Universidad de Guadalajara.

Centro Universitario de Ciencias Exactas e Ingenierías.

Ejercicio 3.1

Principios de prevención de defectos
Computación tolerante a fallas NRC 179961

Nombre del alumno: Loredo Padilla Orlando Javier

Código: 217560328

Departamento: Departamento de ciencias computacionales

Nombre del maestro: López Franco Michel Emanuel

Carrera: Ingeniería en computación

Sección: D06

Ciclo: 23B

Fecha: 4/9/2023.

Introducción

En el trabajo anterior vimos algunas herramientas que nos pueden ayudar a corregir los errores presentados en nuestros programas, pero esos sistemas estábamos contemplando la solución a un problema que ya ocurrió, ahora cambiaremos el enfoque a usar herramientas para evitar errores que sabemos que pueden ocurrir, pero no podremos evitar, véase el ejemplo un apagón de luz al momento de estar contestando un formulario.

Métodos para la prevención de defectos

Algunos de los métodos que podemos utilizar para prevención de defectos son los siguientes:

Revisión e inspección

Este método incluye la revisión por parte de un miembro individual del equipo (autocomprobación), revisiones por pares e inspección de todos los productos de trabajo. Es un método muy simple y llega ser tedioso, pero hacer una revisión del código muchas veces es lo único que se necesita para tener un buen código.

Tutorial

Este es cercano a lo que sería una revisión, pero se relaciona principalmente con comparar el sistema con el prototipo, lo que le dará una mejor idea sobre la corrección y/o la apariencia del sistema ya que el tener una referencia ayuda más a tener un desarrollo más guiado y seguro.

Registro y documentación de defectos

Este método proporciona información clave, argumentos / parámetros que se pueden utilizar para respaldar el análisis de defectos. Aquí deberemos estar haciendo un reporte de lo que va ocurriendo así cuando volvamos a trabajar u otra persona retome nuestro trabajo estará al tanto de lo que pueden originar hacer tales cambios al código.

Análisis de la causa raíz

El análisis de la causa raíz se basa en dos enfoques principales:

Análisis de Pareto

El análisis de Pareto es una técnica formal y simple que ayuda a priorizar el orden de resolución del problema para lograr el máximo impacto. Afirma que el 80% del problema surge por razones del 20%.

Los problemas una vez identificados se priorizan de acuerdo con la frecuencia y se realiza un análisis estadístico detallado para encontrar qué 20% de las razones atribuyen al 80% de los problemas. Simplemente centrándose en ese pequeño campo y eliminándolas, los resultados están garantizados mientras se optimiza la extensión del trabajo involucrado.

Al centrarnos en un pequeño campo podremos trabajar más rápido mientras que lo demás se va solucionando con lo cambios que hagamos sin intervenir directamente.

Análisis de espina de pescado

Se conoce como Análisis de Ishikawa, este método es una técnica de análisis de causa raíz más visual. No hay estadísticas involucradas ya que se basa en una lluvia de ideas de todo el equipo.

En un diagrama ubicaremos el problema en el lado más a la derecha y en una línea horizontal que lo atraviese, se enumeran las diversas causas. La rama que tenga más huesos de causa-subcláusula es el problema más grave y que se debe trabajar para su eliminación.

Conclusión

Muchas veces hay situaciones en las que no podremos evitar que ocurran accidentes o errores y esto se debe a que no somos capaces de controlar todo lo que sucede en nuestro alrededor, pero como programadores podremos crear medidas para evitar o reducir los daños que causaran esos errores que sabemos que pueden ocurrir.

Con el trabajo hecho podemos apreciar que existen muchas maneras de crear un salvavidas para nuestros progresos. Sin embargo, estos métodos son un poco más prácticos por lo cual nos tocará a nosotros la mayor parte del trabajo, diferente a los sistemas que corrigen a los errores ya que de ahí ya obtenemos lo que está mal, aquí nosotros, o nuestro equipo, hará muchas veces pruebas en las que se deberán de estar buscando los posibles errores que pueden ocurrir durante la ejecución del sistema y a partir de ese análisis deberemos de buscar el como evitar o amortiguar los daños que podrán causar esos fallos si se llegaran a presentar.

Referencias

- S/A (S/F). Métodos y técnicas de prevención de defectos en Myservername.com. Recuperado el 1/9/2023 en: https://spa.myservername.com/defect-severity-priority-testing-with-examples
- Ventura M. (S/F). Costo de la Calidad en el Desarrollo de Software. Recuperado el 1/9/2023 en: http://www.paginaspersonales.unam.mx/files/69/Publica_20110622215614.pdf