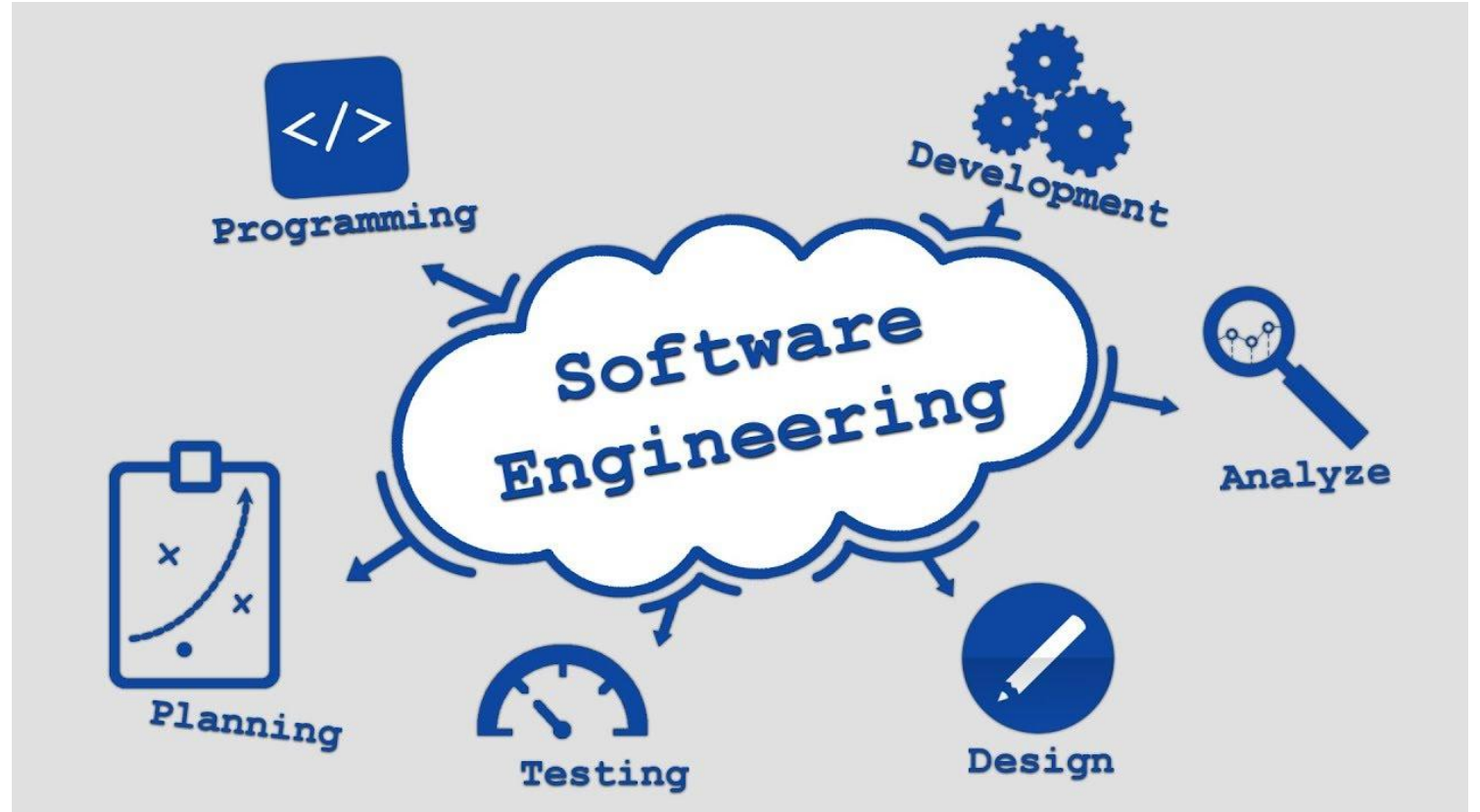


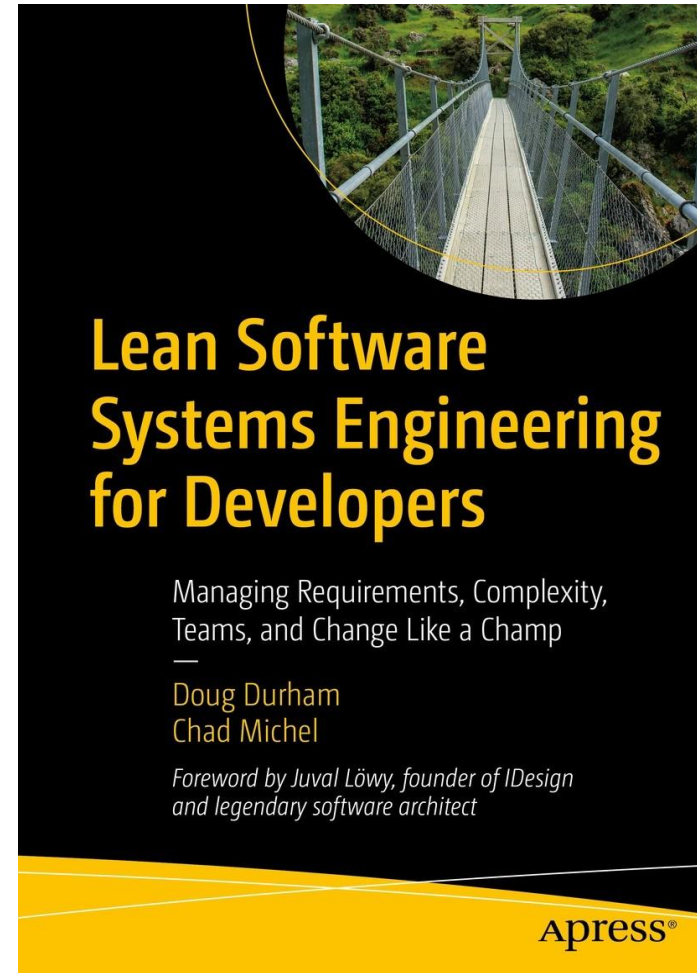
Software as an Engineering Discipline

Zinedine Álvarez Sais
Darío Cristóbal González
Daniel Alonso Fernández



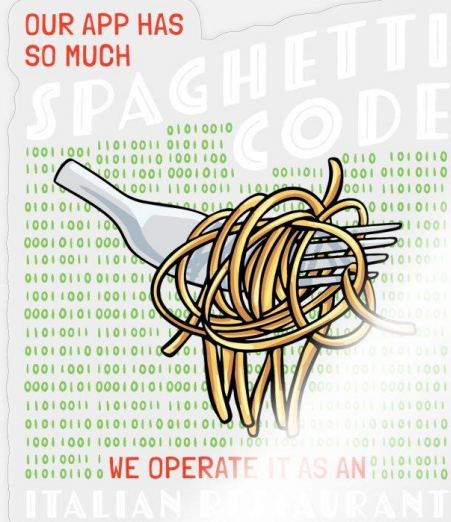
[SE Radio 574: Chad Michel on Software as an Engineering Discipline](#)

¿Quién es Chad Michel?



Desarrollo Software vs Ingeniería Software

```
public class HelloWorld {  
  
    public static void main(String[] args) {  
        System.out.println("Hello world!");  
    }  
}
```



OUR APP HAS
SO MUCH
SPAGHETTI
CODE
WE OPERATE IT AS AN
ITALIAN RESTAURANT

Planificación antes de escribir código

Definir arquitecturas robustas

Asegurar que el software pueda escalar con facilidad.

Aplicar metodologías de calidad y pruebas.

Rigurosidad en la Ingeniería Software

Código difícil de mantener

Sistemas frágiles ante cambios

Problemas de escalabilidad

Costos elevados de corrección y mantenimiento

Gestión de requisitos bien definida

Diseño arquitectónico antes de programar

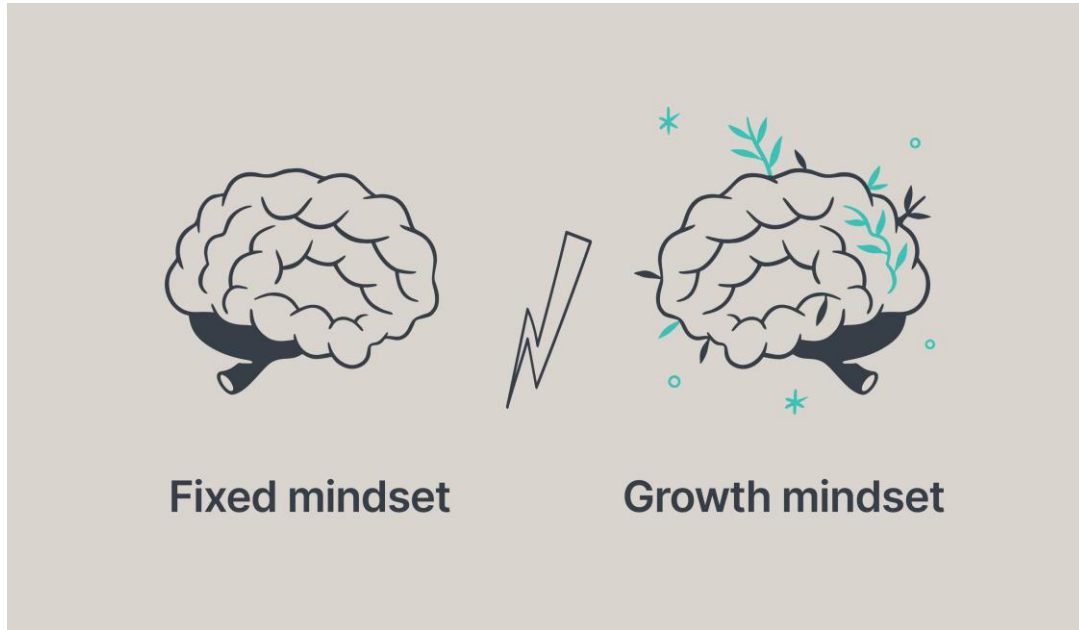
Uso de estándares y marcos de referencia

Pruebas continuas para garantizar calidad:



*Guide to the Software
Engineering Body of Knowledge*

Resistencia al Cambio

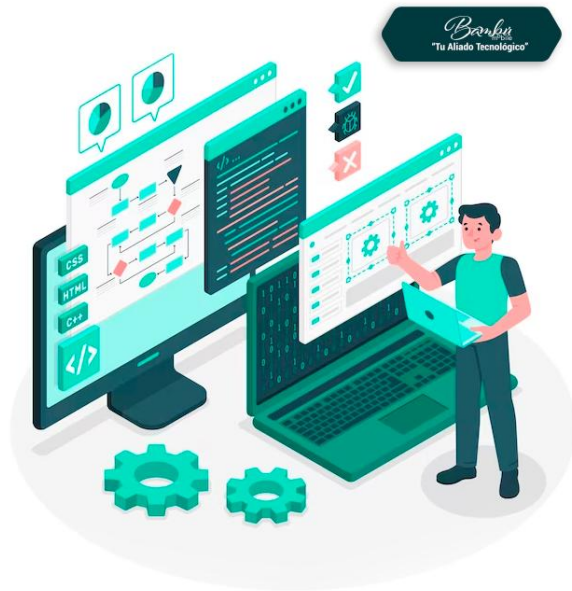


DISEÑAR PARA EL CAMBIO



DEGRADACIÓN DEL SOFTWARE

Diseño Inicial Válido



Paso del Tiempo



Añadir Funcionalidades



METODOLOGÍA DE DESARROLLO

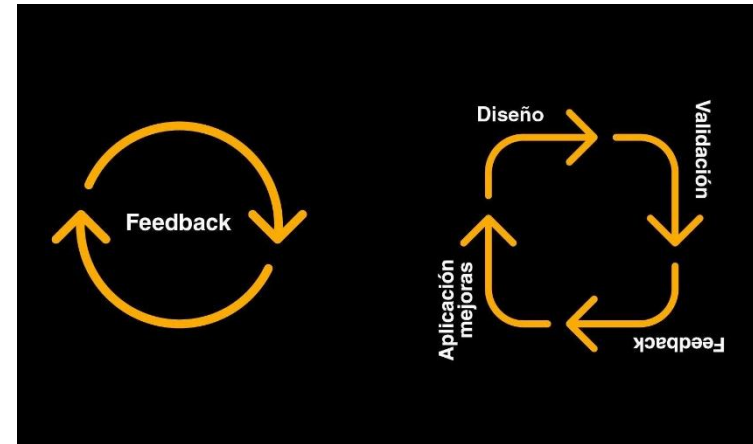
DEFINICIÓN DE REQUISITOS



PERIODO DE TIEMPO RAZONABLE



Proceso Iterativo e Incremental



Si ese cambio ocurriera, ¿cómo afectaría esto al diseño del sistema?



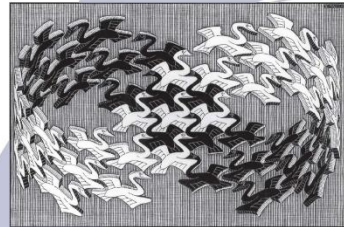
ATACAR DISEÑO

PATRONES DE DISEÑO

Design Patterns

Elements of Reusable
Object-Oriented Software

Erich Gamma
Richard Helm
Ralph Johnson
John Vlissides



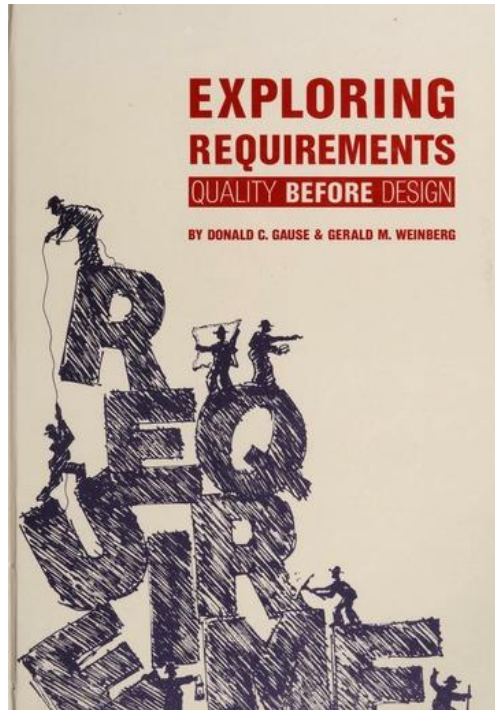
Cover art © 1994 M.C. Escher / Gordon Art - Baarn - Holland. All rights reserved.

Foreword by Grady Booch

ADDISON-WESLEY PROFESSIONAL COMPUTING SERIES

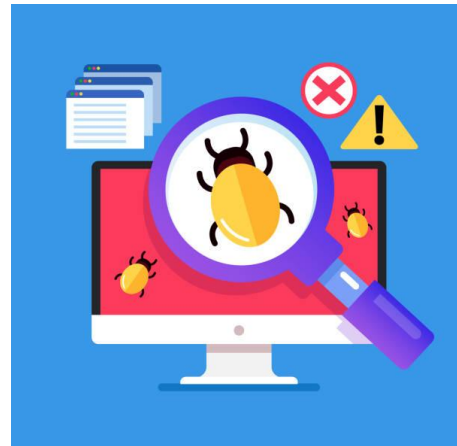


ERRORES DURANTE EL DESAROLLO

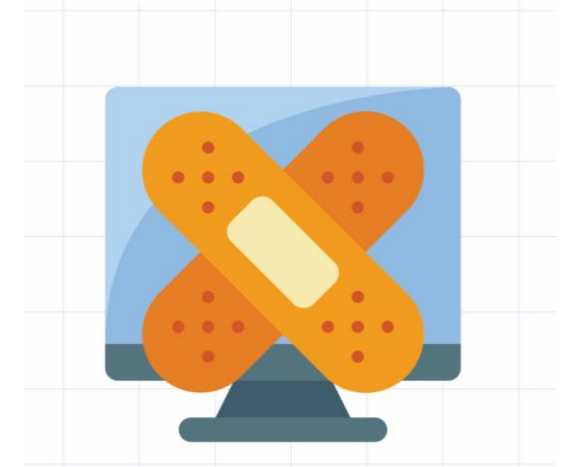


”El lugar más barato para encontrar un defecto es en los requisitos.”- Gerald Weinberg en su libro Exploring Requirements

- Los errores en los requisitos y diseño son responsables de más del 50% de los defectos.
- Encontrar un error en producción es entre mil y diez mil veces más caro que detectarlo en la etapa de requisitos.

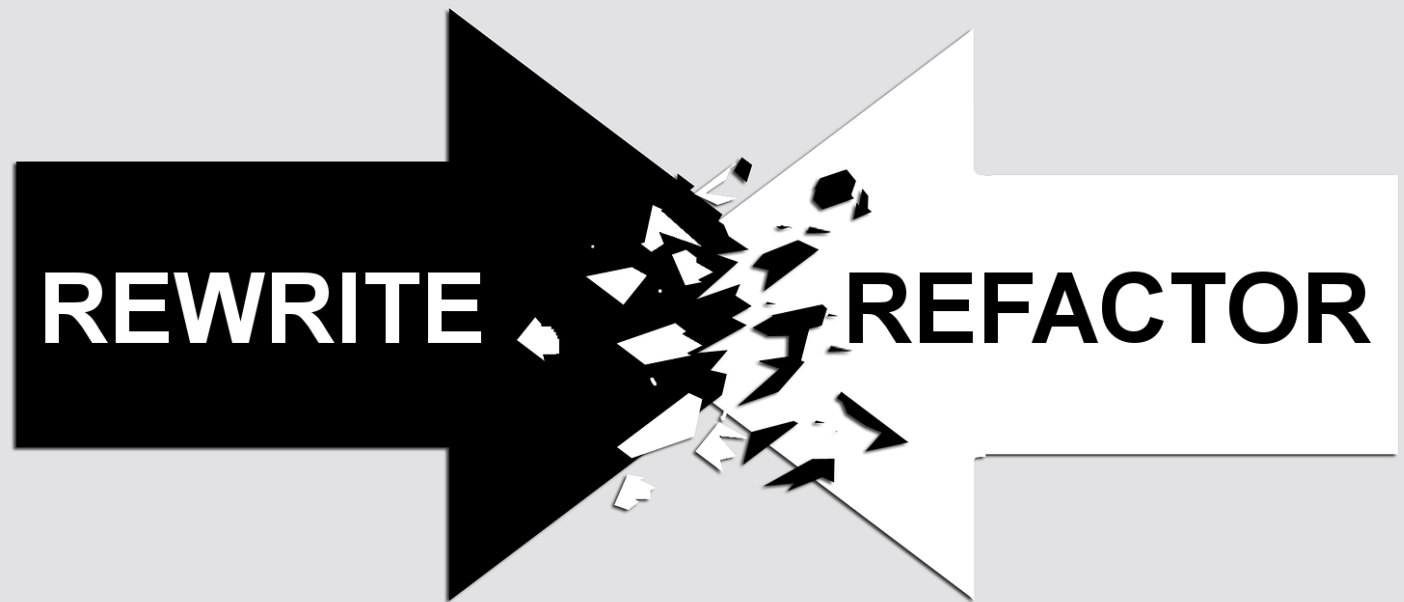


Bugs

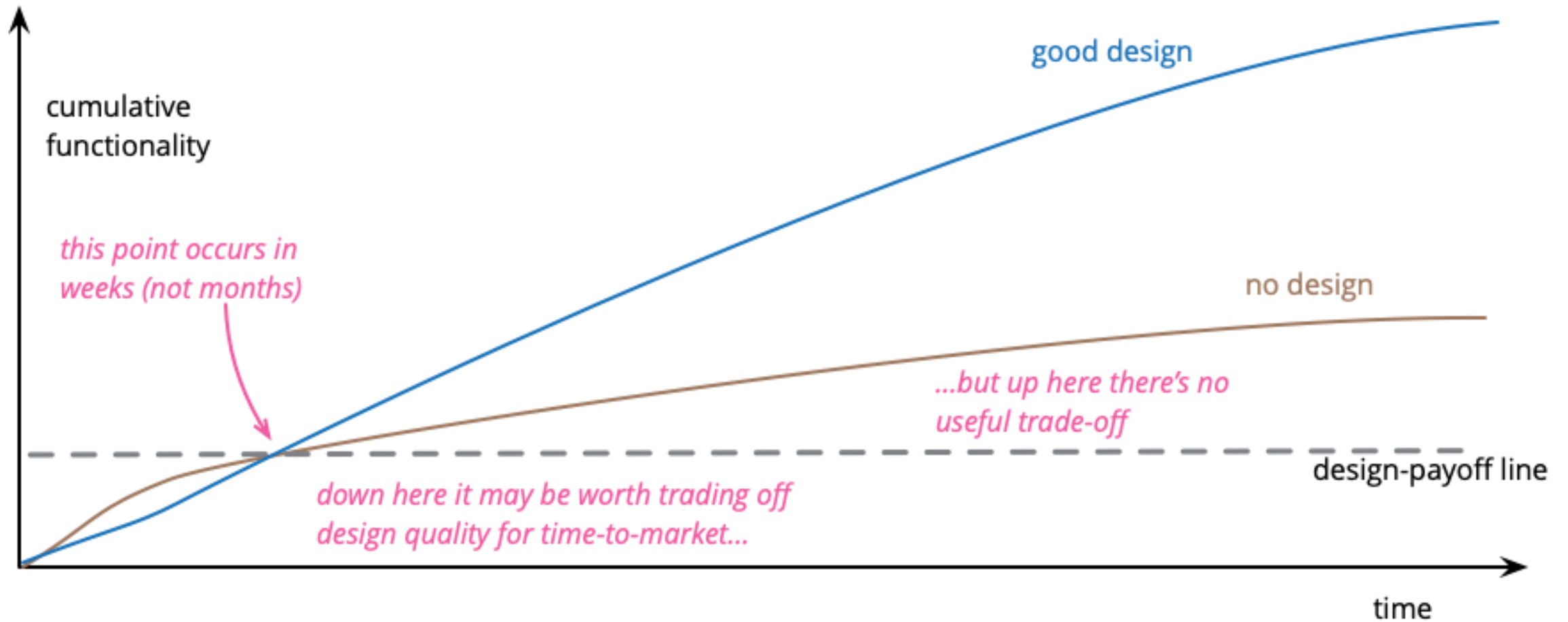


Hotfix

**Refactor
vs
Rework**



Hipótesis de resistencia del diseño



TEMAS



Aseguramiento de calidad vs control de calidad

Control de calidad

Rol de “Ingeniero Jefe” y preparación de equipos

Evolución de la Ingeniería del Software

Aseguramiento de calidad vs control de calidad

Aseguramiento de calidad (QA) 	Control de Calidad (QC) 
Prevención de defectos	Detección de defectos
Se aplica desde el inicio del desarrollo	Se aplica cuando el producto ya está construido
Ejemplo: Revisiones de código, auditorías	Ejemplo: Pruebas unitarias, <i>crash testing</i>

Control de calidad

- Cordón de Andon
- Responsabilidad compartida



Rol de “Ingeniero Jefe” y preparación de equipos

- Visión global del proyecto
- Formación de nuevos integrantes
- Capacidad de adaptarse a los cambios

Evolución de la Ingeniería del Software

- Expectativas de los usuarios
- Desarrollo más complejo
- Líderes que promuevan las mejores practicas
- Colaboración dentro de la comunidad de desarrolladores