

# **CENTRO UNIVERSITARIO DE CIENCIAS EXACTAS E INGENIERIAS**

## **DEPARTAMENTO DE CIENCIAS COMPUTACIONALES**



Materia: Computación Tolerante a Fallas

Actividad: Principios y prevención de defectos (2)

Nombre: Rodríguez de León Ricardo Emmanuel

## Introducción

Hablaremos sobre el Orthogonal Defect Classification (en español, Clasificación Ortogonal de Defectos), de como se usa, y el porqué.

## Desarrollo

Como se mencionó, se hablará sobre esta clasificación, para así tener prevención de defectos, este método por decirlo de una manera se refiere a la actividad que se esta realizando cuando se produce la falla. Por ejemplo, en el caso de las inspecciones, que es la actividad que detecta el defecto directamente.

Tiene 4 pasos, la inspección de código, de diseño, unitaria, de integración y de sistema.

La inspección de código, como bien se menciona, se revisará el código para que se pueda detectar e identificar anomalías en un producto de software.

En cuanto al diseño, simplemente es sobre los documentos de diseño, para poder inspeccionarlos también.

Luego esta la prueba unitaria, la cual es una forma de verificar una unidad de software de forma aislada.

Prueba de integración, es aquella, que se realiza para aprobar las pruebas unitarias. Es decir, verifica la comunicación entre dos o más competentes.

Por último, está la prueba de sistema donde tienen como objetivo verificar el sistema software en su totalidad, para comprobar si el mismo cumple con los requerimientos.

Como se puede observar, a fin de cuentas, todo lo que se debe hacer, es que siempre se tiene que revisar para que no haya algún defecto.

Defecto. Este atributo es utilizado para proveer una medida de los aspectos de verificación del proceso de desarrollo y se mide en la fase de Apertura.

Compatibilidad con versiones Está relacionado con la comprensión de cómo la versión actual del producto funciona con

versiones anteriores.

Compatibilidad Lateral está relacionado con el producto el cual debe trabajar con otros productos con la misma configuración de software.

Conformidad del diseño está relacionado con la completitud del diseño del producto respecto a los requerimientos y objetivos del producto.

Concurrencia está relacionado con la comprensión de las cuestiones de timing y serialización relacionadas a la implementación del producto. Por ejemplo, mecanismos de bloqueo, regiones compartidas y secciones críticas.

Semántica Operacional relacionado a la comprensión del flujo de la lógica respecto a la implementación de un diseño.

Consistencia/Complejidad de Documento está relacionado con la completitud de un diseño y asegurar la consistencia entre las distintas partes del diseño propuesto y la implementación.

Situaciones Raras relacionado con implementaciones inusuales, idiosincrasias o información específica de un dominio que no es común.

Los valores posibles para el atributo Trigger en Pruebas Unitarias o Funcionales son:

Prueba de Cubrimiento se refiere a la ejecución de una función a través de varias entradas para cubrir la mayor cantidad de casos posibles del dominio de parámetros. Es un trigger de prueba de caja negra.

Prueba Secuencial estos son casos de pruebas que apuntan a ejecutar múltiples secciones del código con diferentes secuencias. También es de caja negra.

## Conclusión

Es evidente que todo esto está relacionado a que siempre debe haber un proceso en donde se debe hacer una revisión si o si, para que este no tenga defectos o tengan lo mínimo, para el correcto funcionamiento, porque a fin de cuentas, si no se trabaja correctamente, el cliente siempre verá cuando algo está bien hecho y cuando no, por eso es importante la revisión y tener un proceso que se debe seguir, porque también pueden llegar a problemas más graves y pues eso se debe evitar lo más que se pueda.

## Referencia

<https://www.uv.mx/personal/ermeneses/files/2017/08/Clasificacion-Ortogonal-de-Defectos.pdf>