

División de Tecnologías para la integración Ciber-humana
Departamento de Ciencias Computacionales



Tema: Principios de prevención

Mercado Hernandez Angel Gabriel

Computación tolerante a fallas. Sección D06. Lunes y Miércoles 11:00
AM – 12:55 AM

Lopez Franco Michel Emanuel

Introducción:

La prevención de defectos en el desarrollo de software constituye un pilar fundamental para garantizar la calidad y eficiencia de los productos tecnológicos. En un entorno donde la rápida evolución y la creciente complejidad de los sistemas demandan soluciones sólidas, minimizar la ocurrencia de defectos desde las etapas iniciales del ciclo de vida del desarrollo se vuelve esencial. Este principio se sustenta en la premisa de que la detección y corrección de defectos en etapas avanzadas del proceso resulta no solo costosa en términos financieros, sino que también conlleva riesgos significativos en términos de tiempo y calidad del producto final.

La adopción de estrategias efectivas de prevención de defectos implica un compromiso integral de las organizaciones de desarrollo de software. Este enfoque no solo se limita a la identificación de defectos durante la etapa de desarrollo, sino que abarca la cultura organizacional, instando a los equipos a analizar y actuar directamente sobre los datos de defectos. Al asumir la responsabilidad de la calidad del producto, las organizaciones se desafían a sí mismas y a sus equipos de desarrollo, incentivando la toma de decisiones orientadas a evitar la ocurrencia de defectos desde su origen.

En este contexto, el artículo que se presenta destaca la importancia de la prevención de defectos, centrándose en métodos específicos para abordar este desafío. Entre estos métodos, la Clasificación Ortogonal de Defectos (ODC) emerge como una técnica prevalente, agrupando los defectos en tipos en lugar de medirlos de manera independiente. La investigación se enfoca en analizar la cantidad total de defectos en proyectos similares, identificando las causas raíz y proponiendo ideas preventivas basadas en el aprendizaje de proyectos anteriores.

Así, la efectividad de la prevención de defectos se convierte en un elemento crucial para la optimización de costos y tiempos en el desarrollo de software, delineando un camino hacia la excelencia en la producción de productos tecnológicos de alta calidad. La interconexión entre el análisis de defectos, la prevención y la mejora continua del proceso se erige como un componente esencial para el logro de un desarrollo de software eficiente y resiliente ante los desafíos cambiantes del entorno tecnológico.

Desarrollo:

El artículo aborda la prevención de defectos en el desarrollo de software. Destaca el método principal, la Clasificación Ortogonal de Defectos (ODC), que agrupa los defectos en tipos en lugar de medirlos de manera independiente. Se enfatiza la importancia de analizar las causas raíz de los defectos y utilizar las lecciones aprendidas en proyectos para medidas preventivas. Se propone un enfoque triple en la prevención de defectos: registrar y revisar defectos, emplear métodos de diseño efectivos y reducir el tiempo de codificación mediante diseños exhaustivos. El artículo también introduce un ciclo de prevención de defectos que implica

reuniones de análisis de causas y reuniones de equipos de acción. Varias empresas, incluyendo IBM y HP, tienen sus propios enfoques de clasificación de defectos.

El artículo menciona varios métodos para la prevención de defectos en el desarrollo de software:

1. Registro y Revisión de Defectos:

- Ingenieros registran y revisan datos de cada defecto encontrado y corregido.
- Se analizan para determinar las causas y realizar cambios en el proceso para eliminarlas.
- Medir los defectos hace que los ingenieros sean más conscientes de sus errores y sensibles a las consecuencias.

2. Método de Diseño Efectivo:

- Utilizar un método y notación de diseño efectivos para producir diseños completos.
- Un diseño completo requiere una comprensión exhaustiva, lo que resulta en menos errores de diseño.

3. Reducción del Tiempo de Codificación:

- Un diseño más detallado reduce el tiempo de codificación, disminuyendo así la introducción de defectos.
- Ingenieros que producen diseños exhaustivos tienden a inyectar menos defectos durante la codificación.

4. Análisis de Causas y Acciones Preventivas:

- Realizar reuniones de análisis de causas y equipos de acción para examinar defectos y determinar las causas raíz.
- Implementar acciones preventivas basadas en estas reuniones en el proceso de desarrollo.

5. Clasificación Ortogonal de Defectos (ODC):

- Agrupar defectos en tipos en lugar de medirlos de manera independiente.
- Destacar áreas en el ciclo de vida del desarrollo de software que requieren atención.

Cabe destacar que estos métodos buscan abordar la prevención de defectos desde diferentes perspectivas, incluyendo diseño, registro y análisis de causas.

Conclusión:

En conclusión, la prevención de defectos en el desarrollo de software se revela como un imperativo estratégico para las organizaciones que buscan la excelencia en la producción de productos tecnológicos. La adopción de métodos específicos, como la Clasificación Ortogonal de Defectos (ODC), resalta la importancia de abordar los defectos desde sus raíces, agrupándolos en tipos para una comprensión más profunda y una intervención preventiva más efectiva.

La investigación presentada, centrada en proyectos similares desarrollados por estudiantes, no solo destaca la necesidad de analizar la cantidad total de defectos, sino también de identificar las causas subyacentes. La conexión entre el análisis de defectos, la toma de medidas preventivas y la aplicación de lecciones aprendidas en nuevos proyectos demuestra ser un ciclo virtuoso para la mejora continua.

La cultura de prevención de defectos, que involucra a los equipos de desarrollo desde las etapas iniciales del ciclo de vida, no solo resulta en una mayor eficiencia económica y temporal, sino que también contribuye a la construcción de una mentalidad organizacional centrada en la calidad. La incorporación de reuniones de análisis de causas, acciones preventivas y la consolidación de estas prácticas como parte integral del proceso de desarrollo establece un marco sólido para el aprendizaje y la evolución constante.

En última instancia, la prevención de defectos no solo se traduce en la minimización de errores y costos asociados, sino que también promueve un enfoque proactivo hacia la calidad del software. Esta perspectiva preventiva se erige como un catalizador esencial para el logro de proyectos exitosos, al tiempo que sienta las bases para la madurez organizacional y la capacidad de adaptarse a los desafíos dinámicos del desarrollo de software en un entorno tecnológico en constante evolución.

Bibliografías:

- Carrizo, D., & Alfaro, A. (2018). Método de aseguramiento de la calidad en una metodología de desarrollo de software: un enfoque práctico. *Ingeniare. Revista Chilena de Ingeniería*, 26(1), 114–129.
<https://doi.org/10.4067/s0718-33052018000100114>
- De código cuyo objetivo principal es detectar e identificar, I. de C. E. U. T. F. de R. (s/f). Clasificación Ortogonal de Defectos Atributo Valores. *Www.uv.mx*. Recuperado el 4 de febrero de 2024, de <https://www.uv.mx/personal/ermeneses/files/2017/08/Clasificacion-Ortogonal-de-Defectos.pdf>