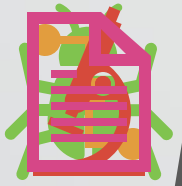
Technical debt

Pablo Fernández Díaz UO270157

Ignacio Gómez Gasch UO271548

Que es Technical debt

"La **deuda técnica** es un concepto que se usa en desarrollo para definir el coste de mantener y arreglar un software mal construido, a menudo por hacerlo rápido o por no haber llevado a cabo un buen control de calidad antes de lanzarlo."



Ejemplos de Technical debt

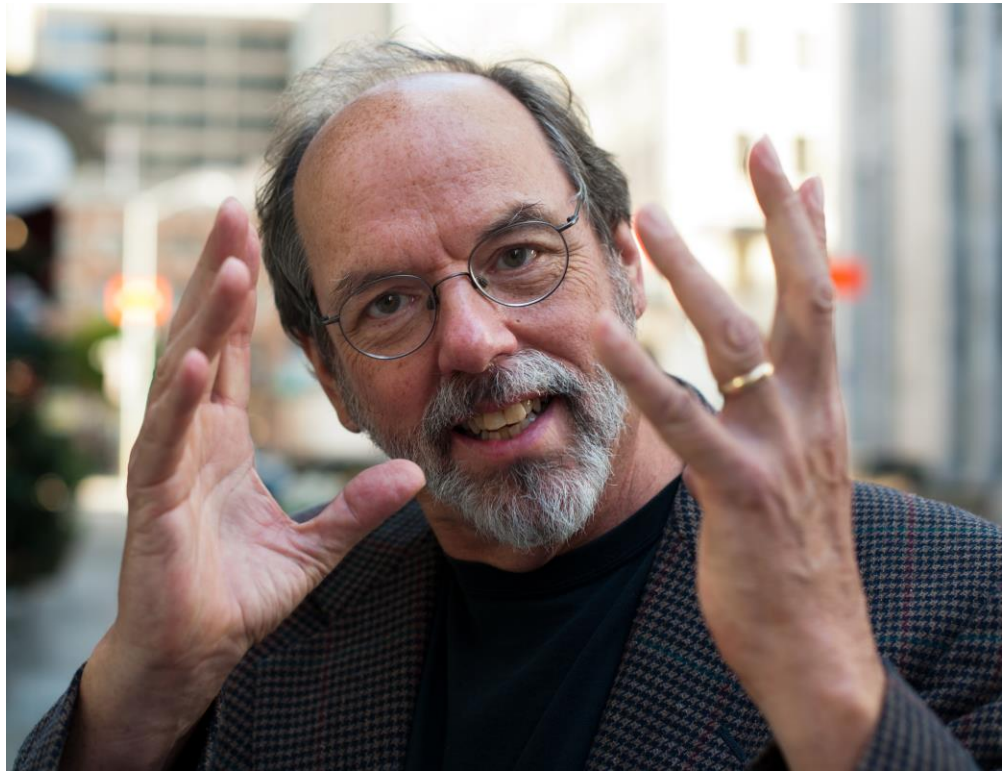
- Errores conocidos y no solucionados.
- Mejoras en el código no implementadas.
- Insuficientes pruebas o pruebas de mala calidad.
- Documentación desactualizada, incompleta o inexistente.

Consecuencias de la Technical debt

- Errores en el código
- Ausencia o deficiente control de versiones.
- Arquitectura no escalable.
- Rigidez para actualizar a nuevas tecnologías o plataformas.

Historia

- Ward Cunningham
  - Creador de la primera Wiki
  - En 1992 introduce el concepto de Technical debt
  - Metáfora con el crédito financiero



Idea Original

- Uso Iterativo en lugar de Cascada
  - Iterativo -> Lista completa de requisitos en la última iteración.
  - Cascada -> Lista completa de requisitos al inicio
- Idea de refactorización no generalizada
- Divergencia entre ideas y código

Tipos más frecuentes

Technical debt planeada

- Ocasionados tras una implementación inicial rápida, debido a la urgente necesidad en el mercado de este producto.
- El mercado cambia rápidamente, lo que requiere centrar todos los esfuerzos en realizar cambios sobre el diseño original y no en rediseñar este.

Technical debt inadvertida

- Originada por desarrolladores sin experiencia o por malas gestiones a la hora del desarrollo

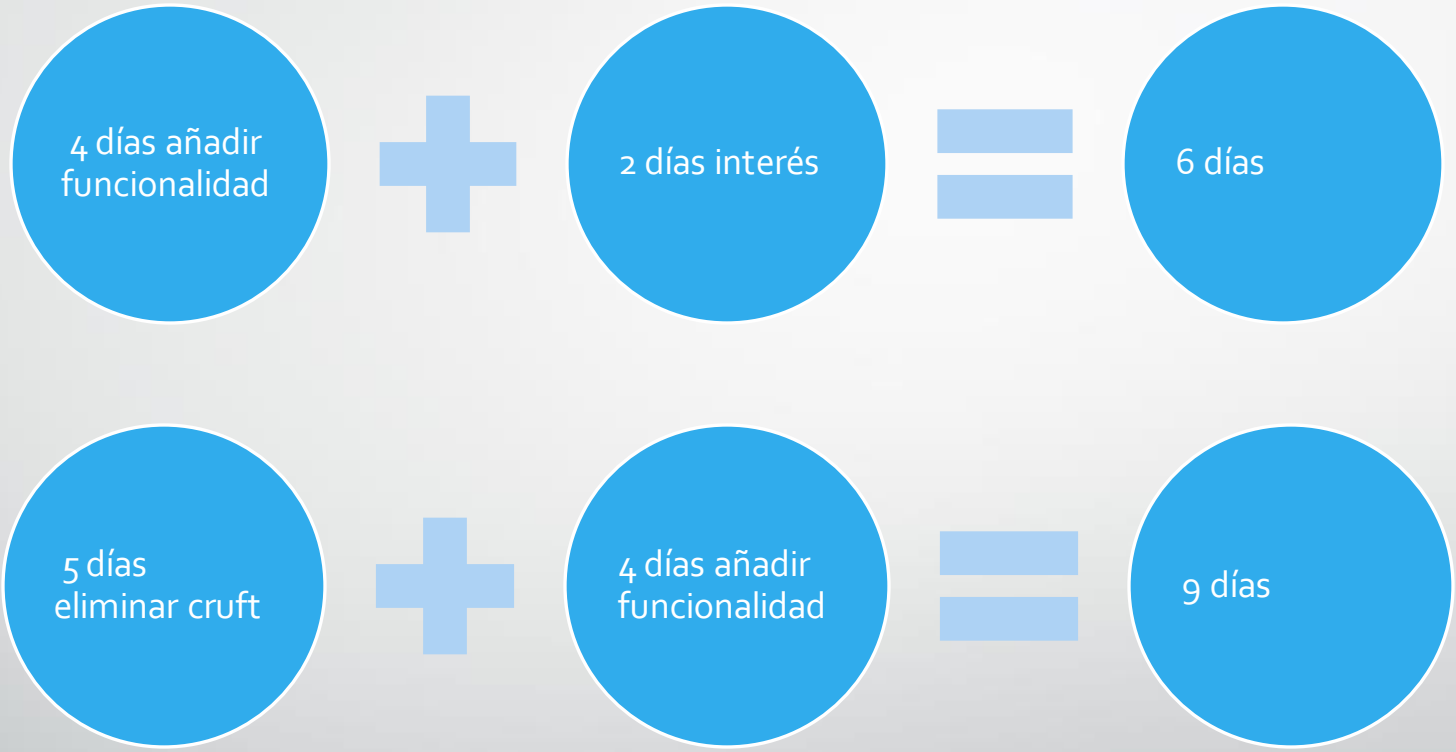
Elementos technical debt

- Cruft: deficiencias en la calidad interna que hacen que sea más difícil de lo que sería idealmente modificar y ampliar aún más el sistema.
- Interés: es el esfuerzo adicional que se necesita para agregar nuevas funciones.

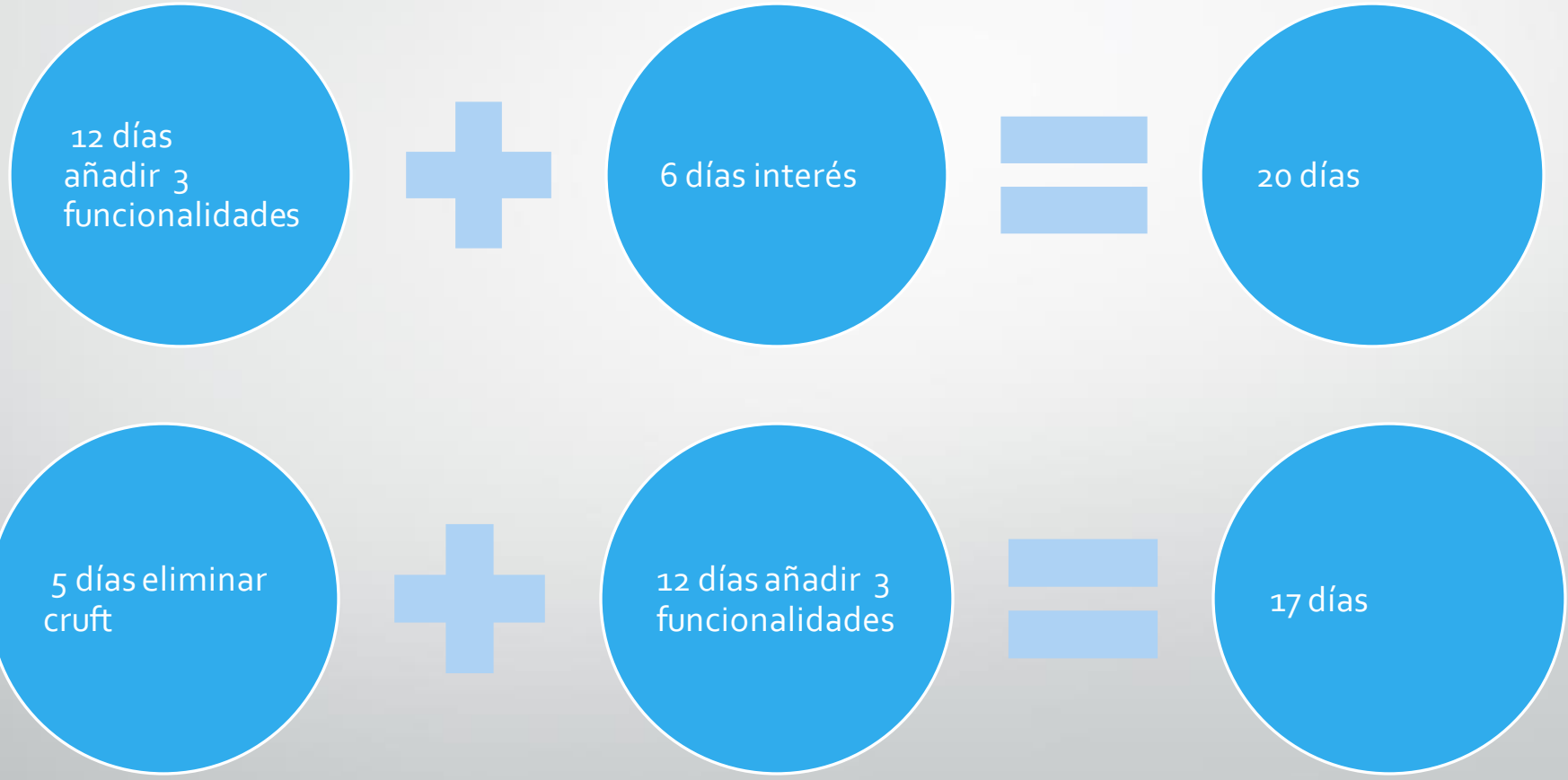
Ejemplo práctico

Es necesario agregar una nueva funcionalidad en nuestro código. Si el código principal estuviese bien estructurado, me tomaría cuatro días agregar la funcionalidad, pero con este cruft, me tomaría seis días. La diferencia de dos días es el interés de la deuda.

Ejemplo práctico: alternativas



Ejemplo práctico: alternativas



Prevención

- Testeo automático
- Establecimiento y estandarización de códigos
- Uso de herramientas de gestión de proyectos
- Uso de rastreadores de problemas
- Pruebas de sistema y descubrimiento