

07-001

### **COMPARATIVE ANALYSIS OF THE BOARD TOOL IN THE AGILE METHODOLOGIES SCRUM, KANBAN AND SCRUMBAN IN SOFTWARE PROJECTS**

Fuentes Del Burgo, Joaquin (1); Sebastián Pérez, Miguel Ángel (2)

(1) Universidad de Castilla-La Mancha, (2) Dpto. Ingeniería de Construcción y Fabricación,  
UNED

Since the application of the principles contained in the Agile Manifesto in the software development industry, methodologies have been deployed that help to improve the management of this type of project. Agile methodologies include Scrum, Kanban and Scrumban, a combination of the above two. All three use boards to visualize the workflow and maximize efficiency in project management. The aim of this paper is to analyze the characteristics, similarities and differences of the boards and their use in these methodologies. A bibliographic review of articles, papers and books focusing on Scrum, Kanban and Scrumban has been carried out, collecting and comparing relevant information on the particularities and application of the board. The analysis showed differences in the configuration of the board and the way to use it between Scrum and Kanban, and more similarities than differences between Kanban and Scrumban. It is appreciated that each board and the way to use it is appropriate to the principles governing the agile methodologies discussed. Among the main qualities of the board stands out its adaptability according to the characteristics of the software project and its management needs.

Keywords: Scrum; Kanban; Scrumban; Board; Agile Methodology; Software Project

### **ANÁLISIS COMPARATIVO DE LA HERRAMIENTA TABLERO EN LAS METODOLOGÍAS ÁGILES SCRUM, KANBAN Y SCRUMBAN EN PROYECTOS DE SOFTWARE**

Desde la aplicación de los principios recogidos en el Manifiesto Ágil en la industria del desarrollo de software, se han desplegado metodologías que ayudan a mejorar la gestión de este tipo de proyectos. Entre las metodologías ágiles se encuentran Scrum, Kanban y Scrumban, una combinación de los dos anteriores. Los tres emplean tableros para la visualización del flujo de trabajo y maximizar la eficiencia en la gestión del proyecto. El objetivo del presente trabajo es analizar las características, similitudes y diferencias de los tableros y su empleo en estas metodologías. Al efecto se ha realizado una revisión bibliográfica de artículos, ponencias y libros centrados en Scrum, Kanban y Scrumban, recopilando y comparando la información relevante sobre las particularidades y aplicación del tablero. En el análisis han aparecido diferencias en la configuración del tablero y la forma de utilizarlo entre Scrum y Kanban, y más similitudes que diferencias entre Kanban y Scrumban. Se aprecia que cada tablero y la forma de utilizarlo se adecua a los principios de cada metodología. Entre las principales cualidades del tablero destaca su adaptabilidad en función de las características del proyecto de software y sus necesidades de gestión.

Palabras clave: Scrum; Kanban; Scrumban; Tablero; Metodología Ágil; Proyecto Software

Correspondencia: Joaquín Fuentes del Burgo. Correo: joaquin.fuentes@uclm.es



©2022 by the authors. Licensee AEIPRO, Spain. This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License (<https://creativecommons.org/licenses/by-nc-nd/4.0/>).

## 1. Introducción

El entorno económico exige aumentar la flexibilidad, generando ocasiones de negocios, forzando a que las organizaciones sean capaces de adaptarse y capitalizar estas oportunidades (Stoica et al., 2016). Además, en el actual mercado dinámico, las empresas de Tecnologías de la Información (TI) deben adecuar su entorno de trabajo para mantener una alta productividad en todos los aspectos de su negocio (Patil & Neve, 2018).

En 2001, un grupo de desarrolladores informáticos redactaron el Manifiesto Ágil para la gestión de proyectos de software (Lledó, 2014), constituida por cuatro valores y doce principios que ayudan a obtener un producto de alto valor, tratando de resolver los problemas históricos del desarrollo de proyectos informáticos (Lasa, Álvarez & de las Heras, 2017; Alqudah & Razali, 2018; Mircea, 2019). En la industria de las TI, el término "Ágil" se asocia a una metodología de gestión de proyectos (Mircea, 2019).

Ante la situación de cambios dinámicos, la mayoría de las empresas TI se han visto obligadas a modificar la gestión del desarrollo de sus proyectos (Sienkiewicz, 2012; Stoica et al., 2016; Albarqi & Qureshi, 2018; Gaborov et al., 2021). Muchas organizaciones migran al desarrollo ágil principalmente debido a la necesidad de acelerar la entrega de productos, mejorar la capacidad de administrar prioridades que cambian rápidamente, incrementar la productividad, mejorar la calidad del software, entregar un producto adaptado a las demandas del cliente con la calidad requerida y aumentar la satisfacción de las partes interesadas (Mahnic, 2014; Brezočnik & Majer, 2016; Albarqi & Qureshi, 2018).

En los últimos veinte años han aparecido numerosas metodologías ágiles para el desarrollo de software (Mahnic, 2014; Fuior, 2019) y aunque muchas comparten los principios Lean (Lledó 2014), se distinguen entre sí debido a la adopción y aplicación de principios y prácticas propias (Albarqi & Qureshi, 2018). Tienen características comunes: son iterativas, incrementales, mejoran la colaboración entre equipos multifuncionales autoorganizados, aplican prácticas orientadas a mantener la flexibilidad frente a los cambios que aparecen durante el proyecto, buscan la colaboración del cliente y la entrega rápida de un software de alta calidad (Banijamali et al., 2017; Albarqi & Qureshi, 2018; Molina et al., 2021).

Entre las metodologías ágiles utilizadas, las que se emplean con mayor frecuencia son Scrum y Kanban (Brezočnik & Majer, 2016; Banijamali et al., 2017; Bhavsar, Shah & Gopalan, 2020; Gaborov et al., 2021). Ambas tienen en común que se centran en los espacios de trabajo informativos, involucran el trabajo colaborativo en equipo y reconocen la importancia de la autoorganización de los equipos (Nikitina, Kajko-Mattsson & Stråle, 2012).

Scrum y Kanban no son metodologías competidoras mutuamente excluyentes. Pueden complementarse para mejorar el rendimiento de los equipos de desarrollo de software. Scrum proporciona un marco para el proceso de desarrollo iterativo e incremental, mientras que Kanban garantiza una alta visibilidad del flujo de trabajo y una rápida identificación de posibles cuellos de botella, lo que permite la mejora continua del proceso (Mahnic, 2014).

Diferentes investigaciones han demostrado la combinación de principios y prácticas de dos o más metodologías entre sí, mejora el proceso de desarrollo de software (Albarqi & Qureshi, 2018). Así, Scrumban, una mezcla de Scrum y Kanban, genera una sólida metodología frente a los desafíos como el control del flujo de trabajo, la gestión del tiempo de desarrollo o la entrega del producto (Bhavsar, Shah & Gopalan, 2020).

Las tres metodologías tienen en común el empleo de tableros y tarjetas como herramienta para la gestión del desarrollo del proyecto de software y facilitar su progreso (Al-Aidaros & Omar, 2017).

En este artículo se realiza una revisión bibliográfica que permite describir las particularidades de los tableros de cada metodología y sus principales características. El documento continúa con el apartado de metodología, seguirá una breve descripción de los aspectos básicos de

las tres metodologías ágiles. Continúa con el apartado de discusión y resultados, centrado en la descripción genérica de los tableros y su empleo. Finaliza con las conclusiones.

## 2. Metodología

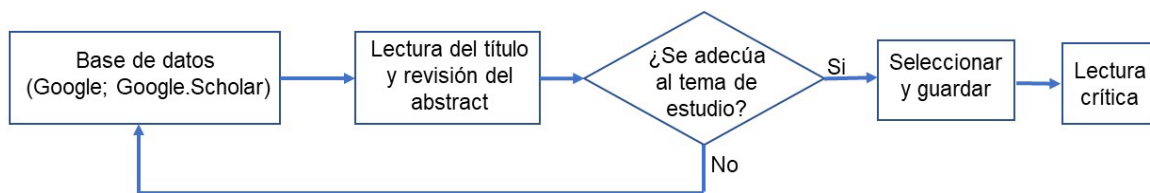
La revisión bibliográfica realizada ha sido un procedimiento estructurado para localizar y recopilar información publicada relacionada con las metodologías ágiles Scrum, Kanban y Scrumban (Gálvez, 2002; Guirao-Goris, Olmedo & Ferrer, 2008; Vilanova, 2012).

La información se encuentra en multitud de formatos y fuentes diferentes, entre las que se han elegido revistas, actas de congresos, libros, tesis y páginas web especializadas en metodologías ágiles (Vilanova, 2012; Gómez-Luna et al., 2014). La búsqueda en internet se realizó en el metabuscador genérico Google y en el buscador específico Google Académico (Scholar.google.com), empleando los descriptores en inglés (Vilanova, 2012): Scrum, Kanban, Scrumban, Scrum board, Kanban board y Scrumban board (Guirao-Goris, Olmedo & Ferrer, 2008; Vilanova, 2012).

Se han seleccionado aquellos documentos reconocidos que contienen información sobre los aspectos formales tratados (Gómez-Luna et al., 2014) (Figura 1). La revisión descriptiva se ha hecho con la información obtenida sobre las características y el empleo de los tableros en las metodologías tratadas (Day, 2005; Guirao-Goris, Olmedo & Ferrer, 2008).

El análisis de la información ha sido la fase que más tiempo ha requerido, con una lectura crítica que ha permitido identificar los conceptos claves, las ideas principales y los procedimientos utilizados y la comprensión del tema investigado (Gómez-Luna et al., 2014; Vilanova, 2012).

Figura 1: Estrategia de búsqueda de referencias



Nota: Adaptada de Vilanova (2012). Elaboración propia.

## 3. Resumen de las metodologías ágiles Scrum, Kanban y Scrumban

### 3.1 Scrum

Como metodología, Scrum prescribe procesos formalizados para el desarrollo y gestión de proyectos informáticos basado en los valores y principios del Manifiesto Ágil, en el empirismo y en los principios Lean (Reddy, 2015; Permana, 2015; Lei et al., 2017; Schwaber & Sutherland, 2020). Se trata de un enfoque holístico que procede con ciclos cortos (denominados *Sprints* o iteraciones), para organizar el desarrollo de software de manera incremental e iterativa. Las funcionalidades se realizan progresivamente y se incluyen en un producto entregable al final de cada iteración (Guérin, 2018).

Scrum tiene tres roles: el Propietario del producto (*Product Owner*) (es el responsable de definir las especificaciones del producto (*Product Backlog*) según las especificaciones del cliente y trata de maximizar el retorno de la inversión (ROI)); el *Scrum Master* (observa al equipo, se asegura de que no se violen las reglas de Scrum y elimina cualquier impedimento que el equipo pueda tener) y el Equipo (equipo multifuncional que es responsable de entregar

incrementos de producto entregables al final de cada *Sprint*) (Deemer et al., 2012; Brezočnik & Majer, 2016).

Scrum tiene una serie de artefactos o eventos que ayudan a realizar seguimiento del proyecto a lo largo de su desarrollo, maximizando la transparencia de la información y facilitando a los equipos oportunidades de inspección y adaptación (Arbeláez, Medina & Chaves, 2011; Gonçalves, 2018; García et al., 2020; Schwaber & Sutherland, 2020). El *Sprint* es una iteración de longitud fija y representa la unidad básica de desarrollo (de 2 a 4 semanas). Antes de cada *Sprint*, se lleva a cabo el evento *Sprint Planning* en el que se define la cartera de pedidos del *Sprint* (*Sprint Backlog*). Todos los *Sprints* finalizan con el *Sprint Review* y el *Sprint Retrospective*. En el *Sprint Review*, el Equipo y el propietario del producto hacen una demostración del producto y buscan oportunidades de mejora. En la *Sprint Retrospective* el Scrum Master y el Equipo tratan de mejorar y optimizar el proceso de desarrollo en sí mismo. Durante el *Sprint*, todas las mañanas se realizan reuniones diarias de 10 a 15 minutos (*Daily Scrum*), en las que cada miembro del equipo explica su avance, sus dificultades y su programa para la jornada (Brezočnik & Majer, 2016; Lasa, Álvarez & de las Heras, 2017; Gonçalves, 2018; Subra, 2020).

### 3.2 Kanban

Kanban es un método visual de gestión de procesos que procede del ámbito industrial, desarrollado en Toyota en la década de 1950, aunque es un concepto relativamente nuevo en el área de la ingeniería de software (Mahnic, 2014, Fuior, 2019; Subra, 2020). El objetivo principal de Kanban es la eliminación de desperdicios y demoras, aplicando la técnica del *Just-In-Time* (JIT) para la programación de las tareas, con el objetivo de reducir el tiempo de ciclo, aumentar la calidad y reducir los costos de producción (Anderson, 2010; Brezočnik & Majer, 2016; Albarqi & Qureshi, 2018).

Las prácticas clave en la aplicación de Kanban para la gestión del proceso generalmente incluyen: la visualización del flujo de trabajo utilizando el tablero Kanban; la limitación del trabajo en curso (*Work-In-Progress* (WIP)) a través de la disminución del número de funciones en la lista de tareas; aplicación de un "sistema *pull*" para desplazar el trabajo a través del proceso; la medición y la optimización del flujo; la clarificación de las políticas de gestión de los flujos de trabajo; la retroalimentación y el control constante; la mejora del proceso colaborativamente de forma incremental; y el empleo de modelos para reconocer oportunidades para la mejora (Anderson, 2010; Sienkiewicz, 2012; Brezočnik & Majer, 2016; Alqudah & Razali, 2018; Morales, 2020; Subra, 2020).

En comparación con Scrum, Kanban es una metodología menos estructurada y prescriptiva. Sin embargo, al igual que Scrum, Kanban también requiere equipos altamente autoorganizados y con una alta participación de liderazgo en la generación de otras prácticas, aunque no emplea roles como el *Scrum Master* (Nikitina, Kajko-Mattsson & Stråle, 2012; Stoica et al., 2016; Alqudah & Razali, 2018).

Kanban también emplea reuniones periódicas, denominadas cadencias, que fomentan la comunicación entre diferentes niveles de la organización. Una de las más importantes es la reunión diaria (*Daily Kanban*), que permite mantener al equipo coordinado e informado sobre el proyecto (Orszewski, 2018).

### 3.3 Scrumban

En su libro seminal, Ladas (2008a) definió Scrumban como un método de transición para mover equipos de desarrollo de software de Scrum a un modelo de desarrollo más evolucionado. Basándose en Scrum y Kanban, Scrumban enfatiza en la aplicación de los sistemas Kanban dentro del contexto de Scrum para ampliar sus capacidades y proporcionar

nuevas perspectivas y capacidades a los equipos de desarrollo (Reddy, 2015; Brezočnik & Majer, 2016; Mircea, 2019).

Como resumen de las prácticas de Scrumban (Ladas, 2008a; Albarqi & Qureshi, 2018; Alqudah & Razali, 2018; Patil & Neve, 2018), la extracción de tareas del *Product Backlog* se realiza de acuerdo a las necesidades, buscando un flujo continuo para reducir el desperdicio, es decir, no crear o analizar demasiadas historias de usuario. Al comienzo del proyecto se pueden emplear los diferentes tipos de *Sprints* de Scrum (*Sprint Planning*, *Sprint Retrospective* y *Sprint Review*), pero conforme el proceso de producción madure, el trabajo se ajustará a la capacidad del proceso, migrando más a un flujo continuo que a iteraciones. La reunión diaria (*Daily Meeting*) solo se realizará cuando sea necesario, para asegurarse de que haya una mejora continua.

A diferencia de Scrum, la estimación del esfuerzo se deja a criterio del equipo. No se enfatiza en tener todos los roles de Scrum, y será el equipo quién decida qué roles son apropiados y cuándo adoptarlos. El equipo estará formado por miembros con diferentes especialidades y experiencia. Además, se emplearán todas las prácticas Kanban que sean necesarias como la visualización del flujo de trabajo con el tablero, la mejora continua, gestionar y medir el flujo, limitar el trabajo en curso, hacer políticas explícitas e implementar comentarios (Ladas, 2008a; Albarqi & Qureshi, 2018; Alqudah & Razali, 2018; Patil & Neve, 2018). Ladas (2008a) propone el uso de buffers entre procesos y diagramas de flujo para mostrar las debilidades y oportunidades del proceso que faciliten la mejora continua (*Kaizen*).

## 4. Discusión y resultados

Aunque existen aplicaciones informáticas que permiten replicar su función, el tablero o pared de tareas simple representa dos valiosos principios Lean: tecnología simple y control visual. La utilidad de este método simple de gestión de flujo de trabajo es que es fácil de administrar y resulta fácil de cambiar (Ladas, 2008b; Morales, 2020).

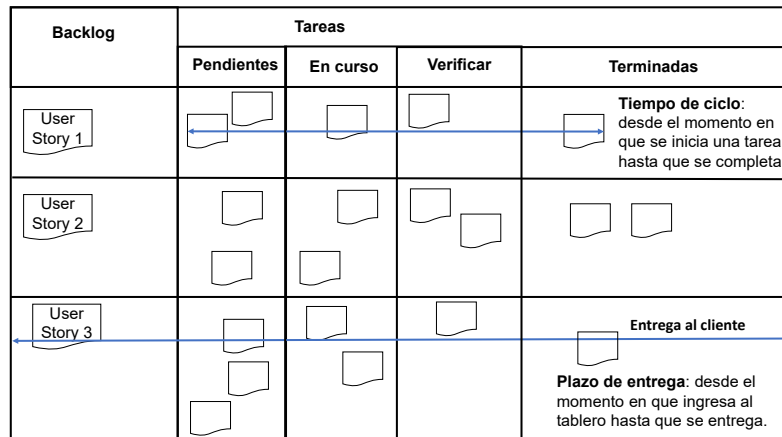
### 4.1 El tablero Scrum (*Scrum Board*)

La propuesta clásica del tablero Scrum consiste en dividirlo en las columnas *Backlog*, Pendientes (*To Do*), En curso (*In Progress*), Verificar (*To Check*) y Terminadas (*Done*) (aunque se pueden utilizar otros nombres) (Figura 2). Las filas del tablero corresponden a las historias de usuario (*User Stories*) que constituyen el *Backlog*, que se emplean para acumular las operaciones asociadas a cada característica del software. Las tarjetas (de tipo *post-it*), representan las tareas asociadas, con el nombre de la actividad indicado en ella (Guérin, 2018; Subra, 2020).

La columna "Pendientes" se dedica a las tareas seleccionadas para ejecutar durante el *Sprint*. Cuando un miembro del equipo decide en qué trabajar, mueve la tarjeta correspondiente a la columna "En curso", donde se encuentran las otras tareas en las que está trabajando el equipo. Una vez que se ha completado la tarea "En curso", se desplaza a la columna "Verificar", donde se probará la nueva funcionalidad del software asociada a la tarea. Finalmente, si se obtiene un resultado positivo en la prueba, se moverá a la columna "Terminado" y la tarea se considerará completa. El objetivo es que al finalizar el *Sprint* todas

las tareas planificadas se encuentren en la columna “Terminadas”, repitiéndose este ciclo de forma continua hasta que se complete el proyecto (Subra, 2020; Grotenfelt, 2021).

**Figura 2: Ejemplo de tablero Scrum**



Fuentes: PMI, 2017; Subra, 2020. Elaboración propia.

El avance de las tarjetas en el tablero permite obtener algunas métricas que se emplean para la gestión del proyecto. Una de las métricas es el tiempo de entrega, que equivale al tiempo total que se tarda en entregar una tarea, medido desde el momento en el que se agrega al tablero hasta el momento en el que se completa. Otras son el tiempo de ciclo, como el tiempo requerido para procesar una tarea y el tiempo de respuesta, tiempo que una tarea espera hasta que se comienza a trabajar en él (PMI, 2017).

El periodo de tiempo dedicado a cada tarea en Scrum se limita a la duración del *Sprint*. En el caso de tareas de mayor duración, se intentan dividir en otras más pequeñas. Con Scrum, la priorización de tareas se realiza al planificar el *Sprint*. Como al inicio del *Sprint* se hacen planes detallados y estimaciones de esfuerzo (*Sprint Planning*), en Scrum no se permite ningún cambio en el plan de trabajo cuando el *Sprint* se está ejecutando. El orden de las tarjetas de la Figura 3 es el que se mantendrá durante todo el *Sprint* (Brezočnik & Majer, 2016).

#### 4.2 El tablero Kanban (*Kanban Board*)

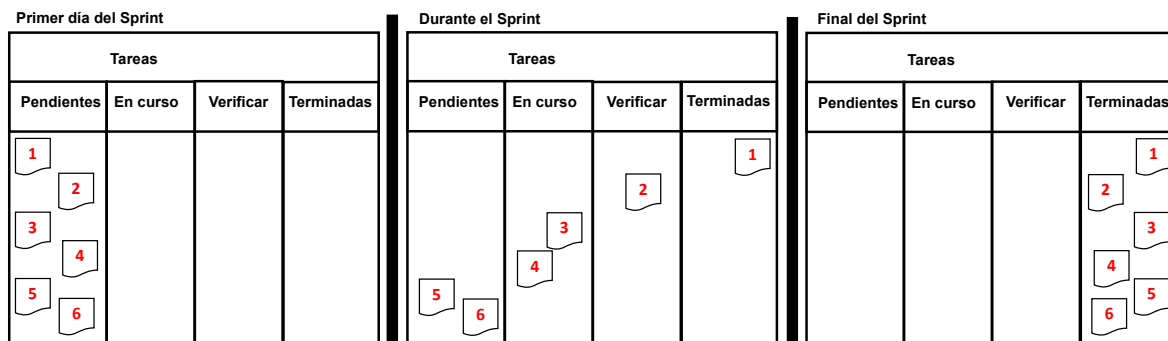
Como paso previo a la creación del tablero Kanban se ha de realizar el mapa del flujo de trabajo, definiendo la secuencia de etapas que sigue cada tarea, el tiempo que se invierte en cada etapa y los criterios seguidos para avanzar entre ellas. El tablero debe disponer de tantas columnas como etapas se hayan definido, lo que hace que no existan dos tableros Kanban iguales (Lasa, Álvarez & de las Heras, 2017; Mircea, 2019).

La primera columna se destina a los elementos del *Backlog* o a las tareas a realizar. Se recomienda que cada columna que sigue al *Backlog* se divida en dos. Una de ellas para los elementos “En curso” y la otra para “Hecho”. La columna “Hecho” realiza la función de buffer de transferencia e informa al equipo de que la tarjeta se puede extraer tan pronto como un recurso adecuado esté disponible (Ladas, 2008b; Edge, 2018). En cada etapa se indica la capacidad máxima del proceso (limitando el WIP) y, en ocasiones, puede ser de utilidad dividir

el tablero con una línea horizontal, dejando la parte superior para tareas urgentes que no puedan esperar en el *Backlog* (Guérin, 2018; Mircea, 2019; Subra, 2020) (Figura 4).

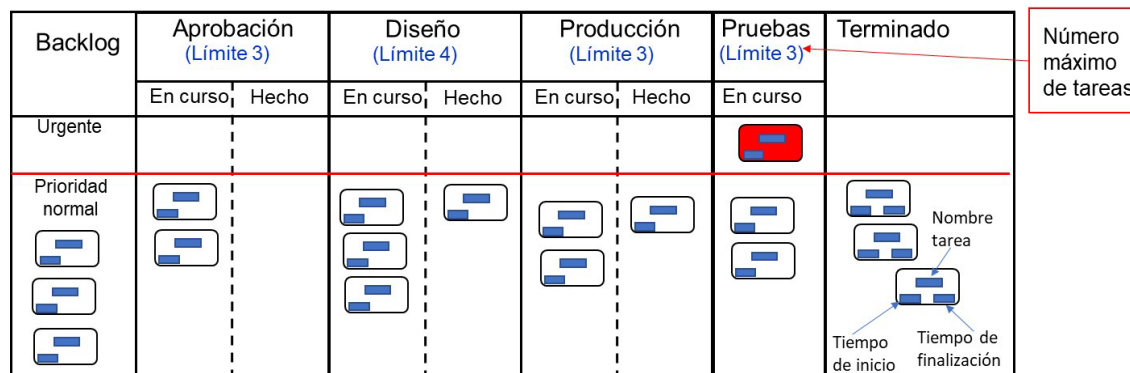
Kanban no utiliza una rutina de planificación precisa. Se suele realizar una planificación bajo demanda, cuando la sección de trabajos pendientes se queda sin elementos. Tampoco tiene iteraciones predefinidas (Kniberg & Skarin, 2010).

**Figura 3: Avance de las tarjetas en el tablero Scrum durante un *Sprint***



Fuente: Brezočnik & Majer, 2016. Elaboración propia.

**Figura 4: Ejemplo de tablero Kanban**



Fuentes: Lasa, Álvarez & de las Heras, 2017; Mircea, 2019; Subra, 2020. Elaboración propia.

La priorización de tareas con Kanban se realiza diariamente con la planificación justo a tiempo y el principio de extracción. Cada vez que se introduce una nueva tarea en el flujo de trabajo, debe tener la máxima prioridad para el equipo. En Kanban, el tablero normalmente es algo persistente, no se necesario limpiarlo y volver a empezar (Kniberg & Skarin, 2010; Brezočnik & Majer, 2016). Las tarjetas, además del nombre de la tarea, contendrán información adicional, como el responsable de la misma, una estimación del esfuerzo de trabajo o duración, prioridad (algún tipo de color significativo (p.e. rojo para las tareas urgentes)) y/o fechas de comienzo y finalización (Williams, 2015).

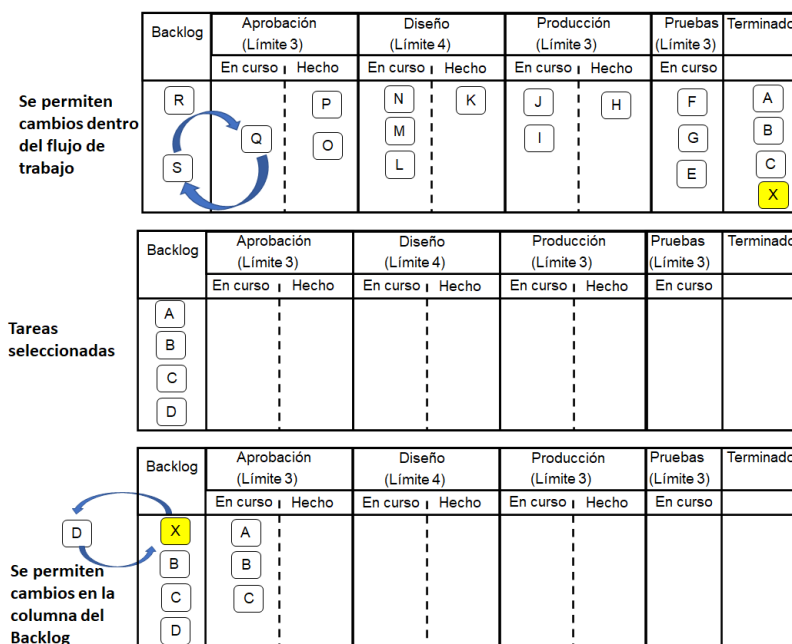
Kanban no proporciona reglas que prohíban cambios en el plan de trabajo en un momento dado. Las tareas en estado "En curso" se pueden reemplazar fácilmente por otras nuevas del

*Backlog*, también las tareas que ya están en proceso se pueden devolver al *Backlog* y se pueden extraer tareas más importantes (Brezočnik & Majer, 2016) (Figura 5).

### 4.3 El tablero Scrumban (*Scrumban Board*)

Las columnas en el tablero representan las etapas del flujo de trabajo por las que pasa cada tarea hasta su finalización (Reddy, 2015). Se emplean las columnas “Hecho” como indicador de solicitud de extracción, informando al equipo que la tarjeta de trabajo cumple la especificación completa del trabajo en esa etapa (Ladas, 2008b). En este aspecto, Scrumban es similar a Kanban en la extracción del trabajo de acuerdo a las necesidades (Albarqui & Qureshi, 2018).

Figura 5: Cambios en el plan de trabajo en Kanban



Nota: Adaptado de Brezočnik & Majer, 2016. Elaboración propia.

En la parte inferior de las columnas se propone añadir una descripción de las operaciones a realizar en cada fase, los recursos que necesita el equipo y la definición de hecho (*Definition of Done*) (Figura 6) (Ladas, 2008b; Reddy, 2015).

En el tablero empleado en Scrumban se mantiene el límite de WIP en cada columna, como en el tablero Kanban, para evitar cuellos de botellas en todas las etapas e informar, cuando se alcance el límite, que existe algún obstáculo en la etapa bloqueando todo el flujo (Ladas, 2008b; Sienkiewicz, 2012). Ladas (2008b) propone incluir el límite del WIP de la iteración en la columna del *Backlog*, equivalente al número de tareas promedio que se puedan lanzar por iteración. Esto supondría una mejora al reducir el tiempo dedicado a la estimación del trabajo en las reuniones de planificación.

Tras la columna del *Backlog*, se puede añadir la columna de “Preparado” (*Ready*), con las tareas extraídas del *Backlog* priorizadas para su ejecución. También se puede indicar la



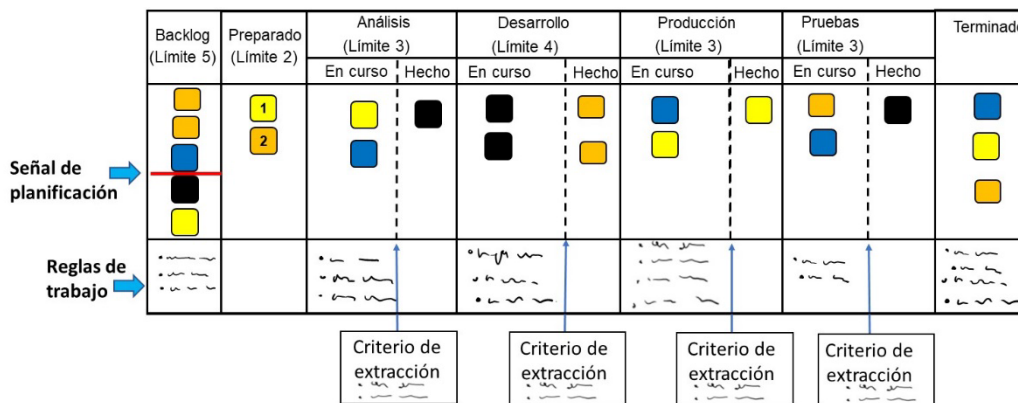
“Señal de planificación” en el *Backlog*, de manera que cuando no queden tareas bajo esta línea se prevea una reunión de planificación (Development That Pays, 2020).

En Scrumban primero se prioriza el trabajo con la planificación del tamaño del *Backlog*, posteriormente las tareas se definen y priorizan para cada iteración y, finalmente, a diario, como con Kanban (Brezočnik & Majec, 2016).

La información que se incorpora a la tarjeta varía de un equipo a otro. El tipo de trabajo se puede representar por el color de la tarjeta. También se puede emplear un código de colores para distinguir las clases de servicio o incluir una política de los límites de WIP para cada tipo de tarjeta en el tablero. Por ejemplo: 10 Tarjetas doradas (50% de capacidad), 4 tarjetas naranjas (20% de capacidad), 5 tarjetas azules (25% de capacidad) y 1 tarjeta negra (5% de capacidad) (Reddy, 2015).

En Scrumban también se emplean bloqueadores. Estos elementos, visualizados por un adhesivo o un imán adicional sobre la tarjeta, indican cuándo se suspende la labor en un elemento debido a la dependencia con otros trabajos. Advierten al equipo que en estas tareas existe algún tipo de impedimento que se ha de resolver (Reddy, 2015).

**Figura 6: Ejemplo de tablero Scrumban**



Fuentes: Ladas, 2008b; Reddy, 2015; Development That Pays, 2020. Elaboración propia.

## 5. Conclusiones

La evolución del tablero producida en cada metodología ha estado asociada a una mejora de la visualización del avance del trabajo con el fin de facilitar el seguimiento y control de las tareas asociadas al proyecto. En todos los casos, el tablero ayuda a la coordinación del equipo y al trabajo colaborativo, posibilitando su autoorganización, además de proporcionar una información visual del estado de las tareas.

Se han encontrado diferencias en la configuración del tablero y la forma de utilizarlo entre Scrum y Kanban, aunque presenta más similitudes que diferencias entre Kanban y Scrumban. Cada tablero, y la forma en que se emplea, se adecua a los principios que rigen las metodologías ágiles tratadas.

La evolución del empleo del tablero de Scrum, Kanban a Scrumban ha sido la de conseguir un flujo continuo en la ejecución de las tareas del proyecto, la detección rápida de problemas y cuellos de botella, así como la implementación de estrategias de mejora continua al proceso.

Como características reseñables del tablero resaltar su flexibilidad y adaptabilidad a las necesidades de cada metodología, así como a los requerimientos de que cada equipo. Además, resulta una herramienta fundamental para determinar las métricas de control y

seguimiento que las metodologías contemplan, recogiendo la información necesaria en las tarjetas, adecuando su formato (tamaño, color, etc.), según el criterio de los equipos.

## 6. Referencias

- Al-Aidaros, H. & Omar, M. (2017). Software project management approaches for monitoring work-in-progress: A review. *Journal of Engineering and Applied Sciences*, 12 (15), 3851-3857.
- Albarqi, A. A., & Qureshi, R. (2018). The proposed L-Scrum methodology to improve the efficiency of agile software development. *I.J. Information Engineering and Electronic Business*, 3, 23-35.
- Alqudah, M., & Razali, R. (2018). An empirical study of Scrum formation based on the selection of Scrum and Kanban practices. *International Journal on Advanced Science Engineering Information Technology*, 8 (6), 2315-2322. doi: <http://dx.doi.org/10.18517/ijaseit.8.6.6566>
- Anderson, D. J. (2010). *Kanban. Cambio evolutivo exitoso para su negocio de tecnología*. Sequim, WA, EEUU: Blue Hole Press.
- Arbeláez, O., Medina, F. A., & Chaves, J. A. (2011). Herramientas para el desarrollo rápido de aplicaciones web. *Scientia et Technica*, 17 (47), 254-258.
- Banijamali, A., Dawadi R., Ahmad, M. O., Similä, J., Oivo, M., & Liukkunen, K. (2017) Empirical investigation of Scrum in global software development. En Hammoudi S., Pires L., Selic B., Desfray P. (Eds) *Model-Driven Engineering and Software Development. MODELSWARD 2016. Communications in Computer and Information Science*, vol 692 (pp. 229-248). Cham: Springer.
- Bhavsar, K., Shah, V., & Gopalan, S. (2020). Scrumbanfall: An agile integration of Scrum and Kanban with waterfall in software engineering. *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, 9 (4), 2075-2084.
- Brezočnik, L., & Majer, Č. (2016). Comparison of agile methods: Scrum, Kanban, and Scrumban. En *Proc. 19th Int. Multiconference Inf. Soc.* Vol. 100 (pp. 1-5).
- Day, R.A. (2005). *Cómo escribir y publicar trabajos científicos (3ª ed)*. Washington, D.C., EEUU: The Oryx Press.
- Deemer, P., Benefield, G., Larman, C., & Vodde, B. (2012). *Una introducción básica a la teoría y práctica de Scrum. Versión 2.0*. InfoQueue. Enterprise Software Development Series.
- Development That Pays (2020). *The Scrumban cheat sheet – Scrum to Kanban in 7 stages*. Obtenido el 2 de febrero de 2022, desde <https://pages.developmentthatpays.com/cheatsheets/scrum-to-scrumban>
- Edge, J. (2018). *Agile: Una guía esencial sobre la gestión de proyectos Agile, el proceso Kanban y el Lean Thinking + Una guía completa sobre Scrum (Spanish Edition)*. Edición de Kindle.
- Fuior, F. (2019). Key elements for the success of the most popular Agile methods. *Romanian Journal of Information Technology and Automatic Control*, 29 (4), 7-16.
- Gaborov, M., Karuović, D., Radosav, D., Kavalić, M., Milosavljev, D., & Stanisavljev, S. (2021). The difference between Scrum and other agile and traditional methodologies. En Csanák, E. (Ed.) *8th International Joint Conference on Environmental and Light Industry Technologies (IJCELIT-2021)* (pp. 29-38). Budapest, Hungary: Rejtő Sándor Faculty of Light Industry and Environmental Engineering, Óbuda University.
- Gálvez, A. (2002). Revisión bibliográfica: usos y utilidades. *Matronas Profesión*, (10), 25-31.
- García, L. A., Oliveira Jr, E., Leal, G. C. L., & Morandini, M. (2020). A unified feature model for Scrum artifacts from a literature and practice perspective. En *Anais da IV Escola Regional de Engenharia de Software* (pp. 296-305). SBC.
- Gómez-Luna, E., Fernando-Navas, D., Aponte-Mayor, G., & Betancourt-Buitrago, L. A. (2014). Metodología para la revisión bibliográfica y la gestión de información de temas científicos, a través de su estructuración y sistematización. *Dyna*, 81 (184), 158-163.

- Gonçalves, L. (2018). Scrum. The methodology to become more agile. *Controlling & Management Review*, 4, 40-42.
- Grotenfelt, J. (2021). *Agile software development and implementation of Scrumban*. Bachelor's Thesis, Metropolia University of Applied Sciences, Vantaa, Finlandia.
- Guérin, B. (2018). *Gestión de proyectos informáticos. Desarrollo, análisis y control (3ª ed)*. Barcelona: Ediciones ENI.
- Guirao-Goris, J. A., Olmedo, A., & Ferrer, E. (2008). El artículo de revisión. *Revista Iberoamericana de Enfermería Comunitaria*, 1 (1), 1-25.
- Kniberg, H., & Skarin, M. (2010). *Kanban y Scrum. Obteniendo lo mejor de ambos*. C4Media, editores de InfoQ.com.
- Ladas, C. (2008a). *Scrumban: Essays on Kanban Systems for Lean Software Development*. Seattle, WA, EEUU: Modus Cooperandi Press, Edición de Kindle.
- Ladas, C. (2008b). *Scrumban*. Obtenido el 14 de enero de 2022, desde <https://www.agilealliance.org/scrumban/>
- Lasa, C., Álvarez, A., & de las Heras, R. (2017). *Métodos ágiles. Scrum, Kanban, Lean*. Madrid: Ediciones Anaya Multimedia.
- Lei, H., Ganjezadeh, F., Jayachandran, P. K., & Ozcan, P. (2017). A statistical análisis of the effects of Scrum and Kanban on software development projects. *Robotics and Computer-Integrated Manufacturing*, 43, 58-67.
- Lledó, P. (2014). *Gestión Lean y Ágil de proyectos. Lean Project Management*. Estados Unidos: Trafford Publishing.
- Mahnic, V. (2014). Improving software development through combination of Scrum and Kanban. *Recent Advances in Computer Engineering, Communications and Information Technology*, 281-288.
- Mircea, E. (2019). Project management using agile frameworks. *Economy Informatics*, 19 (1), 34-44.
- Molina, J. R., Honores, J. A., Pedreira-Souto, N., & Pardo, H. P. (2021). Comparativa de metodologías de desarrollo de aplicaciones móviles. *3C Tecnología. Glosas de innovación aplicadas a la pyme*, 10 (2), 73-93. Doi: <http://dx.doi.org/10.17993/3ctecno/2021.v10n2e38.73-93>
- Morales, R. (2020). *Gestión de tareas con Kanban. Una introducción a la gestión visual del trabajo (2ª ed)*. Torrazza Piemonte (TO), Italia: El autor.
- Nikitina, N., Kajko-Mattsson, M., & Stråle, M. (2012). From Scrum to Scrumban: A Case Study of a Process Transition. En *2012 International Conference on Software and System Process (ICSSP)* (pp. 140-149). IEEE.
- Orszewski, R. (2018). *¿Cómo le ayudan las cadencias Kanban a alinear la comunicación empresarial?* Obtenido el 18 de agosto de 2021, desde <https://kanbanize.com/blog/kanban-cadences/>
- Patil, S. P., & Neve, J. R. (2018). Productivity improvement of software development process through Scrumban: A practitioner's approach. En *2018 International Conference on Advances in Communication and Computing Technology (ICACCT)* (pp. 314-318). IEEE.
- Permana, P. A. G. (2015). Scrum method implementation in a software development project management. *International Journal of Advanced Computer Science and Applications*, 6 (9), 198-204.
- Project Management Institute (PMI) (2017). *Guía práctica de Ágil*. Pennsylvania, EEUU: Project Management Institute.
- Reddy, A. (2015). *The Scrumban [r]evolution: getting the most out of Agile, Scrum, and lean Kanban*. Crawfordsville, Indiana, United States: Addison-Wesley Professional. Pearson Education.
- Schwaber, K., & Sutherland, J. (2020). *La guía Scrum. La guía definitiva de Scrum: Las Reglas del Juego*. Obtenido el 10 de agosto de 2021, desde <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Spanish-European.pdf>

- Sienkiewicz, Ł. D. (2012). Scrumban – the Kanban as an addition to Scrum software development method in a network organization. *Informatyka Ekonomiczna Business Informatics*, 2 (24), 73-81.
- Stoica, M., Ghilic-Micu, B., Mircea, M., & Uscatu, C. (2016). Analyzing agile development – from waterfall style to Scrumban. *Informatica Economică*, 20 (4), 5-14. Doi: <http://dx.doi.org/10.12948/issn14531305/20.4.2016.01>
- Subra, J. (2020). *Scrum. Un método ágil para sus proyectos*. Cornellá de Llobregat, España: Ediciones ENI.
- Vilanova, J. C. (2012). Revisión bibliográfica del tema de estudio de un proyecto de investigación. *Radiología*, 54 (2), 108-114. Doi: <http://dx.doi.org/10.1016/j.rx.2011.05.015>
- Williams, P. R. (2015). *Visual project management*. De Pere, Wisconsin, EEUU: Think for Change Publishing.