

UNIVERSIDAD DE GUADALAJARA
CENTRO UNIVERSITARIO DE CIENCIAS EXACTAS E INGENIERIAS

Materia: Computación tolerante a fallas

Clave: I7036

Sección: D06



NRC: 179961

Principio de prevención de defectos

Código estudiante: 221350567

Alumno: Rauf Alfonso Hamden Estrada

Carrera: Ingeniería en Computación

Fecha: 3/09/2023

Docente: Michel Emanuel Lopez Franco

2023B

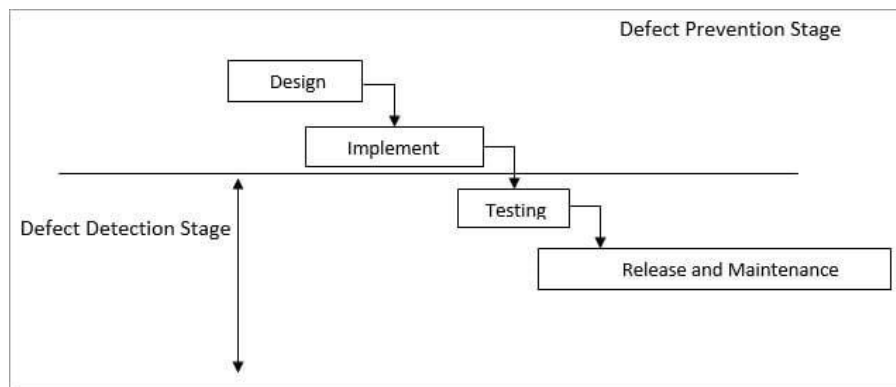
Introducción

En nuestra vida cotidiana utilizamos multitud de aparatos electrodomésticos, vehículos, equipos informáticos que, con mayor o menor frecuencia, fallan o presentan defectos. Los fabricantes, a medida que los van evolucionando, se ocupan también analizar estos defectos con el fin de evitar que se repitan en futuras versiones. En la mayoría de los casos, se requiere tener los menos posibles con los dispositivos por lo cual cada día se buscan nuevas alternativas para poder evitar la menor cantidad de estos

Desarrollo

La prevención de defectos es un paso o actividad crucial en cualquier proceso de desarrollo de software y, como se puede ver en el diagrama a continuación, es casi la mitad de nuestras tareas de prueba

Para implantar un proceso de pruebas basado en la prevención de defectos es necesario definir qué parámetros van a servir para decidir si un defecto debe ser analizado o no. Partiendo de la base de que los recursos son limitados, no es viable analizar la causa raíz de todos los defectos, sino sólo de aquellos que por su tipología merecen ese esfuerzo. Los parámetros que habitualmente se utilizan son: el daño potencial del defecto (económico o de imagen), frecuencia de ocurrencia, el coste que supone solucionarlo y el impacto del defecto en el conjunto de la aplicación.



Métodos y técnicas de prevención de defectos

1) Revisión e inspección: Este método incluye la revisión por parte de un miembro individual del equipo, revisiones por pares e inspección de todos los productos de trabajo.

2) Tutorial: Esto es más o menos como 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.

3) Registro y documentación de defectos: Este método proporciona información clave, argumentos / parámetros que se pueden utilizar para respaldar el análisis de defectos.

4) Análisis de la causa raíz: El análisis de la causa raíz incluye dos enfoques principales:

Análisis de Pareto:

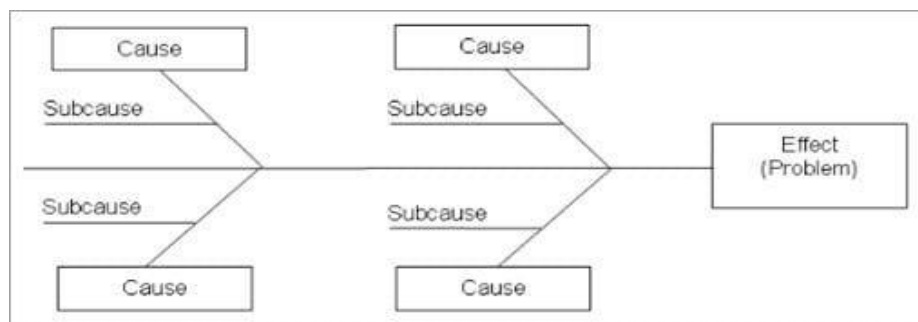
El análisis de Pareto es una técnica formal y simple que ayuda a priorizar el orden de resolución del problema para lograr el máximo impacto. Afirma que el 80% del problema surge por razones del 20%.

Por lo tanto, 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 esas razones del 20% y eliminándolas, los resultados están garantizados mientras se optimiza la extensión del trabajo involucrado.

Análisis de espina de pescado:

También conocido como Análisis de Ishikawa este método es una técnica de análisis de causa raíz más visual. No hay estadísticas involucradas ya que este método se basa en una lluvia de ideas de todo el equipo. El siguiente diagrama ayuda a comprender esto mejor.

El problema se escribe primero en el lado más a la derecha y en la línea horizontal que lo atraviesa, se enumeran las diversas causas. La rama que tiene más huesos de causa-subcláusula es el problema más grave y que se debe trabajar para su eliminación. Esta técnica también se llama a veces análisis de causa y efecto .



Conclusión

La prevención de defectos desempeña un papel esencial y crítico en todo el proceso de desarrollo de software. No solo ayuda a garantizar la calidad del producto final, sino que también tiene un impacto positivo en la eficiencia del proyecto. Esto se centra en problemas desde las etapas iniciales del proceso de desarrollo, antes de que tengan la oportunidad de manifestarse en la aplicación final. La búsqueda de la causa principal es una herramienta fundamental para identificar y eliminar problemas en su origen, lo que contribuye a mantener un software sólido y confiable.

Referencias

- Métodos y técnicas de prevención de defectos. (s. f.). Seguimiento De Defectos De Errores. <https://spa.myservername.com/defect-prevention-methods>
- Prueba de Software. (1d. C.). IBM. Recuperado 3 de septiembre de 2021, de <https://www.ibm.com/mx-es/topics/soft>
- Hamilton, D. (2023, 4 agosto). Estándares de codificación de software y pautas de programación. Parasoft. <https://es.parasoft.com/blog/an-ounce-of-prevention-software-safety-security-through-coding-standards/>