

Kanban (*tarjeta para visualizar*)



IIC2143 : Ingeniería de Software

看板

Kanban es un método para mejorar el proceso

Basado en la mentalidad *lean* (eliminar el *waste*, decidir lo más tarde posible, entregar lo más rápido posible, y otros)

Diferente de Scrum y XP:

- no prescribe roles ni prácticas de gestión o desarrollo
- no es un método de gestión de proyectos ni de desarrollo de proyectos

Kanban es que miremos la forma en que trabajamos hoy

... entendamos cómo el trabajo está fluyendo a través del sistema

... y experimentemos con cambios pequeños

... y con los límites al trabajo en progreso (*WIP*) para ayudar al equipo a establecer un sistema *pull* y eliminar lo que no sirve (*waste*)

Los siete *wastes* del desarrollo de software

Trabajo hecho a medias: al empezar muchas cosas al mismo tiempo y no poder terminarlas, al final de la iteración vas a tener mucho trabajo que no está listo para ser demostrado ni menos entregado

Procesos extras: no te ayudan a entregar software —documentación de requisitos que no se desarrollan, reuniones de avance en las que te piden informes de avance— pero te hacen trabajar más

...

...

Funcionalidades (*features*) extras: nadie las pidió; si las agregamos, va a ser a costa del tiempo para las funcionalidades que el cliente/usuario final si pidió; no cualquier innovación es beneficiosa para el proyecto

Cambios entre tareas: se le pide muchas cosas al equipo, como si no tuviera costo, y cada integrante termina teniendo que hacer cuatro, cinco o más tareas al mismo tiempo, cambiando de una a otra frecuentemente

Movimiento: cuando hay que ir constantemente de una oficina a otra para poder hacer el trabajo

...

...

Esperas: el equipo tiene que esperar a que alguien revise una especificación, o instale un sistema, o proporcione acceso a una base de datos (y no puede hacer mucho al respecto*)

Defectos: mientras más tarde nos damos cuenta de un error —p.ej., durante las pruebas de aceptación— más tiempo nos toma encontrarlo y corregirlo; es mejor que lo encuentre y lo corrija lo antes posible quien quiera que lo puso ahí

*: Hay *waste* que simplemente no se puede eliminar

... por eso es importante identificar todo el *waste* que sea posible, para eliminar lo que se pueda

Sistemas *pull*

Cada paso en el proceso es iniciado por el paso **siguiente**, que **tira** del resultado de ese paso anterior solo cuando ha terminado

Se establece un flujo constante de trabajo terminado a través de sistema,

...y los features se terminan de implementar tan rápido como es posible

Hay que establecer límites al **work in progress** (*WIP*) —el número e ítemes en los que se está trabajando en un momento dado para cada paso del sistema:

- p.ej., si un equipo de testing puede probar solo cuatro features a la vez sin que el proceso empiece a andar más lento,
- ...entonces ellos **establecen un límite WIP a la etapa anterior** de modo que el equipo de desarrollo nunca está desarrollando más que cuatro features a la vez

Así se reduce el tiempo *lead* total: desde cuando el feature es identificado hasta que es entregado

Cambiarse (de un sistema *push*) a un sistema *pull* es cambiar la forma en que pensamos sobre el trabajo

... y también la forma en que hacemos el trabajo

El tablero Kanban nos permite visualizar el flujo de trabajo

Dividido en columnas, que representan el estado en que están los ítemes de trabajo:

- p.ej., *Definir, Diseñar, Construir, Testear, Instalar*

Los ítemes de trabajo se escriben en tarjetas:

- se escriben al nivel de propiedad del sistema (*features*)
- no están descompuestos en tareas (*tasks*), como en Scrum

...



...

Las tarjetas se van moviendo de columna para representar el flujo a través del proceso

Cada columna (estado) muestra claramente, en el encabezado junto al nombre, el límite al WIP correspondiente (acordado por el equipo):

- para que no se agregue más trabajo a un estado que ha alcanzado su límite

Cómo Kanban nos ayuda a mejorar el proceso:

Seis prácticas Kanban

1) Visualizar el flujo de trabajo:

- los equipos Kanban tienen que ver el todo
- lo primero que hacen es mirar cómo están trabajando y producir una representación visual —un diagrama— del proceso que están usando
- p.ej., Definir, Diseñar, Construir, Testear, Instalar

2) Limitar el trabajo en progreso (*WIP*):

- los equipos observan cómo los ítemes fluyen a través del sistema
- luego empiezan a experimentar limitando el número de ítemes en cada estado
... a ver si esto los ayuda a focalizarse, a que el trabajo fluya parejo, y a lograr más
- p.ej., Definir 2, Diseñar 2, Construir 9 6, Testear ?, Instalar ?

3) Gestionar el flujo:

- el equipo pone atención a qué tan rápido el trabajo avanza por el proceso
... y ve cuáles límites WIP producen el menor tiempo para entregar features al cliente
... tratando de mantener constante el ritmo de entrega
- el equipo mide el *lead time* (tiempo de principio a fin)
- p.ej., Definir 2, Diseñar 2, Construir 4, Testear 2, Instalar 6

4) Hacer explícitas las políticas del proceso:

- a medida que el equipo aprende a trabajar con el flujo, va estableciendo políticas que guían su trabajo
- los equipos Kanban comunican claramente estas políticas
 - ... así como se esfuerzan por descubrir, y escribir, las reglas no escritas que usan al tomar decisiones
 - ... para poder evaluar las políticas y las reglas y, si es necesario, cambiarlas
- p.ej., “Todos los miembros del equipo participan en las estimaciones y usan la escala de puntos de relato en la que nos pusimos de acuerdo”
 - ... “Cuando un miembro del equipo termina su ítem de trabajo, toma a continuación el ítem de más alta prioridad del backlog en el que pueda trabajar”
 - ... “Nadie va a trabajar en un ítem que no está en nuestro backlog”

5) Implementar ciclos de realimentación:

- crear un *check* para cada paso del proceso
... y para cada política y mejora que el equipo está implementando
... y así medir sus efectos y asegurarse de que están funcionando
- medir los tiempos de *lead* y de ciclo para asegurarse de que el proceso no se está haciendo más lento
- p.ej., Definir 2, *revisión*, Diseñar 2, *reunión de diseño*, Construir 4, *cobertura de los tests unitarios*, Testear 2, *resultados de los tests automatizados*, Instalar 6

6) Mejorar colaborativamente el proceso:

- p.ej., compartir todas las mediciones que se van juntando
... y alentar al equipo para que sugieran cómo seguir experimentando