



Universidad de Guadalajara
Centro Universitario de Ciencias Exactas e Ingenierías
División de Tecnologías para la Integración Ciber-Humana
Departamento de Ciencias Computacionales
Ingeniería en Computación



Ejercicio de Aprendizaje 05.

Principios y prevención de defectos 2: Orthogonal Defect Classification.

Alumno

Hernández Cortez Kevin Uriel.

Materia

Computación Tolerante a Fallas.

I7036, D06, 2024A

Profesor

Lopez Franco Michel Emanuel.

Orthogonal Defect Classification (ODC): Una Exploración Detallada

Introducción

En el vertiginoso mundo del desarrollo de software, la calidad es un factor crucial. Los defectos, aunque inevitables, pueden afectar significativamente la eficiencia y la confiabilidad de un producto. Aquí es donde entra en juego el concepto de Orthogonal Defect Classification (ODC). Esta metodología proporciona una visión profunda y estructurada de los defectos, permitiendo a los equipos de desarrollo comprender mejor su proceso y tomar medidas para mejorarlo.

Desarrollo

ODC no solo es una técnica de clasificación de defectos; es un enfoque para analizar y comprender las imperfecciones en el software. Cada defecto se etiqueta con atributos específicos, como la actividad en la que se realizó, el desencadenante que provocó el defecto y la compatibilidad con versiones anteriores. Estos atributos proporcionan y permiten una segmentación más profunda de los defectos.

Los atributos de clasificación son los siguientes:

1. **Actividad:** Se refiere a la tarea en la que se estaba trabajando cuando se detectó el defecto. Puede ser una inspección de código, diseño, pruebas unitarias o pruebas de sistema.
2. **Desencadenante:** Es la acción o evento que provocó la aparición del defecto. Identificar el desencadenante ayuda a prevenir futuros problemas similares.
3. **Compatibilidad con versiones anteriores:** Evalúa cómo la versión actual del producto interactúa con versiones anteriores.
4. **Compatibilidad lateral:** Considera cómo el producto debe interactuar con otros productos similares.

ODC utiliza la información semántica de los defectos para identificar patrones y relaciones. Por ejemplo, si se detecta un defecto durante una inspección de código, ODC analiza qué actividad específica desencadenó ese defecto y cómo se relaciona con otros defectos similares. Estas relaciones ayudan a comprender las causas subyacentes y a tomar decisiones informadas.

ODC convierte la información de los defectos en una medida de proceso. Esto permite identificar áreas problemáticas, tendencias y oportunidades de mejora. Los equipos pueden ajustar sus prácticas en función de estas métricas para optimizar la calidad del software.

Conclusión

Yo concluyo que ODC no es solo una herramienta técnica, sino también un enfoque preciso. Al usar ODC, los equipos de desarrollo pueden fomentar una mentalidad de mejora continua y de colaboración. La comprensión profunda de los defectos no solo ayuda a resolver problemas actuales, sino que también crea el camino para un software más robusto y confiable en el futuro. ODC es una brújula que guía a los desarrolladores hacia la excelencia en la creación de software.

Bibliografía

- Software Quality Exp. (2018, 21 marzo). What is Orthogonal Defect Classification (ODC)? By Vivek Vasudeva. *Medium*. <https://medium.com/@SWQuality3/what-is-orthogonal-defect-classification-odc-by-vivek-vasudeva-f2e49917f478>
- *Orthogonal defect classification-a concept for in-process measurements*. (1992, 1 noviembre). IEEE Journals & Magazine | IEEE Xplore. <https://ieeexplore.ieee.org/document/177364>
- Software Quality Exp. (2018b, mayo 29). Software testing and quality via ODC (Orthogonal Defect Classification). *Medium*. <https://medium.com/@SWQuality3/software-testing-and-quality-via-odc-d4b7cf7cc684>