

CENTRO UNIVERSITARIO DE CIENCIAS EXACTAS E INGENIERÍAS



COMPUTACIÓN TOLERANTE A FALLAS
I7036
SECCIÓN D06

Principios de prevención de defectos

BAUTISTA MARTINEZ EDEN ABDIEL
CÓDIGO: 221350524

PROFESOR: MICHEL EMANUEL LOPEZ FRANCO

Existen varios métodos para la prevención de defectos en el desarrollo de software:

1. **Análisis de requisitos de software:** Un análisis en los requisitos de software es esencial para prevenir defectos. Esto implica una comprensión clara de las necesidades del cliente y la creación de especificaciones detalladas. La colaboración estrecha con los stakeholders ayuda a reducir malentendidos y errores en esta etapa crucial.
2. **Revisión e Inspección:** Las revisiones y las inspecciones son fundamentales en la detección temprana de posibles defectos. Las revisiones de diseño, la verificación de programas y la inspección de código permiten a los equipos identificar y corregir problemas antes de que se arraiguen en el desarrollo, mejorando así la calidad general del software.
3. **Registro y documentación de defectos:** Mantener un registro detallado de los defectos encontrados proporciona información valiosa para futuras mejoras. El análisis de estos registros ayuda a comprender patrones recurrentes, permitiendo la implementación de medidas preventivas más efectivas.
4. **Análisis de defectos detectados:** Analizar cada defecto detectado durante las pruebas y en producción es esencial. Identificar las causas subyacentes permite no solo corregir el problema inmediato, sino también implementar soluciones preventivas para evitar la repetición de errores similares en el futuro.
5. **Prácticas de codificación seguras:** Adoptar prácticas de codificación seguras desde el principio es crucial. Un enfoque de seguridad por diseño implica considerar la seguridad en cada fase del desarrollo, reduciendo la posibilidad de introducir defectos relacionados con vulnerabilidades y errores de codificación.
6. **Gestión de cambios, librerías, control de versiones:** Organizar eficientemente la gestión de cambios, el uso de librerías y el control de versiones ayuda a mantener la integridad del software. Evita conflictos, garantiza la consistencia y facilita la identificación rápida y corrección de posibles defectos derivados de cambios no controlados.

7. **Pruebas del sistema (“system testing”)**: Las pruebas del sistema son cruciales antes de la entrega del software. Identificar y corregir defectos en esta etapa asegura que el producto final cumpla con los requisitos y funcione de manera confiable en diversos entornos, minimizando la posibilidad de problemas post-implementación.

Mtp. (2023, February 22). *Prevención de Defectos en el aseguramiento de la calidad software*. MTP.

<https://www.mtp.es/blog/testing-software/prevencion-de-defectos-en-el-aseguramiento-de-la-calidad-del-software/>

Fusari, G., & Fusari, G. (2023, March 1). *Calidad del software: metodologías y tecnologías para garantizarla*. InnovaciónDigital360.

<https://www.innovaciondigital360.com/software/calidad-del-software-metodologias-y-tecnologias-para-garantizarla/>

Parra, J. (2023, October 28). *Prevén eficazmente las fallas de software: causas y soluciones*. Amisando.

<https://amisando.es/preven-eficazmente-las-fallas-de-software-causas-y-soluciones/>