Desarrollo Ágil con Kanban

Kanban, llega como metodología de gestión de proyectos, de la mano de la automotriz Toyota, representando estadísticamente, la metodología ágil que menor resistencia presenta en las compañías acostumbradas a las metodologías tradicionales.
La palabra Kanban, de origen japonés, se compone de dos términos: Kan que puede traducirse como "visual" y ban, como "insignia", siendo una traducción aproximada, "insignia visual".

Orígenes: De Toyota al Software

Kanban se basa en un sistema de producción que dispara trabajo solo cuando existe capacidad para procesarlo. El disparador de trabajo es representado por tarjetas kanban de las cuales se dispone de una cantidad limitada.
Cada tarjeta Kanban acompaña a un item de trabajo durante todo el proceso de producción, hasta que éste, es empujado fuera del sistema, liberando una tarjeta. Un nuevo ítem de trabajo, solo podrá ser ingresado/aceptado si se dispone de una tarjeta kanban libre.

Este proceso de producción, donde un trabajo se introduce al sistema solo cuando existe disponibilidad para procesarlo, se denomina pull (tirar) en contrapartida al mecanismo push (empujar), donde el trabajo se introduce en función de la demanda.

En el desarrollo de Software, Kanban fue introducido por David Anderson de la Unidad de Negocios XIT de Microsoft, en 2004, reemplazando el sistemas de tarjetas por un tablero visual similar al de Scrum, pero con características extendidas que veremos a continuación.

Las tres reglas de Kanban

Con tan solo tres simples reglas, Kanban demuestra ser una de las metodologías adaptativas que menos resistencia al cambio presenta. Dichas reglas son:

1. Mostrar el proceso
2. Limitar el trabajo en curso
3. Optimizar el flujo de trabajo

Veamos cada una de ellas.

Regla #1: Mostrar el proceso

Consiste en la visualización de todo el proceso de desarrollo, mediante un tablero físico, generalmente, públicamente asequible. El objetivo de mostrar el proceso, consiste en:

- Entender mejor el proceso de trabajo actual;
- Conocer los problemas que puedan surgir y tomar decisiones;

- Mejorar la comunicación entre todos los interesados/participantes del proyecto;
- Hacer los futuros procesos más predecibles.

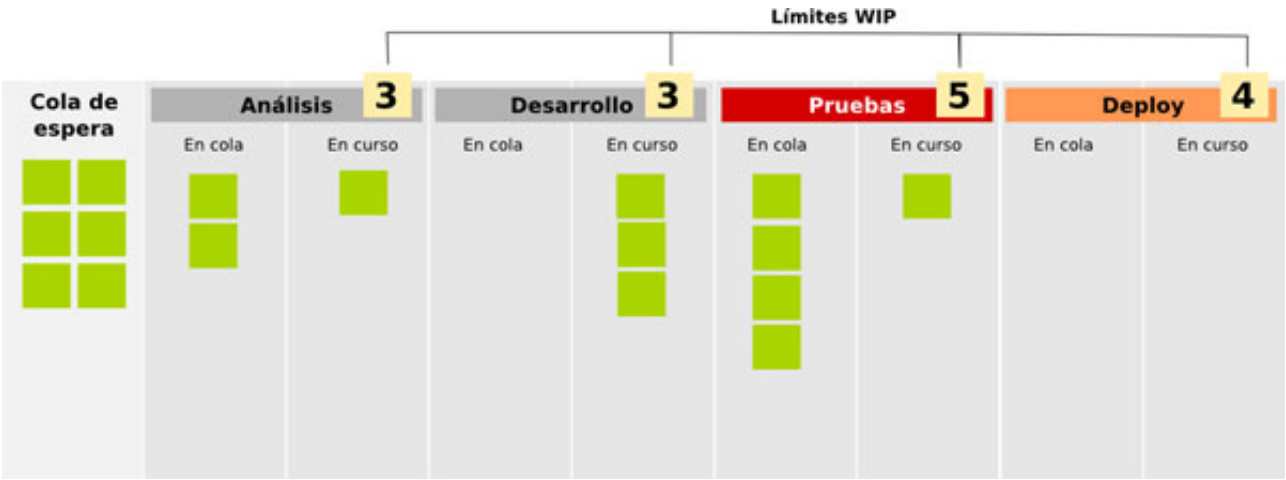
Un tablero Kanban, se divide en columnas las cuales representan un proceso de trabajo. Un ejemplo clásico de columnas para dividir un tablero Kanaban, sería el siguiente:
Cola de entrada | Análisis | Desarrollo | Test | Deploy | Producción

La cantidad y nombre de las columnas, varía de acuerdo a las necesidades de cada equipo y en la mayoría de los casos, éstas, son subdivididas en dos columnas: cola de espera y en curso

Regla #2: Limitar el trabajo en curso (WIP)

Los límites del WIP (work in progress trabajo en curso -) consisten en acordar anticipadamente, la cantidad de ítems que pueden abordarse por cada proceso (es decir, por columnas del tablero).
El principal objetivo de establecer estos límites, es el de detectar cuellos de botella.

Los cuellos de botella representan el estancamiento de un proceso determinado. Viendo el siguiente tablero ficticio, se puede comprender mejor:



En el tablero anterior, podemos visualizar claramente, que en la columna "pruebas" se produce un cuello de botella, pues el límite WIP está cubierto, mientras que el proceso siguiente (deploy), está totalmente libre. Esto, claramente marca un problema a resolver en el proceso correspondiente a las pruebas.

Es un valor a tener en cuenta, que la resolución de cuellos de botella, la mayoría de las veces, motiva la colaboración del equipo entre los diferentes procesos. Pues mientras existen procesos colapsados, existen a la vez, procesos libres para aceptar nuevos ítems. El cuello de botella ha generado un estancamiento, y los procesos libres, pueden ayudar a "desetancar" a los procesos colapsados.

Regla #3: Optimizar el flujo de trabajo

El objetivo una la producción estable, continua y previsible. Midiendo el tiempo que el ciclo completo de ejecución del proyecto demanda (por ejemplo, cantidad de días desde el inicio del análisis hasta el fin del deploy según el ejemplo del tablero anterior), se obtiene el CycleTime.

Al dividir, el CycleTime por el WIP, se obtiene el "rendimiento de trabajo", denominado Throughput, es decir, la cantidad de ítems que un equipo puede terminar en un determinado período de tiempo.

Throughput = CycleTime/WIP

Con estos valores, la optimización del flujo de trabajo consistirá en la búsqueda de:

- Minimizar el CycleTime
- Maximizar el Throughput
- Lograr una variabilidad mínima entre CycleTime y Throughput

Lectura recomendada:
A lo largo del manual de Desarrollo Ágil, hemos visto sobre Scrum y Kanban. Aún nos queda eXtreme Programming. Pero, mientras tanto, un buen libro para "agilizarnos" con Scrum y Kanban, es "Kanban y Scrum, Obteniendo lo mejor de ambos" de Henrik Kniberg y Mattias Skarin. Puedes acceder a la versión traducida al castellano por Agile Spain en:
http://www.proyectalis.com/documentos/KanbanVsScrum_Castellano_FINAL-printed.pdf