

Improving Legacy Code – Nicolas Carlo

Lucas Gisbert López - UO295526

Pablo Pérez Saavedra - UO288816

Introducción al Legacy Code

- **Distintas definiciones del Legacy Code**
- **Aclaraciones sobre sus características**

El problema sobre el enfoque actual incorrecto

- **Prioridad a nuevas funcionalidades**
- **Percepción negativa del legacy code**
- **Propuesta de cambio de paradigma**

Formas de actuar ante el Legacy Code

- **Enfoques útiles que seguir**
 - Enfoque de tratar el código como una escena del crimen
 - Arqueología del código
- **Buenas prácticas al modificar legacy code:**
 - Tests de comportamiento esperado
 - Mikado Method

Tipos de cambios: Refactor vs Rewrite

- **Refactor (Refactorización)**
- **Rewrite (Reescritura)**

Ayudas e inconvenientes de la IA en el tratamiento de Legacy Code

La inteligencia artificial (IA) es una herramienta poderosa en el mantenimiento de código heredado (*legacy code*), destacando en:

- **Refactorización:** Ayuda a reorganizar y mejorar la estructura del código sin alterar su funcionalidad.
- **Sugerencia de test cases:** Genera pruebas automatizadas para comprender mejor el código y aumentar su cobertura.

Sin embargo, presenta inconvenientes:

- **Limitaciones en la generación de pruebas:** En códigos complejos, los test cases pueden no ser lo suficientemente flexibles.
- **Riesgos en la refactorización automática:** Puede introducir errores o regresiones, siendo más segura con una suite de pruebas sólida.

Una solución prometedora: IA + refactorización automatizada

Combinar IA con herramientas automatizadas permitiría sugerencias de refactorización aplicadas de forma segura por el editor, garantizando cambios confiables.

Herramientas y técnicas clave

Herramientas:

- **Code Scene:** Analiza el comportamiento del código e identifica problemas.
- **Multiple Development:** Enfoque colaborativo, con *Glamorous Toolkit* como herramienta destacada.

Técnicas:

- **Naming:** estos evolucionan con el código.
- **Método Mikado:** Divide tareas en subtareas pequeñas con *micro committing*.
- **Approval Testing:** Técnica rápida para cubrir código con pruebas automatizadas.