

El Kanban

Marcos Bermejo

PID_00198057



Los textos e imágenes publicados en esta obra están sujetos –excepto que se indique lo contrario– a una licencia de Reconocimiento-NoComercial-SinObraDerivada (BY-NC-ND) v.3.0 España de Creative Commons. Podéis copiarlos, distribuirlos y transmitirlos públicamente siempre que citéis el autor y la fuente (FUOC. Fundació para la Universitat Oberta de Catalunya), no hagáis de ellos un uso comercial y ni obra derivada. La licencia completa se puede consultar en <http://creativecommons.org/licenses/by-nc-nd/3.0/es/legalcode.es>

Índice

Introducción	5
1. Introducción al Kanban	7
2. Definición de Kanban	8
3. Aplicación práctica del Kanban	10
3.1. El primer paso	10
3.2. Un panel Kanban para gestionarlos a todos	12
3.3. Leyendo un panel Kanban	12
4. El trabajo en curso (WIP)	14
4.1. Limitando el WIP	14
4.1.1. Limitando el WIP en la fase de desarrollo	14
4.1.2. Limitando el WIP a la fase de análisis	15
4.2. Multitarea y tiempo de <i>switch</i>	16
4.2.1. Experimento para entender el problema de la multitarea	17
5. Los cuellos de botella	19
6. Optimizando el equipo	21
6.1. El trabajador T	21
6.2. Los beneficios de desarrollar en parejas	22
6.3. El <i>bus factor</i>	23
6.4. Kaizen: una cultura de mejora continua	23
7. Optimizando el flujo	25
7.1. Dividiendo fases para conseguir flujo	25
7.1.1. Dividiendo fases para conseguir espacios	26
7.2. Métricas en el Kanban	27
8. El Kanban combinado con otras metodologías	29
8.1. Mini resumen de qué es y cómo funciona Scrum	29
8.2. Diferencias básicas entre Scrum y Kanban	30
8.3. Combinando Scrum y Kanban: Scrumban	30
8.3.1. El acercamiento Kanban mejorado con técnicas de Scrum	30
8.3.2. El acercamiento Scrum mejorado con Kanban	31
8.4. Buscando el equilibrio Scrum-Kanban	32

9. Conclusión.....	33
Bibliografía.....	35

Introducción

Derivado de la combinación de las dos palabras japonesas, *kan*, que quiere decir ‘visual’, y *ban*, que quiere decir ‘tarjeta’, nace la palabra *kanban*, con la que se denomina una metodología de producción u organización del trabajo que se basa en señales visuales para gestionar el esfuerzo y dedicación del equipo de producción.

El Kanban permite al gestor del equipo de producción multimedia identificar atascos en la producción, mejorar el tiempo de servicio de tareas y mejorar la calidad en el proceso de producción.

El Kanban requiere una alta madurez y profesionalidad en el equipo y da como resultado una importante mejora en la capacidad de producción del equipo. En este módulo, vamos a ver cómo.

1. Introducción al Kanban

Para introducir al estudiante en la gestión de procesos Kanban, hablaremos de dos ejemplos reales, donde se aplica el Kanban de una manera muy sutil para obtener mejoras en la gestión del trabajo en curso.

IKEA

En la empresa de muebles IKEA, hay una gran guardería donde poder dejar a los niños pequeños mientras los adultos compran en el centro. En esta guardería, tienen un sistema muy curioso para controlar que nunca haya más niños de los que podrían asumir los monitores: disponen de cincuenta chalecos, que van colocando a los niños a medida que sus padres los van dejando en el parque de actividades. Cuando la guardería llega al máximo de su capacidad (están colocados los cincuenta chalecos), ningún niño nuevo puede entrar al parque sin que antes se haya liberado un nuevo chaleco. De esta forma, se aseguran de que el parque de actividades y los monitores (en definitiva, su equipo de producción) nunca asumirá más trabajo de su capacidad, ni por equivocación del equipo ni por influencia externa. A medida que profundicemos en el módulo, el estudiante entenderá la similitud de los chalecos de los niños y los monitores del parque con las tareas que va a desempeñar en un proyecto multimedia y el equipo de desarrollo.

Toyota

En un caso más complejo, en la producción de automóviles de Toyota, el sistema que controla la producción está basado en unas tarjetas que circulan por la cadena de producción, donde los trabajadores están organizados de tal forma que cada uno produce su parte siempre y cuando les llegue una tarjeta que les informe de que pueden producir. Si no les llega la tarjeta, tienen prohibido producir su parte de trabajo.

Tanto los chalecos del ejemplo de la guardería como la tarjeta en la cadena de producción de Toyota, ambos son los **Kanban**. La palabra *kanban* en minúscula la utilizamos para denominar esta señal visual que sirve para indicar al equipo (los monitores de la guardería o los ingenieros de Toyota) que el sistema organizativo de la empresa puede asumir un nuevo trabajo.



Kanban, en *kanji* 看板, donde *kan* (看) significa visual y *ban* (板) significa tarjeta.

En un **sistema Kanban**, ningún trabajador puede producir si no le llega una nueva Kanban.

2. Definición de Kanban

El **Kanban** es un sistema de gestión donde se produce exactamente aquella cantidad de trabajo que el sistema es capaz de asumir.

El Kanban es un sistema de gestión del **trabajo en curso (WIP¹)**, que sirve principalmente para asegurar una producción continua y sin sobrecargas en el equipo de producción multimedia. El Kanban es un sistema de gestión donde se produce exactamente aquella cantidad de trabajo que el sistema es capaz de asumir. El Kanban es un sistema de trabajo ***just in time***, lo que significa que evita sobrantes innecesarios de stock, que en la gestión de proyectos multimedia equivale a la inversión innecesaria de tiempo y esfuerzo en lo que no necesitaremos (o simplemente es menos prioritario) y evita sobrecargar al equipo.

⁽¹⁾Del inglés, *work in progress*.

El Kanban es una aproximación a la gestión del **cambio organizativo**, no es un proceso de desarrollo de productos multimedia o de gestión de proyectos. El Kanban es una aproximación a la introducción de cambios en el ciclo de vida de desarrollo de productos multimedia o metodología de gestión de proyectos ya existente. Con el Kanban, empezáis con algo en lo que estás ahora mismo en la gestión de vuestro equipo de producción. No hay que empezar de cero en la organización de una empresa para adoptar el Kanban.

En la gestión del trabajo en curso con Kanban, se busca un concepto clave como es **limitar el trabajo en curso**. Está demostrado que, cuanto más trabajo en curso se gestione a la vez, los índices de calidad disminuyen drásticamente. En la producción de proyectos multimedia, aumentar el trabajo en curso implica aumentar la cantidad de errores que este proyecto multimedia tendrá como consecuencia de la poca capacidad de concentración que los desarrolladores podrán dedicarle a las tareas.

Limitar el **WIP** implica un aumento considerable de la calidad del software generado en la producción de un proyecto multimedia.

Limitar el trabajo en curso mediante la gestión del trabajo con Kanban también tiene una consecuencia importante y es que **disminuimos el tiempo de servicio** de una tarea desde que entra al sistema hasta que sale. Disminuyendo la cantidad de trabajo en curso, conseguimos que el enfoque en cada una de las tareas sea mayor y que el tiempo dedicado a todas ellas, sumado, sea menor que el empleado en asumirlas todas de golpe.

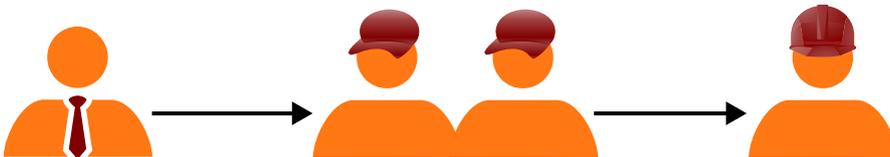
Por ello, podemos decir que el **Kanban**, pidiendo una estricta disciplina de limitar la cantidad de trabajo que el equipo lleva a cabo a la vez, nos retorna mayores índices de calidad y un tiempo de servicio bastante menor.

3. Aplicación práctica del Kanban

Para hacer más comprensible la aplicación práctica del Kanban, en este apartado vamos a imaginar un caso típico de la producción de proyectos multimedia, donde tenemos un equipo dedicado a dar servicio de esta clase de productos. En concreto, el servicio que proporciona este equipo se basa en el desarrollo de pequeñas tareas evolutivas de un producto ya instalado en casa del cliente. Nuestro equipo trabaja para diferentes clientes, que tienen contratado un servicio de mantenimiento donde desarrolla estos pequeños *plug-ins* a medida que los clientes van pidiendo.

El equipo está dividido en tres programadores, uno de ellos es analista-programador (el que tiene más experiencia) y un técnico experto en sistemas, este último es el que se dedica a solucionar problemas relacionados con configuraciones de servidor, además de pasar la fase de pruebas y calidad.

Supongamos que la realidad actual de este equipo es algo caótica. Están trabajando en varias tareas previstas para el mes que viene, además, día a día apagan varios fuegos debido a errores en desarrollos anteriores y podemos decir que el tiempo dedicado a testear es más bien corto, debido a las constantes tomas para cumplir los tiempos de entrega. Las predicciones y fechas de entrega no sirven, ya que se ven rotas por tareas más urgentes que les roban tiempo.



Entráis en este equipo y os piden gestionarlo. Por suerte, conocéis el Kanban y empezáis a aplicarlo.

3.1. El primer paso

Lo primero que tenemos que hacer en este equipo es representar su realidad actual de forma visual.

La forma más habitual de aplicar el Kanban en la producción de productos multimedia es mediante el *visual management*, es decir, la representación visual del flujo de trabajo mediante paneles que tienen que reflejar la realidad del equipo en cada instante.

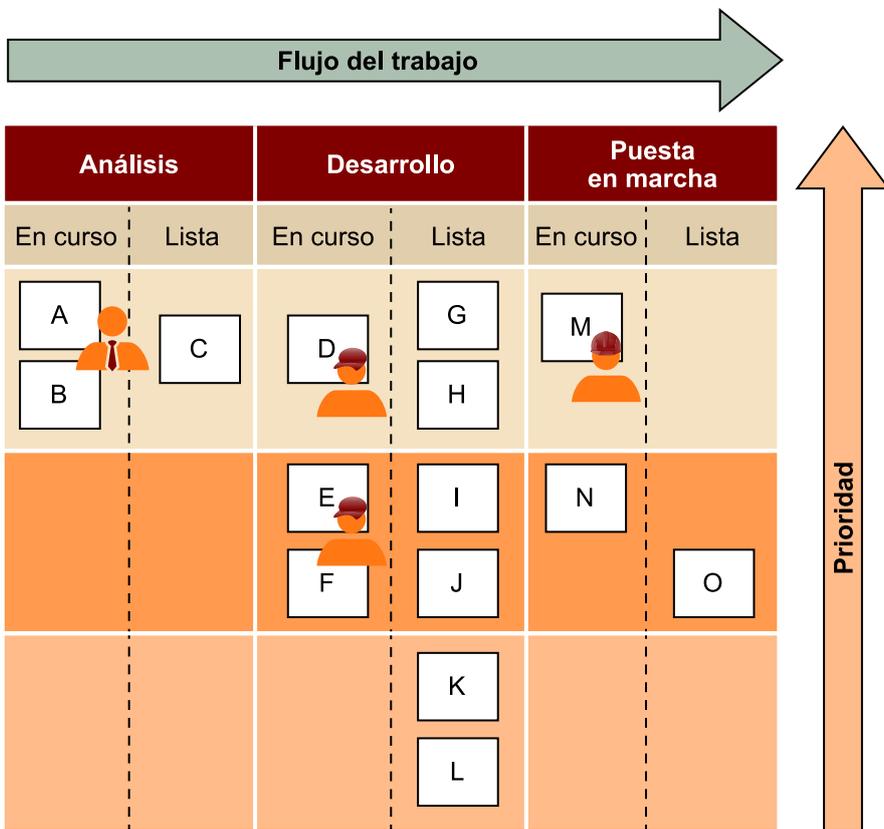
Representación visual

La representación visual del flujo de trabajo mediante paneles tiene que ser fiel a la realidad y estar constantemente actualizada.

- Tenemos que dar luz a todas las fases por las que pasan las tareas desde que entran en el sistema hasta que salen.
- Representar visualmente las tareas que el equipo está llevando a cabo ahora mismo.
- Aporta mucha información visual indicar qué miembro del equipo está ejecutando cada tarea.

Aseguraos de que el panel Kanban refleja las **fases**, las **tareas** y el **equipo**.

Un panel Kanban típico se implementaría tal como se muestra en la imagen siguiente:



En esta imagen vemos un panel constituido por tres columnas, que representan las diferentes fases por las que una tarea tiene que fluir para ser desarrollada (análisis, desarrollo y puesta en marcha). Cada fase está subdividida en dos estados, que son en curso y lista, para pasar a la siguiente fase; esta división está representada por la línea discontinua de cada fase. El estado en curso significa que el equipo está actualmente trabajando en esta tarea, en esta fase y el estado lista significa que el equipo ya ha acabado el trabajo que tenía que

ejecutar en esta fase y la tarea está esperando a que el sistema pueda asumirla para la siguiente fase. Esta división nos ayuda a localizar atascos en el proceso de producción, lo aprenderemos más adelante en el módulo.

Las filas podrían ser diferentes proyectos en los que la empresa está trabajando, pero lo más habitual es que las filas indiquen prioridad, donde las tareas más superiores son las más prioritarias.

Separar en cada fase las tareas en el estado en curso y lista nos ayuda a localizar atascos en el proceso de producción de productos multimedia.

3.2. Un panel Kanban para gestionarlos a todos

Nuestro panel Kanban tiene que ser **estudiado y juzgado en cada iteración** para detectar fases que falten o sobren. Nunca hay que adoptar un panel como solución universal.

En la gestión del trabajo en curso con Kanban, no existe un único modelo de panel adecuado para todos los equipos ni que cumpla todas las necesidades de la empresa. Un error frecuente de aquellos que empiezan a implantar Kanban en el equipo es intentar adoptar las fases de un modelo de panel externo en la realidad de la producción de su equipo. No se trata de cambiar las fases por otras, sino de estudiarlas, comprenderlas y hacerlas visibles. Cuando adoptamos Kanban, el panel tiene que ser construido y mejorado constantemente. En la gestión ágil de proyectos, una de las claves fundamentales es la iteración constante y la mejora a cada iteración. Como buenos agilistas, nuestro panel Kanban tiene que ser estudiado y juzgado en cada iteración para detectar fases que nos falten o nos sobren.

En el Kanban **no existe un único modelo de panel** adecuado para todos los equipos ni todas las situaciones. Simplemente representa de forma fiel la producción del equipo.

3.3. Leyendo un panel Kanban

Al acabar el primer mes como jefe de proyecto en el equipo del ejemplo presentado en la introducción de este apartado, tenemos un bonito panel Kanban en una de las paredes de la sala donde tenemos representada la realidad de nuestro equipo. El equipo lo mantiene actualizado de forma constante para que realmente represente el estado actual de los proyectos en curso.

En el panel del ejemplo, tenemos tareas que fluyen de izquierda a derecha de la siguiente manera:

- Los dos programadores están actualmente produciendo la funcionalidad de D, E y F en simultáneo. A la vez, el experto de sistemas está testeando en la actualidad la funcionalidad M.
- Mientras tanto, el analista-programador ha hecho el análisis y la estimación de algunas tareas nuevas que les han encomendado, esta estimación es muy importante, puesto que en ella se basa el presupuesto que la empresa hace de las tareas que el equipo lleva a cabo.
- A medida que el tiempo va pasando, los dos programadores van desarrollando a buen ritmo, tienen acabados seis módulos y están programando otros.
- El experto en sistemas está teniendo problemas, el módulo M no pasa algunos de los requerimientos de calidad de la empresa y solo él sabe cómo se hace, además, tiene problemas para instalar algunas tareas ya terminadas (N) en algunos clientes debido a la configuración de sus servidores.
- Solo una tarea ha sido registrada como finalizada en este mes (la tarea O) y, por lo tanto, solo un cliente ha recibido su petición con éxito.

El panel Kanban parece que señala el problema clave del equipo: ¿por qué seguimos programando más y más funcionalidades cuando muy pocas salen del sistema? En definitiva:

Un porcentaje muy bajo del esfuerzo y tiempo del equipo se está convirtiendo finalmente en ingresos y satisfacción del cliente.

Recordatorio

Tal como se explicó en la introducción de este módulo, el Kanban es un sistema de trabajo *just in time*, lo que significa que evita sobrantes innecesarios de stock.

En la producción de proyectos multimedia, estos **sobrantes de stock** son tiempo y esfuerzos dedicados a trabajo que no produce ingresos.

4. El trabajo en curso (WIP)

En el Kanban, hay que hacer que los trabajadores del equipo solo produzcan si el sistema acepta más cantidad de este trabajo producido.

Una vez hemos representando la realidad actual del equipo, tenemos que actuar y aplicar la primera restricción del Kanban, que es **limitar el WIP**. Tenemos que hacer obligatorio que los programadores dejen de programar y se dediquen a acabar de terminar aquellas tareas que están bloqueadas. Hay que lograr que los trabajadores del equipo solo produzcan si el sistema acepta más cantidad de este trabajo producido.

Recordemos los ejemplos de la introducción del módulo: hay que hacer que no entren más niños a la guardería si no hay más chalecos libres, puesto que el sistema podría sobrecargarse.

Tenemos que limitar el trabajo en curso.

Norma Kanban

Limitar el WIP es una de las normas que impone el Kanban. Mejorar el tiempo de entrega y mejorar enormemente la calidad de los desarrollos.

4.1. Limitando el WIP

¿Cómo limitamos el WIP? Ya lo hemos mencionado antes, en la gestión ágil de proyectos la clave es experimentar empíricamente e iterar para mejorar decisiones tomadas con anterioridad. Así que lo que vamos a hacer será limitar el WIP en un número no demasiado pequeño al principio.

Al inicio de la adopción del Kanban en el equipo, es importante empezar con un límite de WIP suficientemente cómodo para que no sea demasiado traumático al principio.

La mejor opción es optar por un límite alto e irlo bajando a cada dos o tres semanas hasta encontrar el equilibrio.

4.1.1. Limitando el WIP en la fase de desarrollo

En el caso del ejemplo expuesto en el apartado "Aplicación práctica del Kanban", donde los programadores están llevando a cabo tres tareas (estado en curso) y tienen finalizadas seis (estado lista), una aproximación de límite del WIP puede ser de nueve tareas. Lo que estamos diciendo con este nueve es que no programaremos nada más hasta que alguna tarea de las que están en nuestra fase no entre en la fase siguiente.

Limitaremos a nueve todo aquello que estamos programando ahora más todo lo que está pendiente de ser asumido por la siguiente fase. Es decir, basándonos en el ejemplo, hasta que los programadores acaben de programar lo que ahora tienen entre manos no les permitiremos programar ninguna tarea nueva.

Recordatorio

¡Acordaos de hacer visible el límite WIP en el panel! Un cartel encima de cada título de fase es el lugar ideal.

Límite WIP \leq n.º de tareas en curso + n.º de tareas hechas pendientes

En este instante, los programadores no podrán programar nada más y su prioridad debe ser conseguir que las tareas fluyan hacia la derecha: ayudarán al experto de sistemas a acabar el trabajo pendiente.

4.1.2. Limitando el WIP a la fase de análisis

Debemos tener en cuenta que limitar el WIP no se aplica solo a la fase de desarrollo, sino que también las fases previas de definición y análisis de tareas deben tener un cierto límite WIP. En nuestro equipo ejemplo, nosotros, como jefes de proyecto, tendremos que limitar el WIP del analista-programador a un número más bien bajo (por ejemplo dos). Esta decisión nos dará como resultado dos consecuencias deseables:

Limitar el WIP a la fase de análisis

Limitar bajo el WIP de la fase más temprana en la producción de proyectos multimedia nos ayuda a blindarnos contra los habituales cambios de prioridad de última hora.

Si al departamento comercial de la empresa le decimos que solo se pueden estimar X desarrollos y que en ningún caso nos saltaremos esta norma, se lo pensarán más de una vez cuando prioricen una tarea. Y estaremos seguros de que lo que estamos haciendo es siempre lo más prioritario.

- 1) La primera y más obvia será que, como hemos estado diciendo, aumentará la calidad y el tiempo de entrega de las estimaciones y análisis del analista. Cuanto menos tareas esté estimando a la vez, más esmeradas serán.
- 2) La segunda es que los miembros de la empresa que actúan previamente al analista irán con más cuidado a la hora de priorizar el trabajo en el equipo.

En una organización de producción multimedia habitual, la persona que envía el trabajo al equipo de desarrollo no suele ser miembro de este equipo de desarrollo y suele enviar trabajo de forma continua, en paralelo, de forma más bien desorganizada y repriorizando constantemente. Las consecuencias negativas de esta falta de organización donde se trabaja con metodologías ASM (a salto de mata) son muy conocidas por todo el mundo (tareas hechas a medias debido al cambio constante de prioridades, bajos índices de calidad debidos a las prisas de última hora). Si limitamos el WIP al analista-programador, lo que conseguimos es decirle a esta persona que ahora mismo no aceptaremos ninguna petición de nuevo análisis y le aseguramos que en el momento en el que haya un nuevo espacio en el Kanban en la fase de análisis se le informará. La primera reacción será de rechazo, pero estaremos seguros de que, cuando

le avisemos de que el equipo está en condiciones de asumir una nueva tarea, esta persona incluirá la más importante (puesto que, en caso contrario, tendrá que volver a esperar) y poco a poco conseguiremos que las prioridades estén siempre muy definidas y estaremos seguros de que lo que entra en el equipo de producción es siempre lo más importante.

Un ejemplo representativo

En una importante empresa de software para la Administración pública donde el autor de este módulo trabajó, el equipo de desarrollo servía peticiones procedentes de seis personas distintas, directivos pertenecientes a áreas diferentes (área de contabilidad, área de expedientes electrónicos, área de documentación, área de tramitación, área de sistemas de gestión del territorio y área de ciudadanía) y el día a día de este equipo era de cambios de prioridades constantes: cuando el director A pedía algo, siempre era más urgente que lo de cualquier otro.

Durante un cierto tiempo, se optó por designar a una única persona que centralizaba las peticiones, pero como carecía de capacidad jerárquica y de decisión, no se convirtió en más que un mensajero y cada vez venía con unas prioridades diferentes.

La solución definitiva fue que el jefe de este equipo decidiera que solo se aceptarían cinco tareas en pendiente y solo cuando hubiera un espacio libre en el Kanban se podría asumir una tarea nueva y les informaríamos por e-mail. El efecto fue inmediato: los seis tenían reuniones diarias donde se priorizaban entre ellos la tarea más importante. Conseguimos orden en la ejecución de las tareas y comunicación entre los directivos de las diferentes áreas.

4.2. Multitarea y tiempo de *switch*

En las autopistas, a medida que aumenta la densidad de tráfico, la velocidad del tráfico disminuye. A priori, esta disminución de velocidad no tendría que ocurrir porque, si todos los coches fueran a la misma velocidad, no habría motivo para que el tráfico fuera lento o incluso se detuviera. El problema viene cuando los coches cambian de carril. Un cambio de carril hace que el coche que cambia reduzca la velocidad, por lo tanto, el de detrás también y esto se propaga hasta reducir considerablemente la velocidad de todo el flujo de coches completo. Esta es una buena analogía de cómo la multitarea afecta en negativo al tiempo de servicio en el Kanban.

Una gran densidad de tareas en desarrollo en el flujo del Kanban hace que la velocidad de servicio decrezca considerablemente.

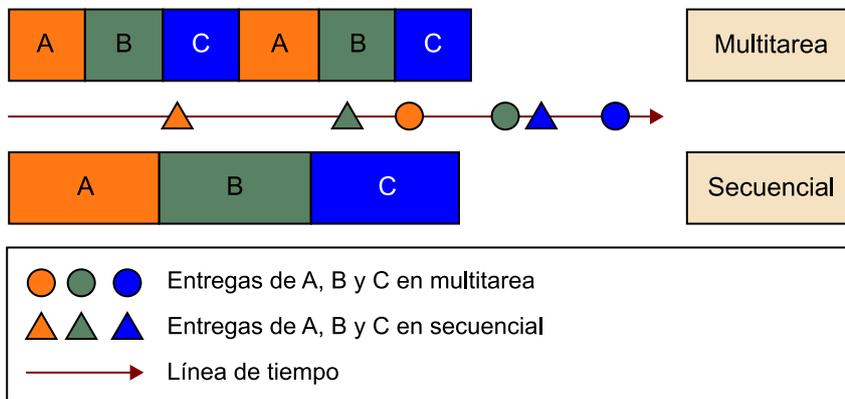
Cuando se cambia de contexto entre una tarea y otra se consume un cierto tiempo, por lo tanto cuantas menos veces tengamos que cambiar de contexto, menos tiempo se perderá. Gerald Weinberg, en el libro *Quality Software Management: Systems Thinking*, sugiere que se pierde un 20% del tiempo por cada tarea adicional que asumimos a la vez.

Por lo tanto, podemos decir que una tarea en desarrollo consume el 100% del tiempo disponible, dos tareas consumirán el 40% de tiempo cada una (habiendo perdido el 20% de tiempo en el cambio de contexto) y tres tareas a la vez consumirán cada una el 20% del tiempo disponible en ser desarrolladas y el tiempo perdido al cambiar de contexto será del 40%.



A medida que tenemos más tareas a la vez, más tiempo se pierde al cambiar de contexto.

Además, desarrollar las tareas de forma secuencial y no en paralelo tiene como consecuencia obtener resultados antes. El diagrama siguiente muestra que desarrollar A, B y C en multitarea tiene como resultado que A se entrega mucho más tarde, incluso C se entrega más tarde aunque se empezó antes. Debajo, el resultado de desarrollar las tareas secuencialmente:



4.2.1. Experimento para entender el problema de la multitarea

Ser capaz de desarrollar varias tareas en paralelo se ve habitualmente como una habilidad deseable a la hora de contratar a una persona, no obstante, se trata de una muy mala práctica. Y es que, como ya hemos aprendido, de este modo se invierte mucho más tiempo en cada tarea del que es necesario. Un experimento clásico para entenderlo nos llega por Jon Cook, en la lista de Yahoo Critical Chain.

Preparad una hoja en blanco y dibujad tres columnas, vamos a desarrollar tres proyectos, primero de forma multitarea y después secuencial. Estos tres proyectos son los siguientes:

- 1) en la primera columna, escribid las letras de la A a la J;
- 2) en la segunda columna escribid los números del 1 al 10;
- 3) en la tercera columna escribid los números romanos del I al X.

El resultado en la hoja tienen que ser tres columnas rellenas de la manera siguiente:

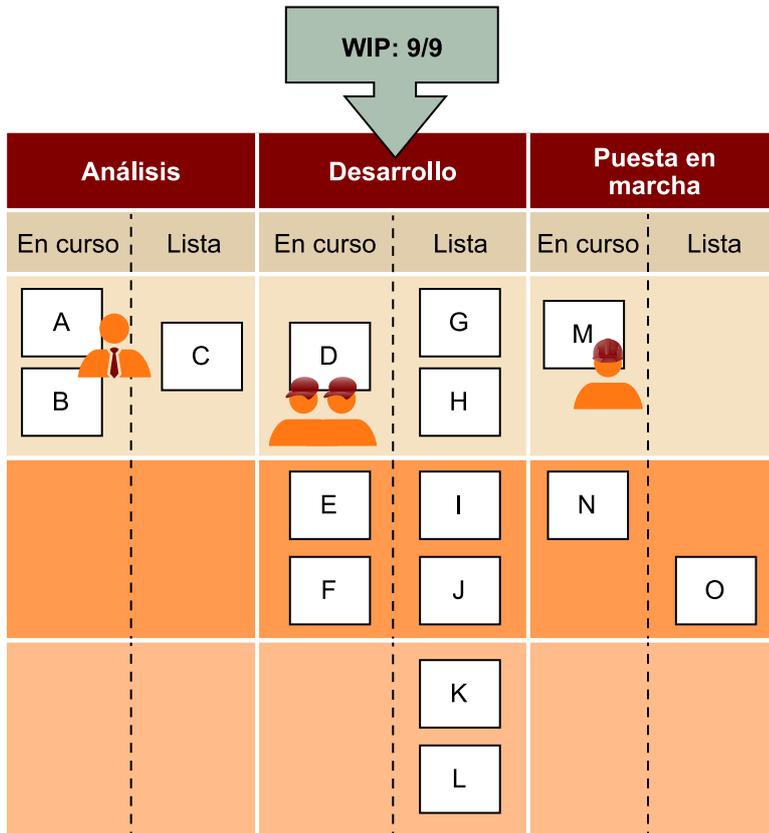
A	1	I
B	2	II
C	3	III
D	4	IV
...

Primero haced el ejercicio de manera multitarea, rellenas las tres columnas fila por fila, empezando por la A, el 1 y el I en la primera fila, después seguid con la B, el 2 y el II en la segunda, y así sucesivamente hasta acabar los tres proyectos.

Ahora, repetid el ejercicio de forma secuencial, primero rellenas la columna de la A a la J, después del 1 al 10 y por último la columna de los números romanos. No hará falta ni siquiera calcular el tiempo empleado con un cronómetro, notaréis la diferencia enseguida.

5. Los cuellos de botella

Una vez que la fase de desarrollo ha llegado al límite WIP, hemos llegado a tener un cuello de botella: la fase de puesta en marcha provoca un atasco.



En ese momento, se pueden tomar tres decisiones diferentes:

1) **La peor.** Aumentar o sacar el límite WIP. Sabemos la teoría y sabemos que el Kanban limita el WIP, pero como **no podemos**, y es **imposible**, decidimos sacarlo y ponemos un límite WIP de 10, para más adelante subirlo a 15. Y realmente no ayuda en nada a la empresa y al equipo.

2) **La mala.** Contratar a gente para la fase en el cuello de botella. La solución en un equipo sobrecargado no siempre es poner más gente. Se debe tener en cuenta que incorporar personal nuevo necesita tiempo de dedicación, hay un periodo en el que se tiene que invertir tiempo en formar al personal nuevo (nunca nadie entra sabiéndolo todo, por muy experto que sea quien se contrata) y este tiempo lo estaréis robando de las personas que se encuentran justamente en la fase de la que depende el tiempo de servicio del equipo (en el ejemplo del apartado "Aplicación práctica del Kanban", el experto en servidores).

3) **La buena.** Poner a uno de los programadores a ayudar en la fase de puesta en marcha. Un buen equipo de alto rendimiento es aquel que está formado por personas expertas en alguna tarea, pero que conoce y se interesa por todas las que le rodean. Se le denomina **trabajador T**.

Implementar el Kanban correctamente comporta un alto grado de compromiso del equipo.

En el momento en el que al programador se le dice "deja de programar y ponte a testear" una respuesta muy habitual es "mi trabajo es programar, a mí me pagan por programar y programar es lo que voy a hacer". En ese momento, hay que tener muy claro que el objetivo de todo el equipo tiene que ser **el servicio al cliente** (es decir, **facturar**). Un trabajador que no está dispuesto a asimilar que el trabajo de programar no es lo que el sistema de producción necesita en ese momento (donde el panel Kanban refleja que tenemos nueve tareas programadas y solo una facturada), no le está haciendo un buen favor a la empresa.

El éxito del Kanban

Implementar el Kanban de forma correcta comporta un alto grado de implicación en el equipo y hacer entender a los trabajadores que su trabajo no tiene que ser programar o instalar aplicaciones, sino hacer que la empresa obtenga beneficios.

Esta cultura tan oriental es lo que lleva a las empresas que implementan el Kanban al éxito.

6. Optimizando el equipo

Como hemos ido diciendo a lo largo del módulo, el Kanban es un espejo que refleja la realidad del trabajo que el equipo está desarrollando, refleja los flujos por donde pasan las tareas y el equipo en sí mismo. El Kanban nos hará aflorar ineficiencias en la gestión de los proyectos multimedia, pero también ineficiencias en el equipo.

6.1. El trabajador T

Las *personas T*² tienen dos características fundamentales y se usa la letra *T* para explicarlas:

⁽²⁾Del inglés, *T-shaped people*

1) La parte vertical de la *T* es una habilidad concreta y trabajada, que les permite contribuir en el equipo como **expertos** en esta característica (diseñador web, programador PHP, servidores APACHE, entre otros).

2) La parte horizontal de la *T* representa la disposición a colaborar a través de otras disciplinas.

La personalidad de un miembro del equipo de **tipoT** tiene dos aspectos principales:

1) Dispone de una alta **empatía** por las demás personas, que le ayuda a ver los problemas desde otros ojos, imaginar el problema desde otras perspectivas y poder contribuir con su opinión en áreas en las que *a priori* no es experto. Por ejemplo, el diseñador web es experto en diseñar, pero puede contribuir a imaginarse cómo poder hacer la tarea más sencilla al programador y el programador es experto en su trabajo, pero tiene que poder ver el proyecto con los ojos del testeador y detectar posibles problemas de arquitectura antes de que los encuentre su compañero.

2) Además, tienen que ser personas muy **entusiastas** y sentir **curiosidad** por las disciplinas de los demás, lo que les impulsa a interesarse e incluso a aprender y practicarlas.

Las personas de tipo T tienen ambas características: profundidad en una disciplina e interés por todas las que le rodean. Aseguraos siempre de formar todo el equipo de personas T.

6.2. Los beneficios de desarrollar en parejas

Una práctica muy extendida y probablemente la más conocida de desarrollo ágil de productos multimedia es el *pair programming* (programación por parejas). Esta práctica es muy recomendada en equipos que aplican el Kanban, puesto que ayuda a hacer posible este movimiento temporal de personas entre fases para desatascar el flujo de tareas.

Se conoce el *pair programming* por ayudar a construir un mejor software: dos jefes son mejor que uno y dos jefes producen habitualmente un software de mejor calidad.

Practicar el *pair programming* (sobre todo de forma rotativa) expande la esfera de conocimiento de todos los desarrolladores de un equipo. Este conocimiento, propagándose y expandiéndose, ayuda a los programadores a localizar el código repetido por todo el producto multimedia que se está desarrollando y, de una manera continua en toda la ejecución del proyecto, los programadores pueden ayudarse mutuamente y diseñar un mejor sistema.

En una oficina llena de gente, es habitual que haya ruidos y distracciones. El *pair programming* ayuda a evitar estas distracciones: cuando dos personas trabajan juntas en una tarea, se suelen distraer mucho menos que una sola. En empresas que practican el *pair programming*, es habitual encontrar a dos programadores tan absortos por una tarea que se abstraen perfectamente del resto de la oficina.

Practicando el *pair programming* surgen conversaciones sobre la tarea que se está ejecutando y se abren debates que siempre son beneficiosos para que afloren problemas que a una persona sola se le podrían escapar.

Las nuevas incorporaciones al equipo se vuelven productivas mucho más rápido que trabajando solas. En equipos de desarrollo, es habitual encontrar a una persona que trabaja sola y es complicado integrarla en el equipo. El *pair programming* socializa a estas personas y saca lo mejor de ellas mismas.

La revisión y el testeo de la programación tiene lugar de forma continua durante el desarrollo de todo el proyecto. Teniendo a una persona al lado, el programador comete menos errores y pasa por alto muchas menos deficiencias técnicas.

Trabajar en pareja es motivador, tendréis al equipo mucho más contento y animado. Evitaréis oír aquello de “cuando programé esto tenía un mal día”, puesto que el trabajo del programador no dependerá solo de él.

Lograréis que las personas de vuestro equipo se conviertan en personas de tipo T.

Como jefe de proyecto, conseguiréis no tener a una única persona que sepa sobre algún tema desarrollado. No tendréis que temer que tal persona se vaya de vacaciones o se ponga enferma. Reduciréis el *bus factor*.

El *pair programming* es una buena forma de conseguir esta unión en el equipo y la cultura y motivación necesarias para implementar el Kanban correctamente. Es una buena forma de implantar una **cultura Kaizen**.

6.3. El *bus factor*

Imaginad que estáis a medio camino de acabar un proyecto multimedia y aquella mañana vuestro equipo decide, como buen equipo Kanban que es, comer todos juntos. Tras comer, cruzan todos juntos por el paso de peatones cuando, de repente, un autobús alocado los atropella a todos. ¿Cuántas personas tienen que resultar gravemente heridas para que vuestro proyecto se tenga que cancelar?

Si respondiendo a la pregunta anterior habéis afirmado que una persona, actualad rápido y cambiadlo.

Como jefe de proyecto multimedia no debéis permitir que el éxito de un proyecto pueda depender de una persona. ¿Qué pasa si se pone enferma? ¿Y si se va de vacaciones o si le toca la lotería? ¿O si cambia de trabajo?

Procurad siempre mantener el *bus factor* muy elevado. En un equipo de cinco personas, es recomendable a partir de tres y, en un equipo de diez, como mínimo cuatro.

6.4. Kaizen: una cultura de mejora continua

La palabra japonesa *kaizen* significa literalmente mejora en el cambio.

La cultura en un puesto de trabajo donde la fuerza del trabajo está enfocada en la mejora de la calidad, la productividad y la satisfacción del cliente de forma continua durante todo el proceso de producción (no solo al principio o al final) se conoce como **una cultura Kaizen**.

Muy pocas empresas han llegado a este punto de cultura y la mayoría son de origen oriental.

Un ejemplo clásico de empresa con una fuerte cultura Kaizen es Toyota, donde la participación de los trabajadores en su dinámica de cultura corporativa es casi del 100%, donde tienen de promedio que cada empleado plantea como mínimo una propuesta de mejora al año.

En una cultura Kaizen, los individuos se sienten libres para emprender acciones y tomar decisiones, son siempre libres de hacer lo correcto por voluntad propia.

Espontáneamente, se forman grupos de discusión sobre problemas reales, tratan opciones e implementan soluciones y mejoras, entre todos deciden acciones concretas de mejora para la empresa. Los individuos tienen libertad de autoorganizarse (dentro de ciertos límites) en torno al trabajo que desarrollan y las tareas de desarrollo son autoasignadas por los trabajadores de forma voluntaria sin que haya ningún superior que las asigne. En esta cultura de empresa, la fuerza de trabajo no tiene miedo a tomar acciones voluntarias.

La norma subyacente es que la dirección sea tolerante a quiebras en experimentaciones e innovaciones si de esta acción se obtiene progreso y mejoras de rendimiento. Es una cultura de alta confianza, donde los trabajadores, sin importar su jerarquía, son los que tienen cuidado de la calidad y mejora de la producción y se involucran en un nivel de colaboración y una atmósfera donde todos cuidan del rendimiento del equipo y el equipo cuida del rendimiento propio.

Las estructuras organizativas de empresas con una cultura Kaizen, al contrario de las organizaciones en empresas de baja confianza, suelen ser muy planas, eliminan capas intermedias de dirección y control innecesarias y tienen como resultado una reducción importante de coste en coordinación.

Resulta bastante evidente que esta cultura de trabajo sea importada de culturas orientales, ya que en las occidentales nos enseñan desde pequeños a ser competitivos y a destacar de forma individual por encima de los demás, sobresaliendo en forma de héroes insustituibles alrededor de los cuales gira todo el resto. Cuanto más lejos nos encontremos de este modelo, mejor Kanban implementaremos.



Kaizen (改善) significa cambio para mejorar.

7. Optimizando el flujo

Los tres pilares básicos del Kanban son los siguientes:

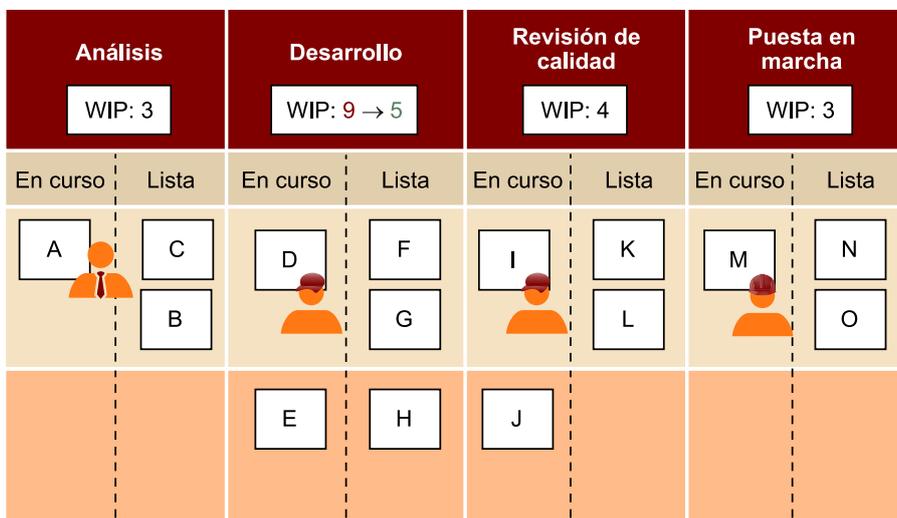
- 1) visualizar de forma continua el estado del equipo de producción,
- 2) limitar el WIP para mejorar la calidad y el tiempo de entrega, y
- 3) potenciar el flujo.

Como jefes de proyectos multimedia en un equipo Kanban, nuestro trabajo es conseguir que las tareas fluyan por el panel cuanto más rápido mejor. Tenemos que minimizar el tiempo que una tarea se queda en una fase.

7.1. Dividiendo fases para conseguir flujo

En nuestro equipo ejemplo del apartado "Aplicación práctica del Kanban", hemos tratado el caso en el que la fase de puesta en marcha se ha transformado en un cuello de botella; para hacerle frente hemos prohibido que ningún trabajo nuevo sea asumido por la fase de desarrollo y hemos puesto a los dos programadores a ayudar al experto en sistemas, así que podemos suponer que hemos conseguido desatascar el cuello de botella de la última fase. ¿Qué opciones de mejora podemos implementar ahora?

Muy probablemente, en este momento los programadores del equipo tengan unos conocimientos mínimos sobre las fases de calidad que tiene que pasar el experto de sistemas, quizás han aprendido algo sobre pruebas de rendimiento o cualquier tarea mecánica y sencilla que liberaría mucho trabajo al experto de sistemas.



¡Cuidado! No os fiéis del límite WIP

En un panel donde la fase X tiene un WIP de cinco, por ejemplo, no nos asegura flujo, puesto que puede haber cuatro tareas bloqueadas y no darnos cuenta de este bloqueo ya que las tareas se van desarrollando en el único espacio libre de la fase.

Revisad a diario el estado del panel, si alguna tarea lleva demasiado tiempo en una fase, estudiad el porqué.

Podemos dividir la fase de puesta en marcha en dos, podemos hacer aparecer la fase de revisión de calidad donde, por cada tarea desarrollada, uno de los programadores pasará aquellas pruebas de calidad que han aprendido del experto de sistemas.

Ahora, el experto se puede dedicar a aquellas tareas de especialista que solo él sabe hacer tan bien y las tareas pueden ser servidas al cliente con más rapidez.

Además, hemos podido disminuir el límite WIP de desarrollo de nueve a cinco, ¡casi la mitad! Ahora las fases están más claras, ninguna tarea será finalizada sin pasar esta revisión de calidad (llevada a cabo por los mismos programadores). Hemos ganado flujo en el panel.

Las acumulaciones de tareas en una fase son muy mala señal. Localizad qué tareas permanecen mucho tiempo en una fase, entended el porqué de esta acumulación y actuad en consecuencia.

7.1.1. Dividiendo fases para conseguir espacios

En apartados anteriores, hemos explicado los beneficios de cara a la organización de la empresa si limitamos el WIP de la entrada a producción (fase de análisis). Y hemos explicado que, a medida que las tareas se vayan sirviendo de izquierda a derecha, iremos obteniendo espacios libres y lo comunicaremos a los comerciales interesados en que un nuevo trabajo pueda ser asumido por el sistema.

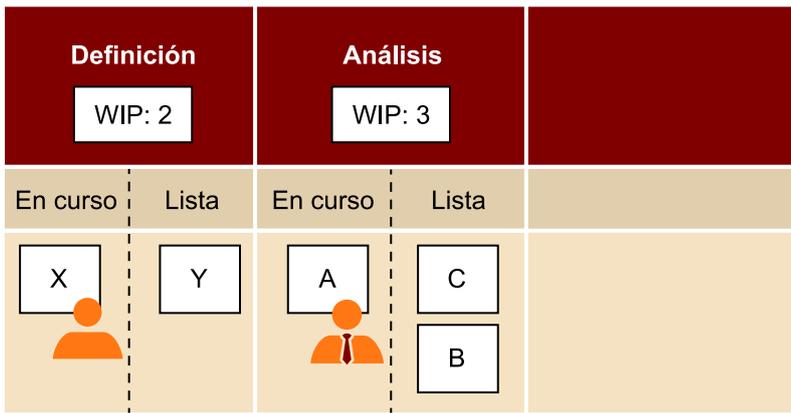
Actitud agilizadora

Como agente catalizador del agilismo en vuestra empresa, debéis estar siempre abiertos a posibles mejoras y modificaciones del proceso de producción de proyectos multimedia.

Y nunca os limitéis a la gestión de solo el equipo de producción.

Incluid en el Kanban todas las fases que os afecten directa o indirectamente.

Si en esta primera fase detectamos una posible división (definición-análisis), podemos ganar flujo si adoptamos una fase nueva en el Kanban y solicitamos a la empresa esta nueva figura de **analista funcional**, aquella persona que puede definir **qué tienen que hacer** las tareas que se van a desarrollar.



Con esta sencilla maniobra, hemos conseguido acoger en el proceso de producción un perfil nuevo, que casi no conocíamos y nos ha dado flujo en una fase muy temprana de la producción.

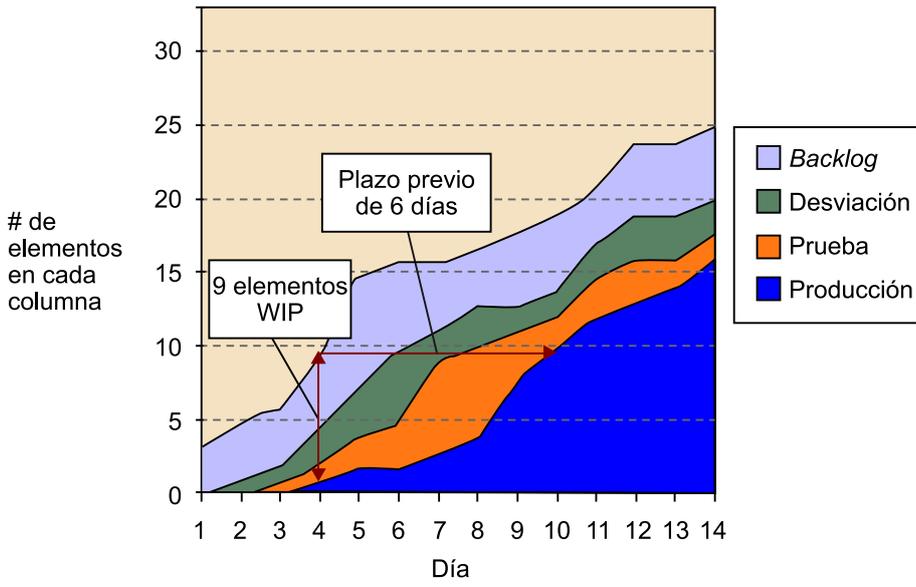
7.2. Métricas en el Kanban

En este punto de madurez del Kanban en nuestro equipo de producción multimedia, podemos empezar a obtener datos estadísticos del equipo.

Una muy buena opción es anotar las fechas de entrada y salida de la tarea por cada fase. Así podréis ir obteniendo toda una serie de gráficos del tiempo que tardan las tareas en ser servidas y qué fases son las que más tiempo exigen.

Un simple gráfico acumulativo nos dará mucha información en un solo vistazo: ¿cuántas tareas pendientes hay a día de hoy? ¿Cuántas hay aproximadamente en producción? ¿Cuál tiene una pendiente más pronunciada? Si crecen más las tareas pendientes que las servidas al cliente tenemos un problema.

Toda esta información se puede extraer de multitud de gráficos usando todo tipo de herramientas para la gestión de proyectos.



8. El Kanban combinado con otras metodologías

Tal y como hemos dicho al principio del módulo, el Kanban por sí solo no es una metodología de gestión de proyectos. No está orientado a la gestión, seguimiento y previsiones de un proyecto multimedia, es una herramienta perfecta para controlar el trabajo en curso y mejorar la calidad y el tiempo de servicio de la producción de proyectos multimedia. Ahora veremos cómo combinarlo con otra metodología ágil que sí está orientada al producto, como **Scrum**.

8.1. Mini resumen de qué es y cómo funciona Scrum

Scrum es una metodología ágil orientada al producto, empírica, iterativa e incremental. Donde se desarrollan iteraciones de tiempo definido, orientadas a obtener funcionalidades (pequeñas partes del proyecto entero) desarrolladas y acabadas con el objetivo de ir construyendo el proyecto en función de pequeñas entregas.

Ved también

El módulo "Scrum" está totalmente dedicado al tratamiento de esta metodología.

Scrum prescribe tres funciones:

- 1) El **product owner** es aquel que tiene la visión del producto, define funcionalidades y establece las prioridades.
- 2) El **equipo** es aquella función (rol único aunque esté compuesto por varias personas) que se encarga de la ejecución y la implementación de las funcionalidades.
- 3) El **Scrum master** es quien elimina impedimentos al equipo y proporciona liderazgo al equipo y lo protege.

Scrum prescribe cuatro tipos de intervalos de tiempo:

- 1) Tenemos la **iteración** (o **sprint**), con una duración de entre dos y cuatro semanas, donde se desarrollan las tareas.
- 2) Cada día del **sprint** se hace una pequeña sesión llamada **daily meeting** de duración máxima de quince minutos con el objetivo de sincronizar al equipo al inicio de la jornada.
- 3) También tenemos el **sprint planning meeting**, que es la reunión (de una a cuatro horas) donde se define el trabajo que se va a desempeñar en la iteración.

4) Finalmente, tenemos la **demo** y la **retrospectiva**, que es la reunión (de una a cuatro horas) donde se muestra el trabajo hecho y se toman decisiones de mejora de cara a la próxima iteración.

8.2. Diferencias básicas entre Scrum y Kanban

El Kanban no prescribe roles ni iteraciones fijas. Las entregas se pueden definir en función de las necesidades de la empresa: de manera regular (todos los viernes) o cada vez que se acabe un servicio y se ponga en producción.

Scrum no prescribe límite WIP o, como mínimo, no directamente. Scrum limita el trabajo en curso al trabajo comprometido en la iteración actual, es decir, si en el *sprint planning meeting* se decide llevar a cabo diez funcionalidades en dos semanas, el primer día del *sprint*, el equipo podría tener en desarrollo las diez a la vez.

Scrum no permite la entrada de trabajo nuevo cuando la iteración ha empezado. Si en el *sprint* se han comprometido diez tareas, Scrum no permite cambiarlas ni añadir otras nuevas a media iteración, por lo tanto en equipos donde las tareas estén definidas a corto plazo, iteraciones muy cortas son casi obligatorias. Si en la empresa pueden entrar tareas de hoy para mañana, estas no pueden ser asumidas por Scrum puesto que sirve desarrollos en iteraciones fijas de como mínimo una semana.

8.3. Combinando Scrum y Kanban: Scrumban

En un equipo de producción multimedia típico, tenemos tareas de las dos naturalezas:

- 1) **a largo plazo**, donde es adecuado hacer entregas iterativas cada cierto tiempo, y
- 2) **tareas de corto plazo**, donde se tienen que servir en el menor tiempo posible.

Uniendo Scrum con Kanban podremos hacer frente a esta realidad.

8.3.1. El acercamiento Kanban mejorado con técnicas de Scrum

El Kanban es libre de ser mejorado o combinado con otras técnicas de diferentes metodologías, siempre y cuando las restricciones y normas básicas del Kanban no se alteren (por ejemplo, no diremos "yo haré el Kanban pero no limito el WIP). ¿Creéis que una reunión diaria *daily Scrum* mejorará la producción del equipo? Pues hacedla. ¿Creéis que es necesaria la figura de un *Scrum master* que lidere y evangelice el agilismo en el equipo y la empresa? ¿Un *Scrum*

Recordatorio

El objetivo es siempre mejorar el proceso, el tiempo de servicio y la calidad. Adoptad aquellas técnicas y metodologías que hacen mejorar la realidad del equipo.

master que siempre tenga en mente el porqué de la limitación del WIP y sepa explicarla a cualquier director de la compañía? Adelante, seguro que obtendréis muy buenos resultados.

En la gestión ágil de proyectos, lo repetimos siempre, investigad, experimentad, revisad los resultados y tomad acciones en consecuencia.

8.3.2. El acercamiento Scrum mejorado con Kanban

En su definición, se dice que Scrum es un *framework*, un conjunto de herramientas y prácticas a vuestro alcance que todas juntas funcionan muy bien. Por lo tanto, se puede pensar que en la implementación de Scrum + Kanban queremos mantener esta filosofía de *framework* para el Scrum y no romperla a la primera de cambio. Por lo tanto, un acercamiento **Scrum entero + Kanban** es lo que el autor de este módulo prefiere: implementaremos Scrum tal y como se recomienda, pero al empezar la iteración, el trabajo en curso lo gestionaremos con el Kanban en los puntos siguientes:

- Scrum recomienda usar un panel para el seguimiento del trabajo en la iteración en curso, pero no pasa de la pendiente-en curso-hecho. Nosotros aplicaremos el Kanban para mejorar la fluidez de las tareas y dividiremos siempre que podamos estas fases. Como jefes de proyecto, tenemos que evitar tener una fase en curso demasiado ambigua y tendremos que saber detectar y hacer visible el flujo real por donde pasan las tareas. ¿Las funcionalidades desarrolladas tienen que ser subidas a un servidor de pruebas? ¿Tienen que pasar por una revisión? ¿Tienen que cumplir un mínimo de calidad definido por una lista de revisión definida por la empresa? Transformar la pendiente-en curso-hecho en pendiente-análisis-desarrollo-revisión-subida a servidor de pruebas-revisión por el departamento de calidad es la primera mejora que el Kanban proporciona a Scrum.
- Scrum no prohíbe que el primer día de iteración el equipo desarrolle el trabajo todo a la vez. Esta mala práctica hace que, muchas veces, el *sprint* acabe con muchas tareas a medias y pocas acabadas del todo. Para evitarlo, limitaremos el WIP durante la iteración. Definir un límite de funcionalidades en curso hará que las iteraciones acaben con menos funcionalidades a medias (que siempre son un problema para el Scrum) y más funcionalidades acabadas.
- Scrum prohíbe la entrada de trabajo nuevo durante la iteración. Esta restricción es ciencia ficción en la realidad de muchas empresas de productos multimedia. Siempre aparecen incidencias que se tienen que resolver antes de la fecha final de la iteración, aparecen fuegos urgentes que hay que atender. También puede ser que puntualmente aparezcan interesantes peticiones para el negocio de la empresa que hay que atender pronto,

concursos de posibles proyectos en los que la empresa quiere participar, pequeños desarrollos para contentar a un cliente histórico, entre otros.

Dotando al Scrum de esta posibilidad de atender imprevistos, el equipo será la envidia del barrio.

8.4. Buscando el equilibrio Scrum-Kanban

"Un momento, ¿qué pasa si la posibilidad de entrar trabajo a mitad del *sprint* se convierte en una costumbre?" En efecto, esta libertad de trabajo nuevo puede convertirse en un claro OALP (¡orcos a las puertas!). Necesitamos encontrar el equilibrio entre el trabajo previsto a largo plazo (Scrum) y el de imperiosa necesidad (Kanban). Para conseguirlo, podemos intentar seguir los consejos siguientes:

- Cuando todo es urgente, nada es urgente. Una excesiva sensación de urgencia en todo lo que entra hace que la credibilidad para la persona que dice que es urgente caiga inevitablemente.
- Atender el Kanban hace que dejemos de atender el Scrum. **Hacedlo visible.** Utilizad los gráficos de *burn-down* que prescribe Scrum y combinadlos con los gráficos de *burn-up* acumulativo del Kanban. **Podréis demostrar rápidamente que una bajada de velocidad en el Scrum se debe a una alta demanda de servicio en el Kanban.**
- Jugad con los límites WIP. Intentad poner un límite WIP diferente a las peticiones Kanban del de las tareas Scrum de la iteración en curso. Si detectáis que se están atendiendo demasiadas tareas Kanban y se está dejando de lado el Scrum, disminuíd el WIP en el Kanban y ampliadlo al Scrum.
- No caigáis en el cortotermicismo, es decir, dejarse llevar por la falsa sensación de alivio cuando dejáis de atender una tarea de largo plazo para dar servicio a todas las de corto plazo.

Recordatorio

El Scrum que no es atendido en esta iteración probablemente se convierta en un Kanban en el próximo *sprint*.

9. Conclusión

La gestión del cambio en un equipo de desarrollo es una tarea emocionante, continua y que representa un reto importante. Muchos equipos de productos multimedia trabajan mal, sin implantar ninguna metodología o implantando de forma incorrecta metodologías ágiles. Cada vez se necesitan más agilistas convencidos que entiendan por qué limitamos el WIP, por qué tenemos que cuestionarnos continuamente lo que nos viene heredado, como documentos inútiles, control y mando innecesario sobre los miembros del equipo o la multitarea.

Adoptad los cambios despacio, no dudéis en seguir el camino ágil y juzgad y mejorad constantemente el trabajo.

Estadísticas

En la web de la Agile Scout, podemos encontrar que más del 30% de las empresas de software trabajan mal, sin implantar ninguna metodología.

Bibliografía

agilescout

Anderson, David J.; Reinertsen, Donald G. (2010). *Kanban: Successful Evolutionary Change for Your Technology Business*. Disponible en: <http://www.amazon.com/kanban-successful-evolutionary-technology-business/dp/0984521402>.

limitedwipsociety

