humano por eso una gran alternativa es automatizarlas haciendo que estas sean más concisas y objetivas.

