



## **COMPUTACION TOLERANTE A FALLAS**



### **Investigación 02 – Principios para l**

### **prevención de defectos**

**Universidad de Guadalajara**

Nombre: Marco Aurelio Domínguez Amezcua

Código: 216818534

Carrera: Ingeniería en computación.

Fecha: 03/09/23

## **Objetivo**

Leer el siguiente artículo e investigar sobre diferentes métodos para la prevención de defectos.

## **Desarrollo**

### **PSP**

Es un proceso de desarrollo de software que se centra en mejorar la calidad y la productividad del programador individual. Una de las fases clave del PSP es la prevención de defectos. También ayuda a comprender y mejorar su desempeño aportando disciplina a la forma en que desarrollan software, durante y previo del código. Ayuda a los desarrolladores a mejorar la calidad de sus productos, de como elaborar un plan sólido y como asumir compromisos.

#### Objetivos del PSP

El objetivo principal es proporcionar a los ingenieros métodos disciplinados para mejorar los procesos de desarrollo de software personal. El PSP ayuda a los ingenieros de software a:

- Mejorar sus habilidades de estimación y planificación.
- Hacer compromisos que se puedan cumplir.
- Gestionar la calidad de sus proyectos.
- Reducir el número de defectos en su trabajo.

#### Estructura

El PSP sigue una estructura evolutiva, cada vez el ingeniero avanza niveles, empezando por el nivel PSP0 y progresa a la madurez del proceso hasta el nivel final PSP2.1. Entonces, los niveles serían los siguientes:

- PSP0, PSP0.1 (Introduce disciplina y medición de procesos).
- PSP1, PSP1.1 (Presenta la estimación y planificación).
- PSP2, PSP2.1 (Presenta la gestión de calidad y el diseño).

## **Blocks**

Es una metodología o enfoque que se centra en la prevención de defectos mediante la identificación y el abordaje de las causas fundamentales. Regularmente, no se asocia a metodologías específicas de prevención de defectos. Sin embargo, proporciona información sobre un enfoque más general para abordar los problemas desde la raíz y tomar medidas preventivas rápidas.

## Analisis de causa raíz (RCA - Root Cause Analysis)

Es una metodología que se utiliza en una variedad de campos, incluyendo la gestión de calidad y la mejora continua, para identificar y abordar las causas fundamentales de un problema o defecto. El objetivo es entender por qué ocurrió un problema en primer lugar y tomar medidas preventivas para evitar su recurrencia.

Unos aspectos claves para el análisis de causa raíz serian:

- **Identificación del problema:** Identificar el problema o defecto de manera clara y específica. Esto implica definir el problema, recopilar datos relevantes y comprender su impacto en el proceso o producto.
- **Recopilación de datos:** Se reúnen datos e información relacionados al problema, lo que puede incluir entrevistas, observaciones, análisis de documentos y datos cuantitativos.
- **Análisis de causas:** Se usan diferentes técnicas y herramientas para analizar las causas del problema. Se incluyen diagramas de espina de pescado (Ishikawa), diagramas de flujo, entre otros.
- **Identificación de la causa raíz:** Identificar la causa raíz o causas fundamentales que contribuyeron al problema.
- **Desarrollo de soluciones preventivas:** Se desarrollan soluciones después de identificar la causa raíz, para evitar que el problema ocurra nuevamente en el futuro.
- **Implementación y seguimiento:** Implementación de las soluciones preventivas y se realiza un seguimiento para asegurarse de que sean afectivas y que el problema no se repita.

Finalmente, se concluye en que el análisis de causa raíz es una herramienta poderosa para abordar problemas y defectos de manera sistemática y preventiva. Ayuda a las organizaciones a mejorar continuamente sus procesos y productos al centrarse en las causas fundamentales en lugar de tratar solo los síntomas superficiales. Esto puede llevar a una mayor eficiencia, calidad y satisfacción del cliente.

## ODC

Orthogonal Defect Classification (ODC) es un enfoque y una metodología utilizada en la gestión de calidad del software para clasificar y analizar los defectos o problemas identificados durante el proceso de desarrollo de software y pruebas. El objetivo principal de ODC es proporcionar una forma sistemática de categorizar los

defectos de manera que se puedan identificar las causas subyacentes y tomar medidas para prevenir su recurrencia en el futuro.

Las características clave de Orthogonal Defect Classification incluyen:

- **Categorización sistemática:** ODC utiliza una serie de categorías predefinidas para clasificar los defectos de manera consistente. Estas categorías son independientes entre sí, lo que significa que cada defecto se clasifica en una sola categoría.
- **Métricas y análisis:** ODC recopila datos sobre los defectos clasificados y permite realizar análisis para identificar patrones y tendencias. Esto ayuda a la organización a comprender las áreas problemáticas y priorizar las mejoras.
- **Prevención y mejora continua:** Una vez que se clasifican los defectos, se pueden tomar medidas para abordar las causas subyacentes y prevenir futuros defectos similares. Esto contribuye a una mejora continua en la calidad del software.
- **Basado en datos:** ODC se basa en datos objetivos y cuantitativos, lo que lo hace útil para tomar decisiones informadas sobre la gestión de la calidad del software.

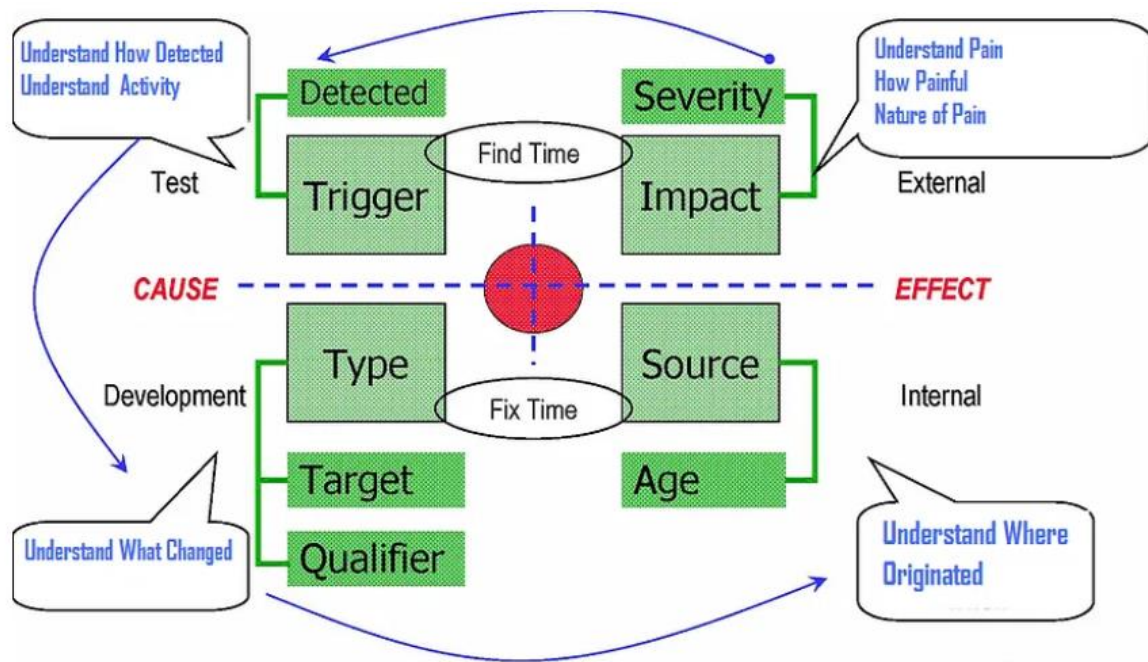
## Secciones

Hay muchas secciones por las cuales pasa un defecto, regularmente este defecto tiene que pasar a través de 2 secciones por lo menos, a continuación, hay una lista de algunas de ellas:

- **Sección Abridor:** Cuando se encuentra un defecto, se pueden clasificar los siguientes atributos:
  - **Actividad:** Esta es la actividad real realizada en el momento del descubrimiento del defecto. Por ejemplo, durante la fase de prueba de funcionamiento, un ingeniero podría decidir realizar una inspección del código. La fase sería prueba de funcionamiento, pero la actividad es inspección de código.
  - **Desencadenante:** El entorno o condición que tenía que existir para que apareciera el defecto. ¿Qué se necesita para reproducir el defecto? Durante las actividades de Revisión e Inspección, elija la selección que mejor describa lo que estaba pensando cuando descubrió el defecto. Para otros defectos, haga coincidir la descripción con el entorno o condición que fue el catalizador de la falla.
  - **Impacto:** para defectos en el proceso, seleccione el impacto que considere que el defecto habría tenido en el cliente si hubiera escapado al campo. Para defectos reportados en campo, seleccione el impacto que tuvo la falla en el cliente.

También mencionar que existen secciones que abarcar objetivos, tipos de defectos, calificadores, fuentes de origen y antigüedad, entre otras.

### Construir Relaciones Causa-Efecto



### Valores

Con los valores que aporta esta metodología nos ayuda a encontrar más rápido soluciones y segmentarlas, debido a que ya hubo un análisis previo a los defectos y ya se está trabajando en prevenirlos de una manera más ágil.

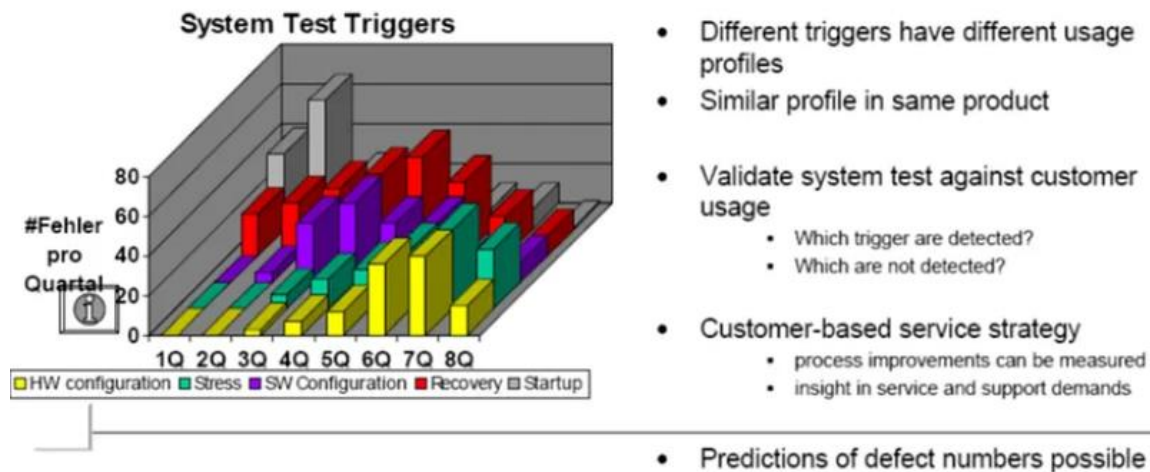
Esta metodología aporta mucho valor para la continuidad del proyecto, nos permite saber que hacer comentarios rápidos y efectivos para los desarrolladores es una forma de ganar tiempo. Además, el valor que cada vez se va agregando es mayor porque cada vez que se prueba o utiliza existe más información que se va añadiendo y empieza a abarcar más espacio, esto con el fin de que en algún momento se tenga la manera de corregir cualquier error ya que los errores que se han detectado ya han sido resueltos. Tal como se menciona anteriormente, este tipo de pruebas nos explica los defectos que ocurrieron durante la falla de desarrollo y uso de campo, lo que nos permite tener un contexto más grande y una solución más sencilla.

También nos permite comprender tendencias de los defectos que existen a lo largo de las fases del ciclo de vida debido a la coherencia de los tipos de defectos, pero al ya tener información recaudada de dichos defectos es más fácil identificarlos y

repararlos. Por otra parte, a través de mediciones y análisis multidimensionales, ODC ayuda a los desarrolladores a gestionar adecuadamente sus procesos de desarrollo y la calidad de sus productos.

## Beneficios de las pruebas de software

Se puede hacer mediciones o graficas que muestren todas las métricas y datos que se usan y procesas con ODC, nos dan una respuesta visual, en la siguiente imagen se tiene la medición por cuartos, en los cuales se ven cada cuanto cuarto se ven con más relevancia las respuestas de las pruebas.



## **Resumen**

En resumen, todas estas son metodologías que ayudan a prevenir defectos en cualquier proyecto de desarrollo de software, tanto a nivel de recurso humano y planeaciones del proyecto. Se debe tener una metodología útil para que los creadores del software puedan desarrollar correctamente su trabajo, así como apegarse a las normas y formas de trabajar. También, el software debe ser pronosticado y planeado de forma correcta para que su desarrollo no sea un problema, y así, se tenga una guía para poder concretarlo de la mejor manera.

## Bibliografía

- Software Quality Exp. (2016, April 18). 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 inprocess measurement. (n.d.). Chillarege.com. Retrieved September 3, 2023, from <https://www.chillarege.com/articles/odc-concept.html>
- Wikipedia contributors. (2023, August 2). Personal software process. Wikipedia, The Free Encyclopedia. [https://en.wikipedia.org/w/index.php?title=Personal\\_software\\_process&oldid=1168441998](https://en.wikipedia.org/w/index.php?title=Personal_software_process&oldid=1168441998)
- Amon. (2022, September 16). Consejos para minimizar errores en proyectos de desarrollo de software. Cisin. <https://www.cisin.com/coffee-break/es/technology/tips-for-minimizing-bugs-in-software-development-projects.html>