

# Kanban

## Su uso en el desarrollo de software

**Autor: Norberto Figuerola**

Kanban (en kanji, donde kan significa "visual", y ban significa "tarjeta" o "tablero") es un concepto de producción justo-a-tiempo (JIT). El kanban es una tarjeta física que se utiliza en el Sistema de Producción de Toyota (TPS - Toyota Production System) para soportar un control productivo descentralizado por demanda ("pull").

Kanban es una herramienta proveniente de la filosofía Lean, de tipo "pull", lo que significa que los recursos deciden cuándo y cuánto trabajo se comprometen a hacer. Los recursos toman ("pull") el trabajo cuando están listos, en lugar de tener que recibirlo ("push") desde el exterior. Al igual que una impresora tira en la página siguiente sólo cuando está lista para imprimir sobre ella. Kanban se basa en la optimización de procesos continuos y empíricos, que se corresponde con el principio de Kaizen Lean. Enfatiza la respuesta al cambio por sobre seguir un plan (generalmente Kanban permite una respuesta más rápida que Scrum).

La mayoría de los métodos ágiles como Scrum y XP ya están bien alineados con los principios de Lean. En 2004, sin embargo, David Anderson fue pionero en una aplicación más directa de Lean Thinking y Teoría de las Restricciones para el desarrollo de software. Bajo la guía de expertos como Don Reinertsen, desarrollaron un "sistema Kanban para el desarrollo de software". Así, mientras Kanban aplicado al desarrollo de software es algo relativamente nuevo, Kanban tal como se utiliza en Lean Production lleva más de un siglo y medio.

Kanban comparte con otras metodologías como FDD o Scrum la idea de crear un Backlog del producto que tenga una serie de items (user stories, features, etc.) priorizados. Pero la principal diferencia con otras aproximaciones ágiles, es que en Kanban no existen las iteraciones tipo timebox. En su lugar, Kanban se centra en controlar el **WIP (Work In Progress)**. Esto es, cuando hay poco WIP, se añade el item más prioritario del Product Backlog, y se controla que nunca se supere una cierta cantidad de WIP. Dadas sus características, no se adapta a un desarrollo basado en entregas, y actualmente se utiliza especialmente en entornos de mantenimiento (corrección de bugs, etc.). Al igual que Scrum o XP, Kanban podría formar parte de las herramientas de proceso de desarrollo ágil de cualquier compañía. La elección de cómo y cuándo usarla recae en cada uno en particular.

De hecho es común que Kanban pueda combinarse con los actuales procesos que usted tenga para el desarrollo de software, y existen muchos casos que lo utilizan para complementar (o reemplazar) a Scrum. Típicamente Kanban es una excelente herramienta de visualización del proceso, y el “Kanban Board” es una de las típicas herramientas denominadas en la jerga del agilismo como “Information Radiator”.

Para algunos practicantes de los métodos ágiles, el uso de cronogramas con diagramas de Gantt pasó a la historia. Incluso los tableros de Scrum pueden ser reemplazados por Kanban boards que refleja el estado del proyecto de forma totalmente transparente para todos los involucrados. En lugar de intentar continuamente capturar y representar el estado actual del proyecto a través de informes de estado, Kanban refleja a través de un tablero la situación de cada tarea del proyecto, con una visualización comprensible para todo el mundo, y permitiendo a la gente pensar por sí mismos y sacar las expectativas razonables sin necesidad de una interpretación excesiva por escrito del estado. Los siguientes elementos son esenciales:

**Stream:** visualizar el flujo real de trabajo a través de los equipos, y la comprensión de lo que están haciendo en cada paso. El tablero de Kanban consiste en una serie de columnas que representan las diferentes etapas de trabajo.

**Contenido:** representar los tipos diferentes de trabajo, y asegurarse de que todo el trabajo se visualiza de alguna manera que sea comprensible para todos los involucrados. Normalmente, esto se hace con las tarjetas que se colocan en las columnas definidas anteriormente.

**Límites:** definir claramente los límites de la cantidad de trabajo que el equipo puede soportar dentro de cada etapa del stream. Estos límites denotan el punto a partir del cual un equipo no logra progresos adicionales al tratar de tomar un trabajo adicional. Estos límites se definen como el número de tarjetas permitidas en cada columna.

**Políticas:** estas políticas toman la forma de declaraciones comprobables que declaran en forma explícita y específica, qué significa avanzar una tarea (tarjeta) de una columna a la siguiente. Las políticas ayudan a crear confianza en la visualización de todos los interesados.

Para ser más exactos y dado que Lean y Kanban nacieron mucho antes que los métodos ágiles en el ámbito de la manufactura, el empleo de estas herramientas para la industria del software es, relativamente nuevo, por lo que, existen numerosas formas de implementarlo, dado que no es una metodología de software que haya sido escrita y que imponga restricciones o prescripciones tal como lo hace Scrum. Por lo tanto, usted podrá leer distintas versiones de cómo implementar Kanban según el autor, pero en su mayoría todas parten de la misma base. Hoy podríamos decir que con la conformación del LSSC (Lean Software and Systems Consortium) ya existe algo unificado sobre la práctica de Lean y Kanban junto con su correspondiente certificación oficial en “métodos ágiles” luego de Scrum y DSDM. Kanban está ganando popularidad en los círculos de project management, fundamentalmente en lo relacionado con la visualización de workflow, mejores reportes, limitación del work in progress y balance del trabajo.

Los principios (leyes y guías) más importantes de estas herramientas y filosofía de trabajo y conforme al LSSC son:

- Eliminar los desperdicios (waste) en el sistema e incrementar la calidad
- Determinar el flujo de valor (“value stream mapping”) del workflow
- Remover las demoras del “workflow” aumentando la velocidad al acortar el “cycle time”
- Controlar el Work In Process (WIP) dado que el exceso incrementa tanto el riesgo como los desperdicios.
- Evitar el Multi-Tasking de los recursos
- Entrega de valor lo más rápido posible para incrementar el ROI, descomponiendo el producto en Minimal Marketable Features.
- Visualización del workflow a través del Kanban-Board

El sueño de todo jefe de proyecto es contar con la experiencia de un equipo sostenible en el tiempo y las entregas de alta calidad con la menor interrupción posible. Elementos esenciales en la realización de este objetivo incluyen una fuerte visión, la comunicación compartida, mejora sistémica del flujo y de la actividad, y la total transparencia creada a través de la representación visual de todo el trabajo. Kanban es un marco simple para hacer realidad estos beneficios, permitiendo visualizar el flujo de trabajo dentro del proyecto, estudiando formas de mejorar ese flujo mediante cambios, para luego estabilizar el flujo con los nuevos comportamientos. Por otra parte, el uso desenfrenado del multi-tasking a menudo invade los equipos y los arrastra hacia abajo en un estado no productivo e incierto. Kanban, aporta un enfoque sobre la limitación del trabajo en proceso (WIP). Esta restricción intencional proporciona un entorno donde los individuos y los equipos se centran en sus tareas hasta su finalización. Esta es la base de flujo continuo y constante de trabajo que se desplaza por el sistema.

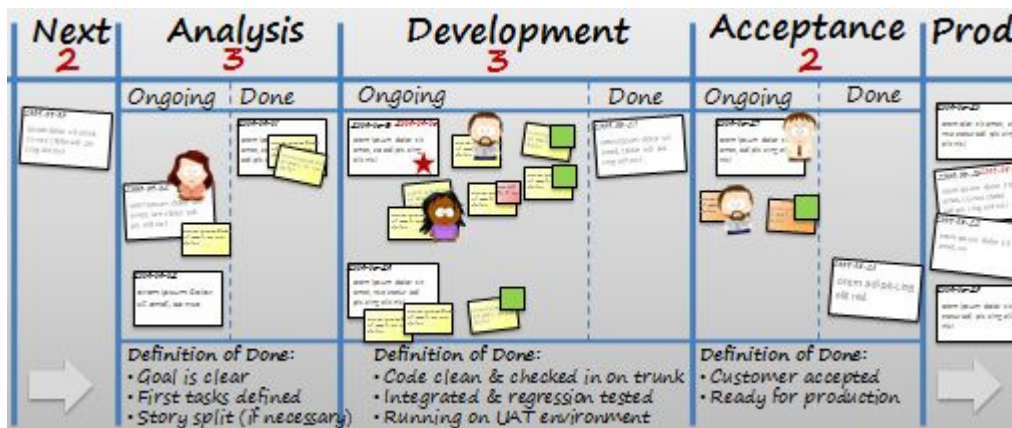
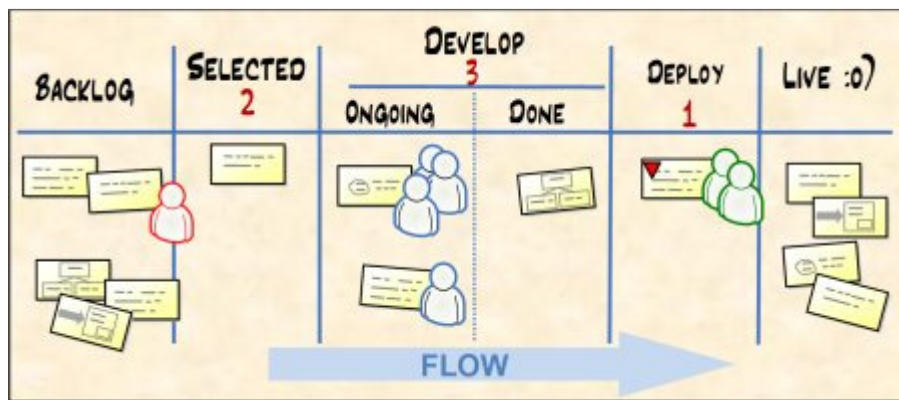
## **Cómo trabaja el Kanban ?**

### **1. Visualiza el “Workflow”**

- Divide el trabajo en piezas, y escribe cada una de ellas en tarjetas que se colocan en el tablero (“kanban board”).
- Utiliza columnas con nombres para ilustrar dónde se encuentra dentro del proceso o workflow cada ítem o tarea.

2. **Limita el “WIP”** (work in progress) – asigna limites específicos de cuantos items pueden estar siendo procesados a la vez dentro de cada columna del workflow.
3. **Mide el “Lead Time”** (también llamado “Cycle Time”) es el tiempo promedio para completar un item, o sea, que el mismo haya pasado por todas las columnas del workflow hasta llegar al final. El Kanban tiene por objetivo optimizar el proceso para hacer el lead time lo más pequeño y predecible que se pueda.

A continuación dos ejemplos de Kanban Board, donde se muestra el stream o flow y sus respectivas columnas que muestran en rojo el WIP limit.



El Kanban board tiene que lucir así? La respuesta es No, los gráficos son tan solo ejemplos, como dijimos anteriormente no es una filosofía prescriptiva. Lo único que Kanban prescribe es que el workflow pueda visualizarse y ser comprendido por todos, y que el WIP sea limitado (tal como lo indican los números en rojo de cada columna). El propósito es crear un flujo continuo a través del sistema y minimizar el lead time o cycle time.

Cada columna representa un estado dentro del workflow, o un buffer entre dos estados del workflow. La cantidad de columnas la determina cada uno de acuerdo a su experiencia y necesidad. Cuando se ha llegado al límite Kanban de una columna y no se tiene otra cosa que hacer, comience a buscar los cuellos de botella y solucionarlos. Si no existen es un indicativo que el límite podría ser demasiado bajo. Si se nota que los ítems permanecen en una columna por largo tiempo, es un indicativo de que el límite es demasiado alto. La regla es la siguiente:

- Límite kanban demasiado bajo = recursos desperdiciados = baja productividad
- Límite kanban demasiado alto = tareas en cola = erróneo lead time

### **Cuáles son los beneficios del Kanban ?**

- La visibilidad de los cuellos de botella (“bottlenecks”) en tiempo real. Esto permite al equipo que colabore para optimizar la cadena de valor en lugar de ocuparse cada uno de su parte. El síntoma de un cuello de botella es cuando vemos que la columna X está repleta de ítems mientras que la columna X+1 está vacía.
- Permite una evolución más gradual para pasar de waterfall a un desarrollo ágil de software, para aquellas empresas que aún no se animan a intentar dar ese paso.
- Es un modo de desarrollo ágil de software sin necesariamente tener que usar iteraciones “timeboxes” fijas como Scrum. Muy útil cuando los sprints no hacen mucho sentido, tal como grupos de operación y Soporte con un alto rango de incertidumbre y variabilidad.

### **Estimación y Velocidad en Kanban**

Existen muchos malentendidos respecto de Kanban cuando se lo compara con otros métodos ágiles. Por ejemplo:

- Con Kanban no se usan iteraciones. La realidad es que las iteraciones son opcionales.
- Con Kanban no hay que estimar. La realidad es que las estimaciones son opcionales también.
- Kanban es mejor que Scrum/XP. Kanban es solo otra herramienta más de proceso como las otras, que está orientada a manejar el workflow. No reemplaza a ninguna otra (si puede complementarla). Con Kanban se puede comenzar con cualquier proceso que se tenga, visualizarlo, introducir los límites WIP y evolucionar a partir de allí.

Para tener una idea clara sobre estimación y velocidad en Kanban repasemos muy brevemente lo que vemos durante el curso sobre estos temas relacionados con Scrum.

Uno de los aspectos únicos de un proyecto Agile es que la carga de trabajo para cada iteración se determina al comienzo de cada iteración. Sólo hay necesidad de planificar para la iteración actual y no una planificación anticipada pesada y completa como el método tradicional. Esto permite que el proyecto Agile sea mucho más flexible

Al comienzo de cada iteración se realiza una reunión entre "product owner" y el equipo del proyecto para determinar la carga de trabajo para la nueva iteración. Durante la reunión, se evalúa la cartera de los requisitos y se selecciona la siguiente serie de casos de uso (o historias de usuario) que son de más alta prioridad. El nivel de esfuerzo para cada caso de uso debería haber sido asignado cuando se añadió al backlog. Allí la estimación no debe ser horas de esfuerzo, sino más bien una serie de "story points."

Los story points son números arbitrarios utilizados para estimar el tamaño relativo de un caso de uso. La escala exacta es diferente de un equipo a otro. Por ejemplo, si un caso de uso tiene 10 puntos, otro caso de uso que es dos veces más complejo debería tener 20 puntos. La estimación total de los puntos incluye el tiempo de análisis, diseño, construcción, prueba e implementación. Durante la reunión de planificación, el equipo del proyecto toma tantos story points como los que pueda completar dentro de la iteración, dentro de los más priorizados. Por supuesto, las prioridades pueden cambiar durante el proyecto. Cuando esto sucede en un proyecto Agile, el cambio de prioridades es simplemente una cuestión de reasignar el orden de los casos de uso en el backlog.

Es importante que el equipo del proyecto pueda determinar rápidamente la cantidad de trabajo que puede completar en cada iteración. Esto permitirá que la carga de trabajo se mantenga relativamente uniforme de iteración a iteración. Si el equipo del proyecto considera que no fue capaz de completar una serie de casos de uso en la iteración anterior, se ponen de acuerdo para tomar menos trabajo en la siguiente iteración. Del mismo modo, si el equipo se da cuenta de que podían haber hecho más trabajo en una iteración, deberían tomar más trabajo en la siguiente iteración. Veamos un ejemplo:

Digamos que el equipo inicia el proyecto y estima que pueden completar 40 puntos en una iteración. Sin embargo, a medida que avanza la iteración, se encuentran con que tienen que trabajar muchas horas extras y que no pueden completar todo el trabajo durante la iteración. En la siguiente iteración, se decide entonces escoger sólo 32 story points. El equipo tiene éxito, pero durante la revisión final de la iteración se dan cuenta que podría haber hecho más. En la siguiente iteración deciden tomar 36 puntos de historia. Después de la iteración el equipo decide que este es un ritmo exigente pero totalmente factible para ellos. Por lo tanto, a partir de entonces el equipo toma 36 puntos de historia de casos de uso por iteración

Este ritmo al que el equipo puede completar los puntos de historia del backlog se conoce como la velocidad del equipo. Los casos de uso que son seleccionados para una iteración necesitan ser completados en esa iteración. En un proyecto ágil es importante mantenerse en un ciclo de repetición constante. Si el caso de uso no está listo cuando la iteración está lista para pasarla a producción, el código de dicho caso de uso debe ser retirado para que el resto de la iteración pueda ser liberada a tiempo. No hay retrasos en la fecha de finalización de la iteración. El equipo se centra en completar todos los story points en la fecha comprometida una y otra vez. Este ritmo constante en cada iteración también se conoce como el ritmo del equipo o cadencia.

Tanto Scrum como Kanban son ambos empíricos en el sentido que se espera que uno experimente con el proceso y haga los cambios necesarios para adaptarlo. Se debe experimentar dado que ninguno de ellos provee respuestas, solo indican una serie de prácticas.

En Scrum, los equipos estiman el tamaño relativo (= cantidad de trabajo) de cada elemento que se comprometen a hacer en una iteración, normalmente story points. Sumando el tamaño de cada elemento completado al final de cada sprint, se obtiene la velocidad. La velocidad es una medida de la capacidad, la cantidad de cosas que podemos entregar al final del sprint. Si conocemos la velocidad media del equipo, supongamos un promedio de 8, eso es bueno, porque entonces podemos hacer predicciones realistas acerca de qué elementos se pueden completar en los próximos sprints, y por lo tanto hacer planes realistas de liberación. Como vimos los equipos de Scrum miden la velocidad por la cantidad de "story points" que logran hacer en cada iteración. Una vez que se conoce la velocidad de un equipo esa será su cadencia o ritmo y además el límite de WIP (o al menos una guía). Un equipo con un promedio de velocidad de 10, normalmente no tomará más de 10 items (o story points) en un sprint. Por lo tanto en Scrum el WIP está limitado por la velocidad

En Kanban, la estimación no está prescrita. Así que si usted necesita dar fecha de compromisos, tendrá que decidir la forma de garantizar la previsibilidad. Es por ello que algunos equipos eligen hacer estimaciones y medir la velocidad al igual que en Scrum o de manera similar (en lugar de "story points", usan tamaño: S, M, L, XL). Lo más común es medir el tiempo de espera promedio por MMF, y tomar la métrica del promedio de lead time por MMF y utilizar eso para establecer el SLA, por ejemplo, "cuando nos comprometemos con un MMF se entregará siempre dentro de 15 días". En Kanban el WIP está limitado por el workflow. En los ejemplos de Kanban boards de más arriba, cada columna del workflow tiene indicado el límite máximo para cada una. Una vez que tenemos los límites de WIP determinados, podemos comenzar a medir y predecir el "lead time", el tiempo promedio que un ítem tarda en moverse a lo largo del Kanban board. Teniendo lead times predecibles, nos permite comprometernos con SLA y hacer planes más realistas.

## Conclusiones

El objetivo de Kanban es minimizar el WIP (Work in Progress o Trabajo en Progreso), o stock, entre los procesos. "Pull" significa que los trabajadores del proceso posterior solo toman el trabajo de los procesos anteriores si no se excede el límite WIP.

En el tablero, las tareas técnicas se representan con tarjetas y el estado de cada tarea se muestra al pegar la misma en distintas áreas o columnas. Una tarjeta Kanban es la contrapartida de una tarea que surgió de una "historia de usuario", las cuales tienen escritas información como la identificación de la tarea, su nombre, tiempo estimado, quién está asignado a la tarea, etc. Pero en sistemas de Kanban lo que fluye debería ser algo con valor reconocible para el cliente. En este caso, es más frecuente que en Kanban las tarjetas se correspondan a una "característica", de forma que todos los involucrados en el proceso, incluso el cliente, pueden comprender el significado y el valor de lo que fluye. Las "Historias de Usuario", "Items del backlog" o "Casos de Uso" se los llama de forma abstracta como MMF (*Minimal Marketable Feature*) de forma que muestran explícitamente que el flujo contiene valor para el cliente. El desarrollo ágil puede definirse como "hacer que las MMF fluyan lo más rápido posible a través de todo el proceso".

Una vez que el flujo queda establecido, los cinco conceptos del "Pensamiento Ágil" pueden aplicarse directamente a todo el proceso. La gestión del proceso sigue los siguientes principios:

- **Especificar el valor** desde la perspectiva del cliente - Especificar y ordenar los MMF.
- Identificar el **flujo de valor** y eliminar los desperdicios - Encontrar los bloqueos (tareas bloqueantes).
- **Hacer que fluya el valor** por la demanda del cliente –
- Involucrar y participar a los **empleados** - Participar al equipo en el Gemba.
- **Mejorar continuamente** en búsqueda de la perfección - Reflexión y Kaizen.

Se utilizan los sistemas Kanban para lograr dos objetivos:

1. mejor control del proceso - mantienen un flujo continuo y limitan el WIP.
2. mejor proceso de mejoras - el flujo es visible y se estimula el Kaizen.