



UNIVERSIDAD CARLOS III DE MADRID

ESCUELA POLITÉCNICA SUPERIOR

AÑO ACADÉMICO (2023 – 2024)

SEGUNDO CURSO

ARTÍCULO DE REFACTORING

**UTILIZACIÓN DE ANTIPATRONES Y PATRONES EN EL
ANÁLISIS DE SOFTWARE**

Grupo 84

Apellidos y Nombres:

Rivera Amor, Gabriel José

Docente:

Carrero de las Peñas, María Natividad

Madrid 26 de abril de 2023

RESUMEN

El contenido que está a punto de leer describe un estudio sobre la implementación de patrones al validar modelos conceptuales. Este trabajo fue realizado por el grupo GIDTSI de la Universidad Tecnológica Nacional, Facultad Regional de Córdoba. El objetivo de este proyecto es mejorar la obtención y especificación de requisitos de software a través de un enfoque particular: la validación de modelos conceptuales.

La metodología propuesta aprovecha BPMN para modelar procesos de negocio, lo que a su vez ayuda a derivar casos de uso del sistema de información y al mismo tiempo garantiza la trazabilidad entre ambos tipos de modelos. Además, nuestra exploración profundiza en el ámbito de la "comprobación de modelos" como una vía para validar modelos de casos de uso.

Se describe el proceso de desarrollo de software con enfoque en las etapas de requisitos, análisis, diseño, implementación y prueba, describiendo también que el trabajo del grupo GIDTSI se centra en la fase de requisitos.

Se presentan la idea de antipatrones (soluciones recurrentes defectuosas y patrones) exitosos. Se dividen en diferentes clasificaciones, como desarrollo de software, arquitectura de software y gestión de proyectos de software.

Se elaboran varios tipos de patrones de software, incluidos patrones de análisis, patrones arquitectónicos, patrones de diseño y expresiones idiomáticas, con ejemplos proporcionados para una mejor comprensión. Además, se profundiza en la relación entre estas dos entidades (patrones y antipatrones), discutiendo cómo evitar errores comunes y adoptar estrategias exitosas.

Por lo tanto, la adopción de antipatrones junto con patrones de análisis no es solo un esfuerzo para evitar trampas: también apunta a elevar la calidad del software desarrollado a través de una mejora cultural en los procesos de comunicación y documentación que buscan evitar que esos errores vuelvan a ocurrir en el futuro.

CONCLUSION

Elegí este artículo porque aborda un tema que me resulta fascinante: la aplicación de patrones y antipatrones en el desarrollo de software. Esto me permite tener una idea más clara de los procesos que hemos llevado a cabo durante esta actividad.

Además, me interesa cómo el artículo explora diferentes aspectos relacionados con la validación de modelos conceptuales y la optimización de la definición inicial de un sistema. Esta perspectiva de integrar patrones y antipatrones en el modelado conceptual para facilitar la verificación y validación de los requerimientos funcionales es innovadora y puede tener un impacto significativo en la eficiencia y calidad del desarrollo de software.

También encuentro valiosa la discusión sobre las diferentes categorías de antipatrones y patrones, así como su relación y evolución. Esta comprensión más profunda de los errores recurrentes y las soluciones exitosas puede proporcionar una base sólida para mejorar continuamente los procesos de desarrollo de software y la toma de decisiones.

En resumen, elegí este artículo porque creo que ofrece una visión integral y práctica sobre cómo aplicar patrones y antipatrones, relacionados a la refactorización de los elementos de un programa, siendo además de gran ayuda para comprender aún más los procesos que estamos llevando a cabo con esta actividad.

BIBLIOGRAFÍA

- Garnero, A. B., & Horenstein, N. Utilización de antipatrones y patrones en el análisis de software.