



Ejercicio 3: Principios de prevención defectos 2.

Materia: Computación tolerante a fallas

Universidad de Guadalajara

Profesor: Michel Emanuel López Franco

5/02/2024

José Maximiliano Díaz Méndez

Introducción

A la hora de escribir software nos encontraremos en la necesidad de gestionar los defectos que se pueden presentar en el desarrollo. Una herramienta que podemos utilizar es Orthogonal Defect Classification o en español, Clasificación Ortogonal de defectos la cual nos brinda de pautas para gestionar mejor y prevenir los defectos en el desarrollo.

Orthogonal Defect Classification

¿Qué es?

En una metodología usada para clasificar y analizar defectos de manera sistemática. Lo de que sea ortogonal se refiere a la independencia entre las categorías utilizadas para clasificar los defectos buscando que estas no se superpongan y un defecto solo puede pertenecer a una categoría.

Valor

- Provee retroalimentación rápida y efectiva a los desarrolladores (Software Quality Experts, 2018).
- Captura información de defectos que ocurrieron durante el desarrollo y el uso del programa (Software Quality Experts, 2018).
- Permite entender las tendencias en los defectos a lo largo de sus fases de vida debido a la consistencia al clasificar los defectos (Software Quality Experts, 2018).
- A través de mediciones y análisis multidimensionales ayuda a los desarrolladores para administrar el proceso de desarrollo y calidad del producto (Software Quality Experts, 2018).

Un defecto mediante ODC pasa por dos secciones:

Sección de apertura

Cuando se encuentra se clasifican los siguiente atributos.

- Actividad: Es la actividad que se estaba realizando al momento de encontrar el defecto
- Desencadenante: El entorno o condición que se tiene que dar para que ocurra el error en el sistema a causa del defecto.
- Impacto: El impacto esperado que pudo provocar el defecto en caso de no lograr haber llegado al cliente o el impacto ocurrido en caso de haber llegado al cliente.

Sección de cierre

Una vez se sabe como se corrijio el defecto se clasifican los siguientes atributos.

- Objetivo: Representa la identidad de alto nivel de la entidad que se arreglo (Diseño, código, etc).
- Defect type: Representa la naturaleza de la corrección actual que se realizó.
- Source: Identifica el origen del objetivo (Diseño, código, etc) que tenía el defecto.
- Age: Identifica la historia del objetivo que tenía el defecto.

Conclusión

Al brindarnos pautas de lo que debemos recopilar a la hora de presentar y solucionar defectos en el sistema nos hace recabar atributos que podemos usar para clasificar de esta forma se documentan mejor los errores y disponemos de más información para prevenir y analizar el proceso de desarrollo para encontrar el origen de los errores más comunes como podría ser una deficiente especificación de requerimientos, negligencias en el desarrollo, falta de experiencia en redes, etc. Aunque no es necesario hacer uso de esta metodología u otras al desarrollar software se puede apreciar las ventajas que estas brindan.

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/>