



Alquifiestas: Sensacional Eventos

Tarea de resumen 7

Hugo Eduardo Rivas Fajardo - 22500
José Santiago Pereira Alvarado - 22318
Nancy Gabriela Mazariegos Molina - 22513
Giovanni Alejandro Santos Hernández - 22523
Mauricio Julio Rodrigo Lemus Guzmán- 22461
Alexis Mesías Flores - 22562

• Que es Exploratory Testing.

a. Como se aplica este en Agile.

En Agile, la rapidez es clave, y dado que el Exploratory Testing no necesita una planificación detallada anticipada, puede llevarse a cabo de forma ágil en cada sprint. Se ajusta fácilmente a los cambios continuos y ofrece retroalimentación inmediata para que el equipo pueda modificar el desarrollo de manera oportuna.

b. Diferencia con testeo manual.

Aunque el Exploratory Testing también se realiza de manera manual, la diferencia con el testeo manual es que este sigue ciertos pasos predefinidos y en el Exploratory Testing el tester tiene la libertad de poder explorar el sistema como mejor le parezca.

c. Ventajas de este vs otros métodos de testing que hemos visto en el curso.

End to End

- Ventajas de Exploratory Testing:
 - Más flexible
 - Permite encontrar problemas específicos sin tener que recorrer todo el flujo.
 - Se ahorra tiempo

Smoke Testing

- Ventajas del Exploratory Testing:
 - Descubre problemas detallados y ocultos.

Pruebas de caja negra y blanca

- Ventajas del Exploratory Testing:
 - Flexibilidad
 - Permite abordar los problemas desde el punto de la caja negra y caja blanca sin tener un enfoque tan rígido

Pruebas funcionales

- Ventajas del Exploratory Testing:
 - Ya que no está limitado por las funcionalidades, permite encontrar problemas que con las pruebas funcionales no se podrían detectar.

UX/UI automatizada

- Ventajas del Exploratory Testing:
 - Se encuentran problemas de UX más sutiles, como problemas de flujo o usabilidad.

Pruebas de rendimiento

- Ventajas del Exploratory Testing:
 - Detecta problemas funcionales que en una prueba de rendimiento quedan al margen.

Pruebas de carga y estrés

- Ventajas del Exploratory Testing:

- Revela errores de funcionalidad y usabilidad, ya que las pruebas de carga y estrés son más técnicas.

Pruebas de integración

- Ventajas del Exploratory Testing:
 - Debido a que no hay restricciones, permite que el tester descubra errores de integración que en los casos predefinidos no se alcanzaría a detectar.

Pruebas de seguridad

- Ventajas del Exploratory Testing:
 - Detecta fallas o brechas de manera casual que otros métodos pueden no identificarlas.

d. Está de acuerdo que esta es una caja negra del testing?, en caso sea su respuesta de Si o No debe de justificar y explicar.

Si, ya que el tester no necesita entender el código, además el principal objetivo es la funcionalidad y no la implementación.

e. Define como incorpora este método hacia su proyecto en grupo.

Se incorpora este método hacia el proyecto realizado de diferentes maneras. Primero, como complemento a las pruebas automatizadas por medio de documentación en tiempo real. Esto permite registrar cualquier descubrimiento o error en el funcionamiento del código y que puede servir como un caso de prueba. Además, este método se utiliza al realizar cobertura al momento de generar pruebas en áreas de alto riesgo y al validar la experiencia del usuario.

• Que es Code Refactoring.

Code refactoring en general es el proceso por el cual se mejora el diseño interno de un código, esto con el fin de que el código en general sea más legible y entendible por sí solo para evitar confusiones con otros desarrolladores.

a. Como se aplica este en Agile.

R// Como cualquier implementación en esta metodología se espera que el código no sea perfecto a la primera vez, y que la perfección de dicho código sea gradualmente aplicada, la refactorización de código ayuda a mantener la calidad del código mientras va evolucionando durante su desarrollo en la metodología.

b. Qué retos importantes tiene para aplicarlo en su proyecto.

R// Algunos de los retos que podemos enfrentar a la hora de aplicar refactorización al código son los siguientes: resistencia al cambio, posible introducción de errores y tiempo limitado. Si bien es cierto que no se cambia la funcionalidad del código a la hora de refactorizar, la estructura interna del código si se cambia, y por el afán de

cambiar la estructura puede que se introduzcan errores, y la falta de tiempo es crucial para no caer en dichos errores.

c. Cuáles son los beneficios de aplicar este de forma general a cualquier proyecto.

R// En general es la menor cantidad de errores, deuda técnica, mayor eficiencia tanto del código como del equipo y facilidad de posibles nuevas integraciones al proyecto.

d. ¿Puede definir si existen distintos tipos de técnicas para aplicar este?.

R// Existen varias formas o métodos para aplicar la refactorización a un código algunas de ellas son las siguientes: Simplificación de expresiones condicionales, renombramiento de variables o métodos, reducción de código duplicado, eliminación de métodos innecesarios entre muchos otros.

e. Define cómo incorporar este método hacia su proyecto en grupo.

R// Para poderlo integrar esta técnica en nuestro proyecto pensamos aplicar las siguientes cosas: Implementar el hábito de refactorización continuo en el equipo, definir reglas, o estándares de codificación en nuestro proyecto, planificación de refactoring y por último utilizar herramientas para la refactorización.