

Technical debt

Pablo Fernández Díaz **UO270157**

Ignacio Gómez Gasch **UO271548**

¿Qué es la Technical debt?

El termino de "Technical debt" hace referencia al coste de mantener y arreglar todos aquellos problemas que surgen de tener un software mal construido, debido a múltiples factores como pueden ser: realizar este de una forma rápida debido a fechas de entrega, por el desconocimiento de alguno de los integrantes del equipo de desarrollo...

Originalmente, esta idea tenía un significado mucho más sencillo, pero con el paso del tiempo se ha convertido en un término que abarca muchos de los problemas derivados de la toma de algunas decisiones en el desarrollo de un software, teniendo en cuenta que es un aspecto muy importante, al cual se le debe prestar atención cuando en el desarrollo del software.

Ejemplos y Consecuencias de la deuda Técnica

Al ser un concepto que, con el paso del tiempo, se ha ampliado su definición hay una multitud de ejemplos y consecuencias derivadas de la deuda técnica. En cuanto a algunos ejemplos estos pueden ser:

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

En cuanto a las consecuencias, estan van acorde a los ejemplos anteriores. Algunas de estas pueden ser:

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

Historia del termino

El termino de "Technical debt" fue acuñado por primera vez por el programador estadounidense Ward Cunningham en 1992. La idea de este término surge del intento de explicar como el uso de los desarrollos iterativos permiten conseguir un trabajo más rápido, a causa de obtener ciertos problemas en el desarrollo de nuestro proyecto, teniendo en esa "deuda" tendrá que seguir siendo pagada.

Esto surge con la intención de hacer una metáfora con el crédito financiero, que, funcionando de igual manera, nos permite poder iniciar un proyecto, pero es esencial que la deuda siga siendo pagada, ya que si no los intereses relacionados podrían llegar a parar el proyecto.

La idea surge en un ámbito donde la idea de refactorización no era generalizada por lo que utilizo esto para explicarle a su gerente como las construcciones iterativas funcionaban. Además, originalmente solo hacía referencia a su aparición en los desarrollos iterativos en lugar de los desarrollos en cascada.

Tipos

En el día a día nos encontraremos con estos 2 tipos de technical debt: planeada e inadvertida.

La planeada surge por una necesidad en el mercado, la cual hace que las empresas deban ser muy rápidas a la hora de desarrollar un producto que cubra esa necesidad, por tanto, el propio equipo de desarrollo asume errores en el diseño y en el código base por esta urgencia.

El otro caso es la inadvertida, originada por errores cometidos en el diseño o programación de desarrolladores inexpertos.

Prevención

Sabiendo que la technical debt ocasiona muchos inconvenientes en las futuras ampliaciones o mejoras debemos esforzarnos en evitarla, para eso existen muchas opciones para la prevención, todas ellas basadas en el testeo y revisión constante de todas las partes creadas y por crear.

Conclusiones

La technical debt o deuda técnica, es un concepto muy importante presente en los proyectos de arquitectura software. Con el paso de los años dicho término ha cogido aún más peso, haciendo referencia a múltiples aspectos presente en el desarrollo de un proyecto de software. Es de vital importancia, tener en cuenta cuales son las deudas técnicas que se encuentran presentes en nuestros proyectos, ya que, si no intentamos remediar estas en el futuro, podrían suponer problemas en el transcurso del proyecto e incluso que esté no se pudiese llevar a cabo.

Referencias

<https://www.georgefairbanks.com/ieee-software-v37-n4-july-2020-ur-technical-debt>

<https://www.martinfowler.com/bliki/TechnicalDebt.html>

https://es.wikipedia.org/wiki/Deuda_t%C3%A9cnica

https://www.scrummanager.net/bok/index.php?title=Deuda_t%C3%A9cnica