Ejercicio 3: Principios de prevención defectos.

Materia: Computación tolerante a fallas

Universidad de Guadalajara

Profesor: Michel Emanuel López Franco

5/02/2024

José Maximiliano Díaz Méndez

Introducción

Los defectos en sistema son algo imposible de evitar. Aun así, se han desarrollado diferente métodos y técnicas con el objetivo de prevenir y suprimir los defectos que se pueden presentar en un programa como los que se listan a continuación.

Métodos y técnicas de prevención de defectos.

- **Análisis de requisitos de software.**
  La mayoría de los defectos en un software son ocasionados debido a errores en los requisitos y el diseño de este. Debido a esto es necesario ser eficiente y minucioso a la hora de generar los requisitos y el diseño del software.

- **Revisión e inspección.**
  Si se implementa bien, es posible identificar y eliminar antes de que estos ocurran y afecten al ambiente de producción. Hay dos tipos de revisión, es decir, auto-revisión y revisión por pares en la que se identifican defecto y se presentan sugerencias de mejoras.

- **Registro y documentación de defectos.**
  Es necesario mantener un registro de los defectos encontrados ya que permite obtener conocimiento y comprensión de los defectos ocurridos que ayudaran a tomar medidas y acciones para identificar y prevenir errores futuros, así como actuales.

- **Análisis de causa raíz.**
  Consiste en busca la razón principal detrás del defecto la cual una vez encontrada facilita el tomar acciones para eliminar el defecto y prevenir defectos similares.

- **Pruebas unitarias.**
  Son pruebas automatizadas enfocadas a probar unidades de código para asegurar que se comporten de acuerdo con lo esperado conforme pase el tiempo y se realicen modificaciones en el código.

- **Desarrollo guiado por pruebas (Test-Driven Development).**
  De la mano con el punto anterior, es un proceso de desarrollo donde se escriben primero las pruebas del resultado esperado para después implementar el código de esta forma asegurando se cumplan los requerimientos.

- **Integración continua y Desarrollo continuo (Cotinuos Integration / Continuous Deployment).**
  Esta práctica de desarrollo se basa en que los cambios hechos al código sean probados y lanzados a producción de forma automatica.

- **Análisis estático de software**
  Mediante el uso de herramientas comunmente conocidas como “linters” se analiza el código para encontrar posibles defectos sin necesidad de ejecutarlo.

- **Manejo de errores.**

Una buena gestión de los errores previene que estos se propaguen y permite que el software pueda manejar situaciones inesperadas.

Conclusión

Las opciones que tenemos para la prevención de errores son muy variadas y se complementan bastante entre sí ya que la mayoría se basan en el análisis del código de distintas formas ya sea automatizadas, manuales y en pares. Además también de contar con herramientas como el uso de pruebas automatizadas para detectar errores que pueden escaparse del proceso de análisis tras cambiar requerimientos y código.

Referencias

danf7861. (2023, 6 febrero). Fault reduction techniques in software engineering. GeeksforGeeks. Recuperado 3 de febrero de 2024, de https://www.geeksforgeeks.org/fault-reduction-techniques-in-software-engineering/

Greyrat, R. (2022, 5 julio). Métodos y técnicas de prevención de defectos – Barcelona Geeks. Barcelona Geeks. Recuperado 3 de febrero de 2024, de https://barcelonageeks.com/metodos-y-tecnicas-de-prevencion-de-defectos/