




Deuda Técnica

Arquitectura del Software

- Israel Solís Iglesias - UO282162
 - Raúl Fernández España - UO278036
 - David Warzynski Abril - UO278968
- 

Deuda Técnica

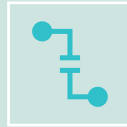
- "La deuda técnica es el costo del trabajo adicional causado por la elección de la solución más rápida en lugar de la más efectiva".
- No siempre es negativo.
- Deberíamos tenerla en cuenta.
- Consecuencias a largo plazo.



Causas de la deuda técnica



Errores conocidos y no solucionados.



Mejoras en el código no implementadas.



Insuficientes pruebas o pruebas de mala calidad.



Documentación desactualizada, incompleta o inexistente.

Reconocer deuda técnica

- Es inevitable
- Requiere conocimiento del software y de los objetivos del negocio.
- Empiezas a observar consecuencias negativas.
- Problemas integrando nuevos desarrolladores



Tipos de deuda técnica

	DEUDA IMPRUDENTE	DEUDA PRUDENTE
DEUDA DELIBERADA	Nos falta de tiempo para diseñar	Tenemos que priorizar la entrega de software y lidiar con las consecuencias
DEUDA INVOLUNTARIA	¿Qué es un diseño por capas?	Ahora sabemos cómo deberíamos haberlo hecho

Afrontar la deuda técnica



Todo sistema tiene deuda técnica.



Es necesario manejarla.



Decisión -> beneficio.

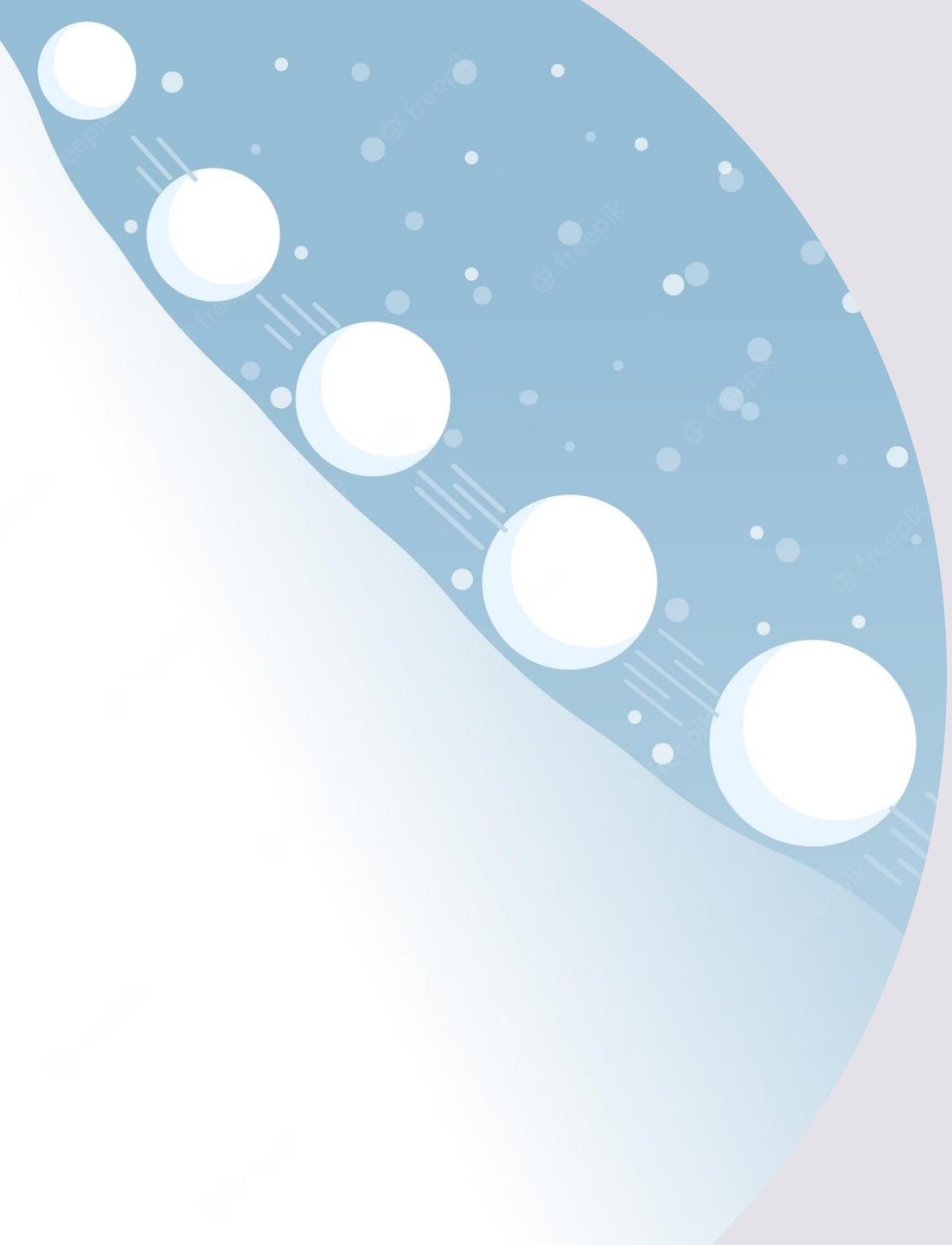


Intervalos regulares.


Gestionar deuda técnica

- No se puede eliminar la deuda técnica al completo.
- Si no se gestiona se va a acumular.
- Consejos:
 - Identificar la deuda técnica
 - Priorizar las tareas
 - Planificar el trabajo
 - Adoptar prácticas de desarrollo sostenible





¿La deuda técnica es mala?

- No es necesariamente mala.
 - Decisiones que aportan valor a corto plazo.
 - Hay que atenderla, ya que se acumula a lo largo del tiempo.
 - Al igual que la deuda financiera, la deuda técnica puede utilizarse tanto de forma positiva como negativa.
- 

Consecuencias deuda técnica

Errores en el código

Ausencia o deficiente control de versiones.

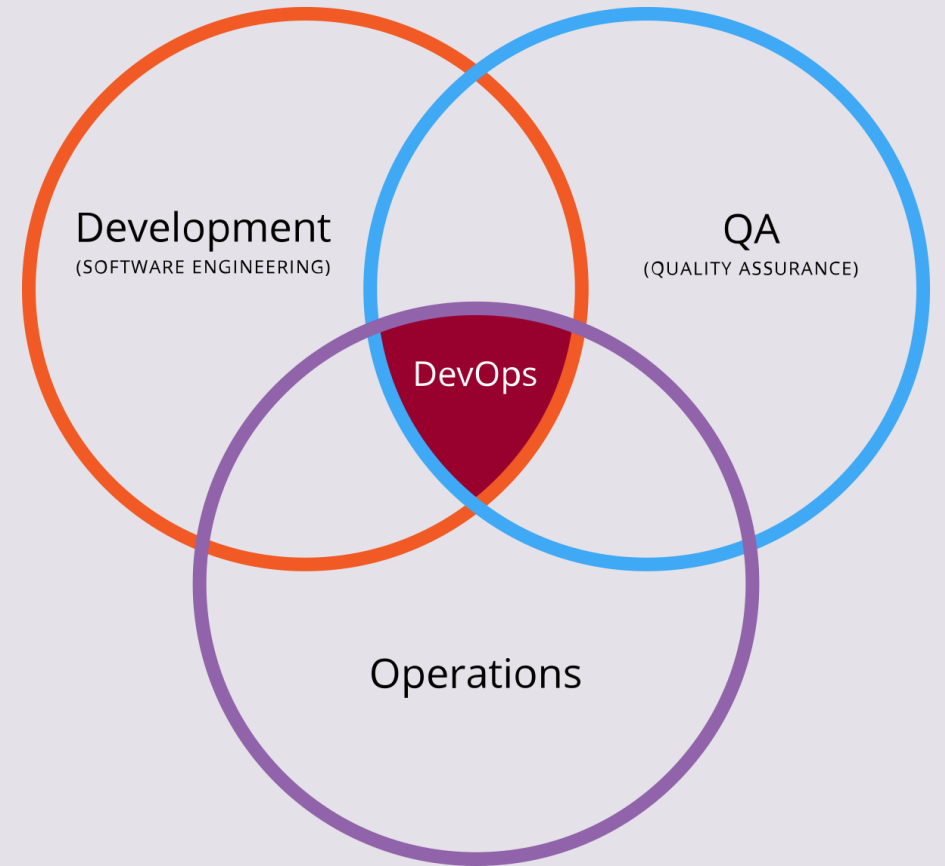
Arquitectura no escalable.

Rigidez para actualizar a nuevas tecnologías o plataformas.

DevOps

Unir development y operations.

- Codificar: Desarrollo y revisión de código
- Construir: Control de versiones
- Probar
- Empaquetar: Gestión de artefactos
- Release: automatización de versiones
- Configurar y gestionar
- Monitorizar: Rendimiento, experiencia del usuario





Conclusiones

- Siempre hay que tomar una decisión.
- No hay una decisión ideal.
- Es importante monitorear constantemente nuestro sistema.
- Hay que tener claro el objetivo de nuestro sistema.



Preguntas
