Adrián Estrada González UO276900

María Fernández Rojo UO263595

Gaspar Pisa Eyaralar UO250825

Technical Debt

¿Qué es?

Este término, que fue acuñado por Ward Cunningham en 1992, se refiere a la "deuda de diseño" o "deuda de código" consecuencia de haber escogido una solución a priori más sencilla, pero que con el paso del tiempo genera muchos más problemas que si desde un primer momento se hubiese invertido el tiempo necesario en buscar la solución óptima.

Se trata de una metáfora que hace referencia a una deuda financiera, de modo que el nuevo tiempo y recursos que se deben invertir en arreglar el código, podría entenderse como los intereses que se pagan sobre la deuda que, en este caso, se trataría del código utilizado inicialmente.

Causas

Existen diversas causas por las que se puede llegar a tener o incluso a acumular deuda técnica. A continuación, se explican algunas de las más comunes.

Tiempo: puede deberse a tiempos de entrega muy ajustados, o a la urgencia de implementación de nuevas características.

Recursos: dinero, tiempo, personal, … puede referirse a varios de estos problemas, pero todos relacionados con una necesidad.

Conocimiento: la falta de entendimiento del problema, o de cómo puede atajarse la solución.

Autoridad: necesidad de la aprobación por parte de un superior o del encargado de esa tarea para llevar a cabo la solución óptima.

Deuda técnica acumulada: debido a la anterior deuda técnica que impide que se lleve a cabo una buena solución y obliga a generar una nueva deuda.

Otros: librerías que ya no reciben mantenimiento, nuevas versiones de framework, …

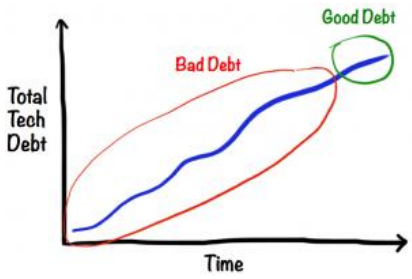
Ventajas y desventajas

A diferencia de lo que a simple vista pueda parecer, la deuda técnica no siempre supone una desventaja, ni utilizarla es algo malo. En realidad, puede ser un recurso muy positivo si se utiliza correctamente. En el punto anterior se explicaban las posibles causas que podrían hacer que nos viésemos en la obligación de desarrollar deuda técnica, muchos de ellos son inevitables, y si no utilizásemos este recurso probablemente habría que paralizar el desarrollo o hasta abandonar el proyecto.

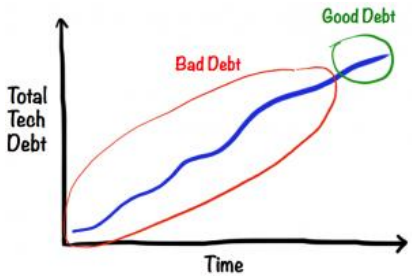
Adrián Estrada González UO276900

María Fernández Rojo UO263595

Gaspar Pisa Eyaralar UO250825



En este sentido se puede ver que la deuda no es mala en sí, siempre y cuando se "pague" lo antes posible. Es decir, la deuda mala es la deuda vieja, existe la posibilidad de que se acumule demasiada deuda y no sea factible continuar con el proyecto. Usando una analogía, es como dejar la cocina desordenada para el día siguiente, vas a perder tiempo teniendo que recoger todo cuando te das cuenta de que no puedes cocinar.



Por otro lado, no deberías estar rebajando la deuda constantemente, a la larga vas a perder tiempo. Cuando estás haciendo una tarea creativa, no es muy práctico estar recogiendo cada vez que usas una cosa, no vas a limpiar un cuchillo después de cada corte. Para aprovechar al máximo hay que encontrar un equilibrio, asume la deuda nueva y no permitas que se acumule, establece un límite a partir del cual vas a gestionarla/rebajarla. Volviendo a la cocina, asume las manchas, pon un límite al desorden y recoge cuando hayas terminado.

Algunos consejos

Planificar correctamente: a veces una buena planificación puede evitar muchas de las situaciones por las que se puede llegar a generar deuda técnica.

Establecer metas alcanzables: ser realistas en cuanto a los objetivos que se pueden alcanzar teniendo en cuenta el tiempo, los recursos, …

Mantener la deuda técnica bajo control: esto quiere decir no acumular demasiada deuda de forma que no sea imposible de sobrellevar a posteriori.

Bibliografía

Ur-Technical Debt, G. Fairbanks

Technical Debt, Martin Fowler

Debt Metaphor, Ward Cunningham

Deuda Técnica, Wikipedia

¿Qué es el Technical Debt?, Gepser Hoil

Technical Debt: The Man, the Metaphor, the Message, NeoPragma

8 consejos para reducir la deuda técnica (sin matar las operaciones), John Edwards

Good and Bad Technical Debt, Henrik Kniberg

The Solution to Technical Debt, Henrik Kniberg