



**Universidad de Guadalajara.**  
**Centro Universitario de Ciencias Exactas e Ingenierías.**



## **Ejercicio 3.2**

### **Principios y prevención de defectos (2)**

### **Computación tolerante a fallas NRC 179961**

**Nombre del alumno:** Loredó Padilla Orlando Javier

**Código:** 217560328

**Departamento:** Departamento de ciencias computacionales

**Nombre del maestro:** López Franco Michel Emanuel

**Carrera:** Ingeniería en computación

**Sección:** D06

**Ciclo:** 23B

**Fecha:** 4/9/2023.

## Introducción

En esta ocasión revisaremos de que trata el Orthogonal Defect Classification, o ODC por sus siglas, y como este sistema puede ayudarnos en la prevención de defectos.

## Orthogonal Defect Classification

Es sistema fue desarrollado por IBM con la intención de poder clasificar defectos y que a partir de esto se pueda desarrollar retroalimentaciones que puedan ayudar al desarrollo de proyectos.

Tiene como objetivo encontrar el paradigma para medir la relación de causa-efecto de los defectos, y que con ello se pueda obtener con facilidad la razón del defecto. Además, se deseaba desarrollar una clasificación de defectos que fuera simple y que estuviera lo más libre posible de errores.

Al crear estas separaciones se obtuvo una visión más limpia al momento de trabajar al mismo tiempo, al clasificar estos errores se puede identificar o relacionar el error con la fase en la que se encuentra un código.

En el ODC se le dio a los defectos dos características: el primero es tipo de defecto y el segundo es el desencadenador (o trigger) de un defecto.

Para clasificar un defecto lo haremos en base a semántica de la reparación que se necesitó hacer para remover al defecto del sistema, aunque La selección del tipo de defecto dependerá de la corrección hecha por el desarrollador. Los tipos de defectos definidos son:

- **Función:** Es cuando afecta a las interfaces de usuario, del producto, con el hardware, y las estructuras de datos. Se requerirá un cambio formal en el diseño.
- **Asignación:** Indica errores en el código, al inicializar o asignar valores de variables o estructuras de datos.
- **Interfase.** Se refiere a errores en las interacciones entre componentes, módulos u otros dispositivos a través de las llamadas a funciones, paso de parámetros, etc.
- **Chequeo:** Errores en la lógica del programa, el cual no hace la verificación de los datos y valores de manera correcta.
- **Serialización/temporal:** Errores que tienen que ver con los recursos compartidos y de tiempo real.
- **Construcción/Empaquetado (Build/Package/Merge):** Errores que ocurren debido a problemas en las librerías, así como problemas en el control de cambios y versiones.
- **Documentación:** Errores que pueden afectar a la documentación de los artefactos o al mantenimiento del sistema.
- **Algoritmo:** Problemas de eficiencia o corrección de una tarea, a través del algoritmo que define a esa tarea.

Los tipos de defectos son ortogonales, con esto nos referimos a que son mutuamente excluyentes, de esta manera se asegura que, al encontrarse un defecto, este será solamente de un tipo posible, y no podrá caerse en la situación de clasificar incorrectamente a un defecto debido a que no es posible colocar uno en dos categorías.

Por otra parte, tenemos el trigger, este se refiere a la condición que conduce a una falla. Si el tipo se trataba de la consecuencia acá nos referimos a la causa. Es no se clasifican por tipos si engloban diversas situaciones que pueden dar origen a un error. Para dar con esto se deben de hacer revisiones del programa y descartar cualquier posible de que una opción pueda resultar en un error, aunque esto a veces se puede volver complicado ya que hay errores que no veremos usando un programa de manera simple. De los triggers encontrados se puede obtener una retroalimentación para mejorar el proyecto en el que se está trabajando.

## **Conclusión**

El ODC es demasiado útil al estar desarrollando grandes programas ya que nos permite tener un mayor orden y control de los errores que se pueden ir presentando, si clasificamos por tipo los defectos podemos dar con más rapidez el tipo de solución que se necesita, de esta manera se optimiza el trabajo y se reducen los tiempos dedicados en encontrar la solución del error. A pesar de ser muy simple el sistema desarrollado por IBM, el tener un sistema que puede ofrecernos la solución a problemas que se originen durante el desarrollo es una gran ayuda que se nos brinda. En resumen el tener un sistema que organice problemas y que ofrezca una mejor visión de como solucionarlos es un sistema que puede aumentar la eficiencia a la hora de hacer correcciones.

## **Referencias**

- Herrera G. (2005). Clasificación y Análisis Post-Proceso de Defectos en Proyectos de Desarrollo de Software. Recuperado el 1/9/2023 en: <https://cimat.repositorioinstitucional.mx/jspui/bitstream/1008/104/2/TE%20264.pdf>
- Software Quality Exp (2023). What is Orthogonal Defect Classification (ODC)? en Medium. Recuperado el 1/9/2023 en: <https://medium.com/@SWQuality3/what-is-orthogonal-defect-classification-odc-by-vivek-vasudeva-f2e49917f478>

