

Resumen de Lean Development

Lean Software Development o simplemente Lean Development es un modelo de desarrollo de software enmarcado dentro de los modelos ágiles como Scrum o Extreme Programming. El término “lean” se puede traducir del inglés como fino o esbelto, ya que se refiere a lo que queda en un proyecto tras eliminar todo lo que no es esencial.

Lean Development es heredero directo de Lean Manufacturing (sistema de producción industrial) y del Sistema de Producción de Toyota y tiene en sus precursores el matrimonio Poppendieck.

El desarrollo de Software Lean se basa en 7 principios fundamentales:

1.- Eliminar los desperdicios

Un desperdicio es todo aquello que no le aporta valor al cliente. Algunos ejemplos pueden ser el código y funciones innecesarias, empezar más trabajo del que se puede terminar, demoras, requisitos poco claros, burocracia, comunicación lenta, mala calidad...

Una forma de controlar estos desperdicios es elaborando un mapa de flujo de valor, cuyo objetivo principal es que nos permita identificar todos aquellos desperdicios que no aportan valor al cliente, así como el tiempo gastado en ellos. De esta forma, trabajando iterativamente, se evitará repetir errores ya cometidos y esta ayudará a identificar nuevos procesos que incluso parecían esenciales, pero que acabarán siendo eliminados como desperdicios.

2.- Amplificar el aprendizaje

El desarrollo de software es entendido como un ejercicio de descubrimiento y aprendizaje. Podemos aprender de la interacción de los usuarios con la interfaz del programa. También podemos acelerar el proceso de aprendizaje mediante iteraciones cortas. La retroalimentación o feedback del cliente es esencial. En estas reuniones, el cliente puede ver en qué fase está el proyecto o consensuar mejoras o nuevas funcionalidades para el futuro. También puede aclarar ambigüedades y aportar información que nos permita decidir que hacer frente a una decisión que habíamos inicialmente aplazado. Además, el cliente consigue comprender qué es lo que realmente necesita y los desarrolladores aprenden a ajustarse a estas necesidades.

3.- Decidir lo más tarde posible

Al ser Lean Development un método ágil, se basa en cortas iteraciones para evitar que se acumulen errores al final del proyecto y que estos sean irreversibles. Del mismo modo, este principio es vital para evitar desperdicios, puesto que los requisitos del cliente pueden variar mucho con respecto al principio del proyecto. Por ello, es importante posponer decisiones que pueden ser cruciales, y siempre trabajar con distintas opciones y sus posibles soluciones, para así lograr un proyecto más flexible que sea capaz de amoldarse a los distintos cambios.

4.- Entregar tan rápido como sea posible

Para conseguir entregar nuestros proyectos lo antes posible, es esencial el uso de iteraciones cortas. Estas mejorarán el aprendizaje y la comunicación dentro del equipo. Además, la entrega rápida imprime una velocidad al proyecto que nos obliga a posponer decisiones importantes por falta de información suficiente (decidir lo más tarde posible). Otro añadido para apostar por la

entrega lo antes posible es que el cliente quiere que se le presente el producto que quiere hoy y no el que quería ayer. Una frase que también representa muy bien el principio de entregar lo más rápido posible es que una solución al 80% hoy es mejor que una al 100% mañana. Esto se explica perfectamente con los cambios de opinión continuos de algunos clientes. Por si fuera poco, los clientes valoran muy positivamente la entrega rápida de un producto de calidad. La alta frecuencia de las iteraciones también evita largos periodos de aislamiento del equipo de desarrollo con el cliente, lo que implicaría una montaña de información que procesar cuando se produjera la próxima reunión.

5.- Capacitar al equipo

En Lean Development nos olvidamos de la creencia de que los administradores deban llevar todo el peso del proyecto y deban decirle a los trabajadores cómo tienen que trabajar. Este principio se basa en dos puntos clave:

- 1-. Los administradores deben dejar a los trabajadores hacer su trabajo ya que, al final, son ellos los especialistas en las distintas facetas del proyecto. Además, deben ser capaces de escuchar a los trabajadores para tener en cuenta posibles mejoras sugeridas por ellos.

- 2-. Los administradores deben encargarse de forma activa de que no se pierda la motivación, el trabajador debe tener en todo momento un objetivo claro y común con el resto, y esto es una tarea que los jefes deben llevar a cabo. En cierto modo, su trabajo es evitar que los desarrolladores se sientan máquinas trabajando de manera lineal sin un objetivo claro, y pasen a ser personas cuya meta sea la principal motivación y el porqué de su trabajo.

6.- Construir integridad intrínseca

Partimos de la necesidad de que el cliente tenga una visión general del sistema. El cliente quiere saber qué hace el sistema, si es intuitivo, cuánto cuesta mantenerlo... Para ello necesitamos la integridad conceptual: que todos los componentes del sistema que funcionan bien de manera separada lo hagan también juntos. Una forma de construir integridad intrínseca es tomar la información en pequeños lotes y prescindir en la medida de lo posible de la documentación, en favor de la comunicación cara a cara, tanto con el cliente como entre los miembros del grupo. Con ello conseguiremos un equipo más cohesionado y un trato más personal con el cliente. Otra forma de construir integridad intrínseca es la refactorización, es decir, limpiar el código de manera que no haya bloques de código repetido (desperdicios) o simplificar el *cómo* hacen una tarea las funciones sin modificar el *qué* hacen.

7.-Ver todo el conjunto

Puesto que los errores y desperdicios tienden a acumularse a medida que el proyecto transcurre, es muy importante que desde el principio se tengan en cuenta las interacciones de nuestro proyecto con el resto de sistemas de nuestra compañía. De esta forma se evitarán errores que ya se habían cometido previamente, así como ayudará a entender mejor cómo se debe llevar a cabo el proyecto para lograr el mayor valor posible para el cliente.

Otro punto a tener en cuenta en este principio es el no gastar demasiado tiempo ni recursos en un factor concreto. Por ejemplo, obsesionarse con una futura falta de tiempo o financiación puede provocar el abandono de otras tareas de igual importancia.