



UNIVERSIDAD DISTRITAL
FRANCISCO JOSÉ DE CALDAS

Scrum

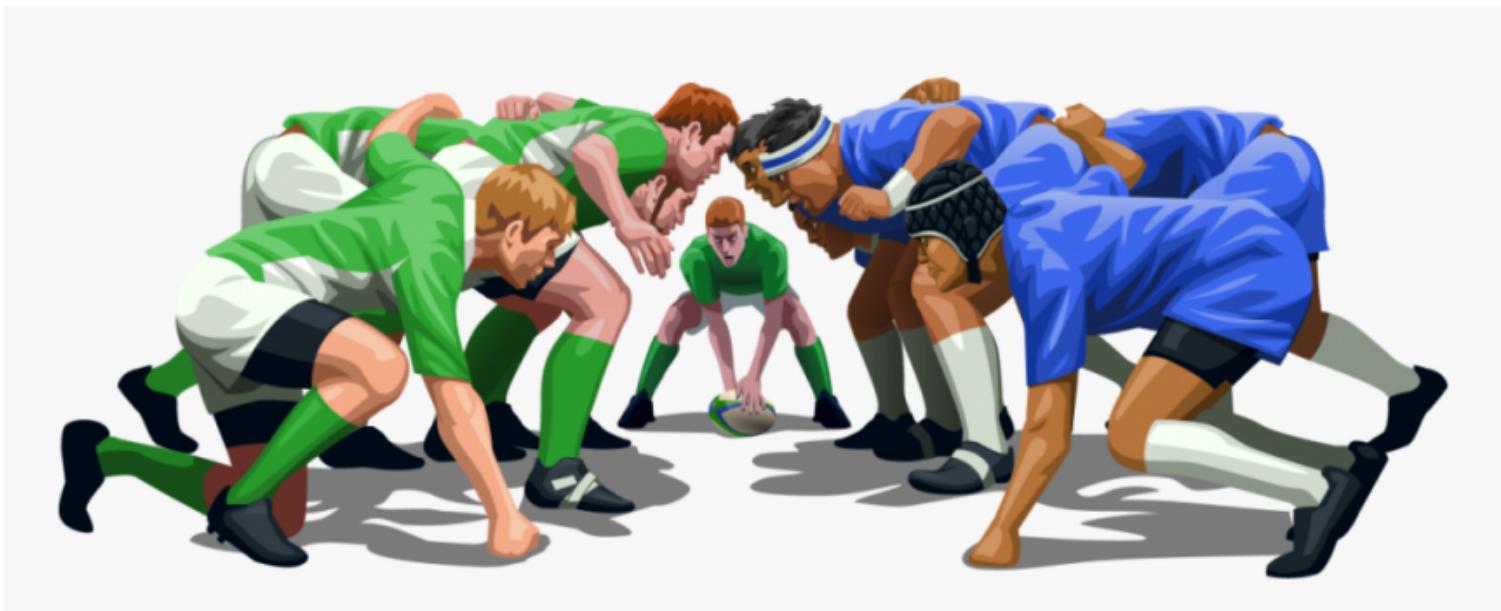
Contenidos

- Introducción y objetivos
- Orígenes y filosofía de Scrum
- Valores de Scrum
- Flujo del proceso Scrum
- Los roles de Scrum
- Los artefactos de Scrum
- Los eventos de Scrum
- Las historias de usuario
- Extensiones de Scrum
- Estimaciones ágiles
- Escalado de Scrum

Contenidos

- Referencias bibliográficas

1. Introducción



Scrum, sigue los principios propuestos en el **Manifiesto ágil**, y los concreta en una serie de indicaciones metodológicas relacionadas con la definición de roles específicos, reuniones y artefactos, que soportan el proceso de desarrollo y permiten guiar el proyecto utilizando una documentación muy ligera a la vez que efectiva.

2. Orígenes y filosofía de Scrum

- Scrum es una metodología de desarrollo de software desarrollada por Jeff Sutherland a comienzos de la década de 1990.
- El nombre proviene de una formación propia del rugby, en la que dos grupos de jugadores, de equipos diferentes, se colocan alrededor del balón y en oposición violenta para ganar el control de este y desplazarlo alrededor del campo.
- Scrum no es propiamente una metodología, sino más bien ***un marco general de trabajo*** para el desarrollo de productos complejos, que se centra en la **administración iterativa** del proceso de desarrollo, y no tanto en prácticas específicas.
- No prescribe actividades concretas ni ofrece una especificación detallada de la manera en que deben organizarse las tareas de desarrollo, sino que proporciona un contexto en el que puedan ser los propios involucrados quienes creen su propio proceso.

2. Orígenes y filosofía de Scrum

- Scrum es especialmente apropiada para combinarla con prácticas y herramientas procedentes de otras metodologías.
- Scrum tiene una fundamentación eminentemente empírica, defendiendo que cualquier decisión debe tomarse a partir del conocimiento acumulado por la experiencia.

2. Orígenes y filosofía de Scrum

Para poder establecer un control del proceso empírico, Scrum se asienta sobre tres pilares (Schwaber y Sutherland, 2017):

TRANSPARENCIA	INSPECCIÓN	ADAPTACIÓN
<p>La información significativa sobre los objetivos, alcance y características del proyecto debe ser accesible y visible para todos los participantes.</p>	<p>Los participantes deben revisar con frecuencia los artefactos del proceso y la información disponible para detectar posibles desviaciones.</p>	<p>Tan pronto como se detecte alguna desviación respecto de los objetivos previstos, se deben tomar medidas correctoras.</p>

3. Valores de Scrum

Scrum se adhiere a los cuatro valores del «Manifiesto ágil». Así son contemplados por esta metodología (Alaimo, 2013).

VALORES DEL «MANIFIESTO ÁGIL» EN SCRUM

Valoración de los individuos y sus interacciones sobre procesos y herramientas

En Scrum los equipos toman el control y la responsabilidad del desarrollo, resuelven los problemas que encuentran —con la ayuda del *Scrum Master*— y se comunican a través de una serie de reuniones de carácter generalmente informal.

3. Valores de Scrum

VALORES DEL «MANIFIESTO ÁGIL» EN SCRUM

Valoración del software funcional sobre la documentación exhaustiva

Al final de cada iteración o *Sprint* se entrega al cliente una nueva versión del producto lista para su utilización. La documentación es una herramienta intermedia que sirve de apoyo al proceso de desarrollo, y no constituye en sí misma un producto entregable.

3. Valores de Scrum

VALORES DEL «MANIFIESTO ÁGIL» EN SCRUM

Valoración de la colaboración con el cliente sobre la negociación contractual

Scrum adopta la figura del *Product Owner* como representante del cliente y otros *stakeholders*. Forma parte del equipo Scrum (no del equipo de desarrollo) y es responsable, en colaboración con el equipo de desarrollo, de que las iteraciones se orienten hacia la maximización del valor aportado a los clientes o usuarios finales.

3. Valores de Scrum

VALORES DEL «MANIFIESTO ÁGIL» EN SCRUM

Valoración de la adaptación al cambio frente al seguimiento de un plan

El enfoque de las reuniones y artefactos empleados facilita que todos los participantes tengan información actualizada sobre las características y alcance del proyecto. De esta manera, todos están capacitados para tomar decisiones e introducir modificaciones, y la información para detectar cualquier variación en el entorno se obtiene gracias a la participación del *Product Owner* y la continua liberación de código funcional.

3. Valores de Scrum

Estos valores generales son acompañados y reforzados por una serie de **valores sobre los que se apoya** la metodología (Alaimo, 2013; Schwaber y Sutherland, 2017)

COMPROMISO

De las responsabilidades y libertad otorgadas a los miembros del equipo en la toma de decisiones sobre el proyecto, surge su compromiso para sacar adelante el trabajo y cumplir los objetivos de cada iteración.

3. Valores de Scrum

CORAJE

Los miembros del equipo se apoyan entre sí para afrontar retos complejos y vencer obstáculos a lo largo de cada *sprint*, con el objetivo común de alcanzar las metas planteadas en cada iteración.

FOCALIZACIÓN

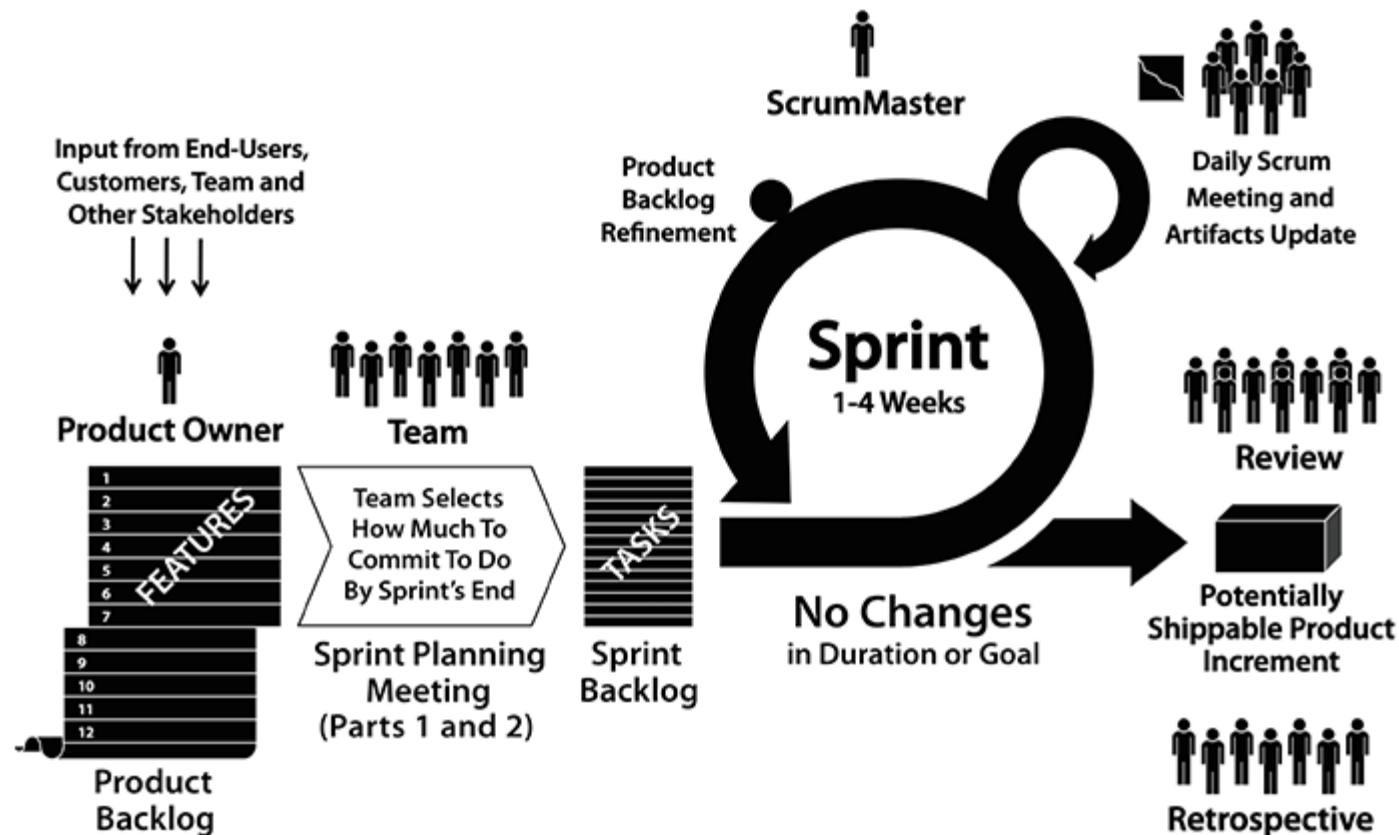
En cada iteración el equipo se centra en un conjunto concreto y acotado de características en cada *sprint*.

3. Valores de Scrum

APERTURA

Durante el desarrollo del proyecto se fomenta la transparencia y el flujo de información, de manera que todos los miembros del equipo tengan acceso permanente a las características del producto y del proceso.

4. Flujo del proceso Scrum



4. Flujo del proceso Scrum

Roles:

- Scrum Master.
- Product Owner.
- Equipo.

Reuniones:

- Planificación de sprint.
- Scrum diario.
- Revisión y retrospectiva del sprint.

Artefactos:

- Pila del producto.
- Pila del sprint.

4. Flujo del proceso Scrum

Scrum es un **proceso iterativo** con comunicación y retroalimentación constantes.

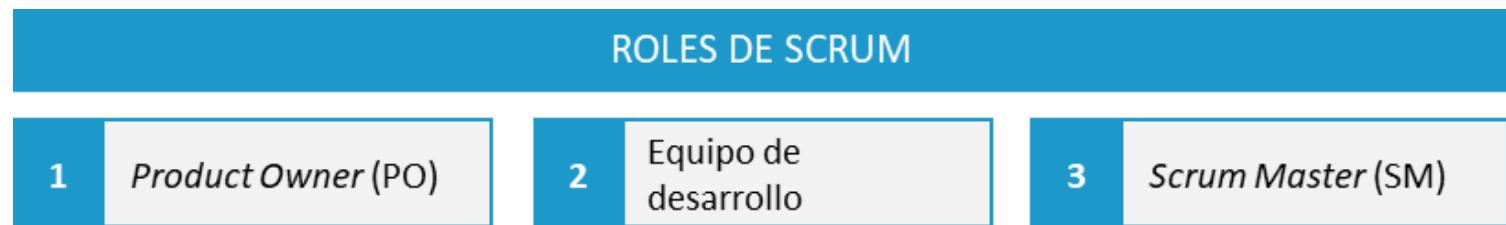
La definición de la «**pila del sprint**» durante la reunión de «**planificación**» marca el comienzo de cada iteración, y el equipo de desarrollo se compromete con el Product Owner en la entrega de un conjunto de funcionalidades identificadas en este documento.

Durante cada sprint, se realiza al comienzo de cada jornada una reunión donde los miembros del equipo de desarrollo comparten información sobre el progreso de la iteración.

Al final de cada sprint se muestra al Product Owner el incremento realizado, y el equipo reflexiona sobre la práctica realizada, identificando posibilidades de mejora sobre el proceso.

5. Los roles de Scrum

Los equipos de Scrum son autorganizados y multifuncionales y están compuestos por tres roles principales (Schwaber y Sutherland, 2017):



El Product Owner (PO)

Es el **responsable último del éxito del producto**, que debe ajustarse a las necesidades de cliente y usuarios finales. Tiene la visión general del producto, define los requisitos y comprende los puntos de vista del cliente y otros stakeholders. Su principal misión es maximizar la rentabilidad del producto o retorno de la inversión —Return of Investment (ROI)— y para ello debe considerar en primer lugar la priorización de las funcionalidades deseadas.

5. Los roles de Scrum

El Product Owner (PO)

Sus principales responsabilidades son:

- **Comprende los objetivos del producto** y se encarga de maximizar el valor aportado en el desarrollo en cada iteración.
- **Se comunica con el cliente, usuarios y stakeholders** para obtener información y representar sus intereses en el proyecto.
- **Se comunica con el equipo de desarrollo** para asegurarse de que todos comprenden los requisitos con el detalle suficiente.
- **Define los requisitos del producto y mantiene la pila de producto**, que va adaptando en cuanto a número de requisitos y prioridades relativas según avanza el proyecto. Para ello cuenta con la ayuda del Scrum Master.

5. Los roles de Scrum

El Product Owner (PO)

Sus **principales responsabilidades** son:

- **Determina el alcance de cada iteración** en colaboración con el equipo de desarrollo, priorizando los requisitos según una visión de negocio.
- **Acepta (o rechaza) la versión entregada al final de cada sprint** y suministra feedback al resto del equipo.

5. Los roles de Scrum

El equipo de desarrollo

Es el **conjunto de profesionales que se responsabiliza de cumplir los objetivos de desarrollo** planteados en cada iteración, y de entregar al final de cada sprint una versión funcional que pueda ser mostrada en el **Sprint Review (SR)**, y deben tener las siguientes **características**:

- Son preferiblemente **de tamaño reducido**, nunca más de nueve personas
- **Ser autorganizados**, y solo ellos, sin intervención de ningún otro rol, determinan la manera de planificar el proceso de desarrollo en cada iteración y la descomposición de funcionalidades en tareas o su asignación a desarrolladores concretos.

5. Los roles de Scrum

El equipo de desarrollo

- **Ser multifuncionales**, englobando perfiles heterogéneos que cubren todas las necesidades del proyecto y de su implementación. Se espera que cada individuo tenga un perfil generalista con capacidades amplias, no siendo especialista en un ámbito exclusivo, y teniendo así la capacidad de colaborar en todo lo que esté a su alcance.
- **No se reconocen títulos específicos ni subequipos** (Schwaber y Sutherland, 2017), independientemente del área de experiencia concreta de cada miembro, de manera que todos son homogéneos desde el punto de vista de la responsabilidad que asumen en el proceso.

5. Los roles de Scrum

El equipo de desarrollo

El tamaño del equipo de desarrollo

El número de desarrolladores debe ser lo bastante pequeño como para que su trabajo resulte **ágil**, pero lo bastante grande como para ser capaces de **producir incrementos significativos** en cada iteración.

- ▶ **Equipos de menos de tres personas** reducen la variedad de habilidades esperadas y la posible comunicación y flujo de información.
- ▶ **Equipos de más de 9 personas** requieren demasiado esfuerzo de coordinación y dificultan que un proceso empírico resulte de utilidad.

5. Los roles de Scrum

El equipo de desarrollo

Además, el equipo de desarrollo tiene **tres responsabilidades fundamentales** (Alaimo, 2013):

- **Realizar estimaciones** sobre el esfuerzo de implementación de cada funcionalidad, que comparten con el Product Owner.
- **Comprometerse con la entrega** de las características acordadas en cada sprint contando con el tiempo y recursos disponibles.
- **Realizar la demostración del producto** en el momento de entrega al final de cada sprint.

5. Los roles de Scrum

El Scrum Master (SM)

El Scrum Master adopta una **función de coaching** dentro del equipo, y es el encargado de ayudar al equipo de desarrollo y al Product Owner a seguir la metodología de manera correcta, resolviendo cualquier impedimento que pueda surgir.

No se debe confundir su figura con la de un jefe de proyecto, pues no está encargado de planificar ni de asignar tareas. Lo ideal es que exista un Scrum Master con dedicación completa para el equipo, aunque en proyectos pequeños puede desarrollar esta labor algún miembro del propio equipo de desarrollo (Sutherland y Schwaber, 2012).

Existen diferentes **guías y listas de comprobación** que pueden ayudar al Scrum Master a desarrollar sus tareas de manera eficiente (CollabNet, 2012; James, 2012; Kniberg, 2010). Estos documentos pueden ayudar a este rol a realizar un correcto seguimiento de las actividades del equipo y a detectar posibles obstáculos.

6. Los artefactos de Scrum

El proceso de Scrum propone un número de elementos que permiten formalizar y controlar el proceso de desarrollo, de una manera ágil y ligera.

Están diseñados para **maximizar la transparencia de la información clave** (Schwaber y Sutherland, 2017).

Es misión del **Scrum Master** asegurar que esta transparencia se da en la práctica, proponiendo acciones correctivas o prácticas específicas en caso de ser necesario.

En ningún caso es recomendable que el Scrum Master y el Product Owner sean la misma persona.

- ➊ Pila del producto (Product Backlog).
- ➋ Pila del sprint (Sprint Backlog).

6. Los artefactos de Scrum

La pila del producto

La pila del producto (***Product Backlog (PB)***) es el **documento fundamental que almacena los requisitos del proyecto**, como una lista ordenada y priorizada de funcionalidades que mantiene el Product Owner.

Es un **documento vivo**, que evoluciona a medida que el desarrollo avanza y el entorno cambia. Puede haber modificaciones en requisitos existentes debidas a cambios en el negocio, el mercado o la propia tecnología, y su posible eliminación o aparición de nuevos requisitos.

Cada uno de los elementos de esta lista es un ***Product Backlog Item (PBI)*** y contiene una descripción, un orden y, preferiblemente, también una estimación de complejidad y de valor respecto del producto final. Muchas veces se incluyen también descripciones de las pruebas que demostrarán su correcta implementación.

En el caso de que trabajen varios equipos en el mismo producto, se mantiene un único PB, pudiéndose emplear un atributo adicional que indica el equipo asignado.

6. Los artefactos de Scrum

La pila del producto

El refinamiento del Product Backlog

Consiste en la revisión continuada e iterativa, por parte del **Product Owner y del equipo de desarrollo** de los elementos del PB, en cuanto a definición, priorización y estimación de esfuerzos.

El *Product Owner* también puede introducir modificaciones de manera unilateral cuando lo considere necesario, atendiendo siempre a criterios de negocio.

En las revisiones de cada sprint, el *Product Owner* puede utilizar este documento para hacer un seguimiento del trabajo restante.

Esta estimación es aproximada y, desde un punto de vista empírico, solo es posible utilizar **la experiencia previa acumulada**, puesto que en entornos complejos **el futuro es a menudo incierto**.

6. Los artefactos de Scrum

La pila del sprint

La pila del sprint (*Sprint Backlog (SB)*) está formada por el conjunto de los PBI seleccionados al comienzo de cada sprint para su implementación.

Hace visible el esfuerzo requerido por parte del equipo de desarrollo para alcanzar el objetivo de la iteración y constituye el documento fundamental para su planificación. La responsabilidad de su mantenimiento y posible adaptación recae en el equipo de desarrollo.

Con el objetivo de facilitar la mejora continua, también se suele incluir algún elemento de mejora desde el punto de vista del proceso, identificado en la reunión de retrospectiva del anterior sprint.

Este documento permite hacer un seguimiento de la iteración comprobando la cantidad y duraciones estimadas de trabajos restantes. Este proceso de seguimiento se realiza en cada una de las reuniones diarias y permite al equipo gestionar su propio progreso.

7. Los eventos de Scrum

Para facilitar el control y el seguimiento del proceso, fomentar los flujos de comunicación y realizar las actividades de planificación necesarias en cada iteración, Scrum propone una serie de eventos con propósitos específicos.

Todos tienen una **duración máxima controlada**, del mismo modo que **la duración de cada sprint es constante**. Cada sprint actúa como un contenedor para el resto de los eventos, y cada evento es considerado una oportunidad de mejora y adaptación.

Los eventos predefinidos por la metodología son los siguientes (Schwaber y Sutherland, 2017):

- ➊ El sprint
- ➋ Planificación del sprint
- ➌ Scrum diario
- ➍ Revisión del sprint
- ➎ Retrospectiva del sprint

7. Los eventos de Scrum

El sprint

Según Schwaber y Sutherland (2017), es cada uno de los períodos de tiempo de duración fija, nunca superior al mes, durante el cual se crea **un incremento del producto «terminado, utilizable y potencialmente desplegable».**

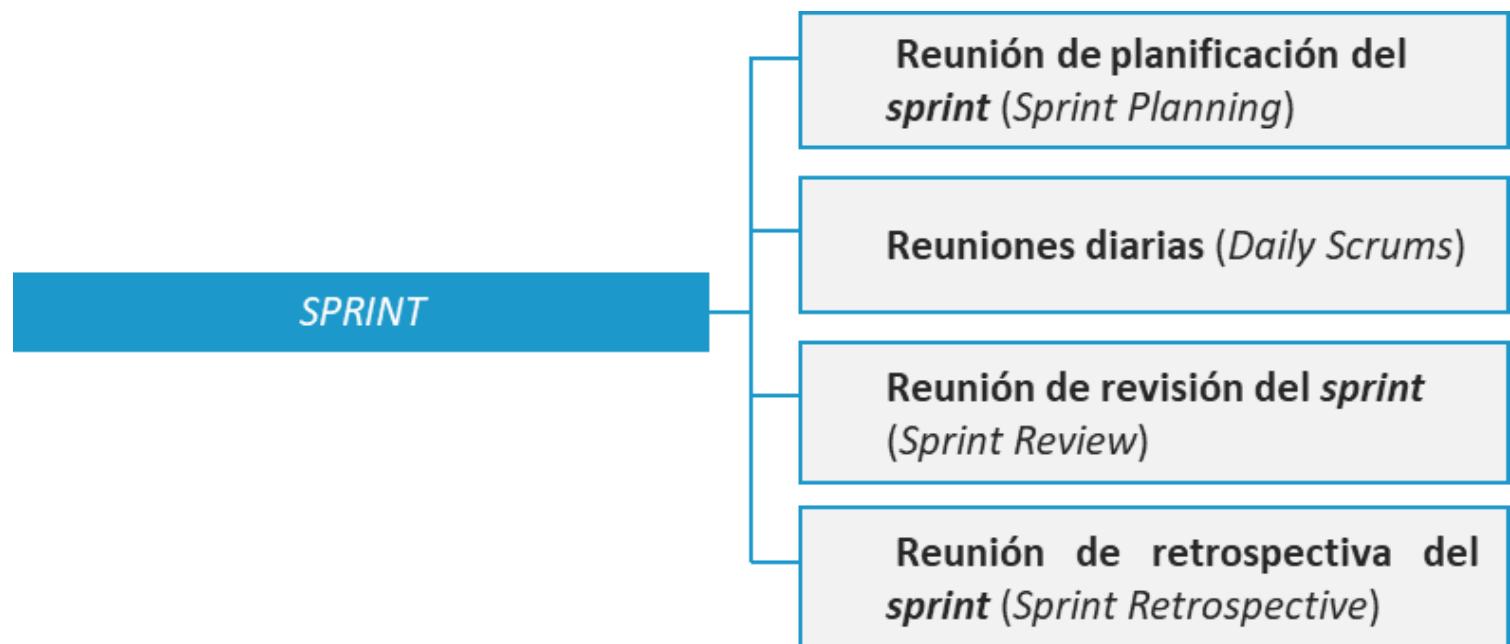
Durante un sprint **no se realizan cambios que puedan afectar el objetivo de ese sprint**, pero se pueden aclarar o renegociar aspectos relativos al alcance en colaboración con el PO.

En hecho de que su duración se mantenga constante ayuda a mejorar la **predictibilidad del proceso general**, aprendiendo de los objetivos alcanzados y problemas surgidos en iteraciones anteriores.

7. Los eventos de Scrum

El sprint

Cada uno de los *sprints* contiene cuatro **eventos fundamentales**:



7. Los eventos de Scrum

Planificación del sprint

La reunión de planificación de sprint (Sprint Planning) es **el punto de arranque de cada iteración**, se establecen los compromisos entre el **equipo de desarrollo** y el **Product Owner** sobre el alcance y características de la nueva entrega.

En esta reunión se debe responder a dos preguntas (Schwaber y Sutherland, 2017):

¿Qué puede hacerse en este sprint?

- Es la primera fase de la reunión. Se parte del Product Backlog, y el Product Owner propone los elementos que deberían formar parte del objetivo.
- El número y configuración final de los PBI seleccionados depende de la capacidad disponible y la experiencia acumulada en iteraciones previas, siendo algo que debe determinar finalmente el equipo de desarrollo.

7. Los eventos de Scrum

Planificación del sprint

¿Cómo se conseguirá alcanzar el objetivo del sprint?

- En esta segunda fase, el equipo de desarrollo analiza las funcionalidades seleccionadas y decide cómo construirá el nuevo incremento terminado.
- Los PBI se descomponen en tareas individuales, cada una de las cuales no debería tener una duración superior a un día, y los miembros del equipo se autoasignan estas tareas comprometiéndose con su desarrollo.
- El resultado de este trabajo, que puede incluir tareas de diseño previo, es la pila del sprint (***Sprint Backlog (SB)***).

7. Los eventos de Scrum

Scrum diario

Los scrum diarios (**Daily Scrums**) son reuniones diarias, al comienzo de la jornada, de una duración máxima de **quince minutos** en la que participa el equipo de desarrollo.

En ella se revisa el trabajo realizado desde la última reunión y se planifica el trabajo de la jornada.

Esta reunión es gestionada y dirigida por el propio **equipo de desarrollo**, y el *Scrum Master* se encarga de que se respeten los plazos y de que los flujos de comunicación sean los adecuados. Puede participar el PO para resolver dudas.

7. Los eventos de Scrum

Scrum diario

En estas reuniones, cada uno de los miembros del equipo de desarrollo debe responder a las siguientes preguntas:

TRABAJO REALIZADO



¿Qué tareas has completado desde la última reunión diaria para alcanzar el objetivo del *sprint*?

7. Los eventos de Scrum

Scrum diario

TRABAJO PLANIFICADO

¿Qué harás hoy para lograr el objetivo del *sprint*?

PROBLEMAS IDENTIFICADOS

¿Detectas algún impedimento que puede comprometer el logro del objetivo del *sprint*?

7. Los eventos de Scrum

Revisión del sprint

Cuando finaliza cada sprint se lleva a cabo la reunión de revisión del sprint (*Sprint Review*), que tiene por objetivo **demostrar el incremento y obtener retroalimentación** de los interesados en el proyecto.

Es fundamental la participación de todos los miembros del equipo en esta **reunión de carácter informal**, en la que se analiza el trabajo realizado y se proponen mejoras o modificaciones sobre la pila del producto de cara al futuro.

Además del *Product Owner*, pueden participar otros stakeholders invitados por él. El resultado de esta reunión es la aceptación (o el rechazo) de la nueva entrega, la decisión de poner la nueva versión en producción, y una pila del producto actualizada a partir del análisis realizado.

7. Los eventos de Scrum

Retrospectiva del sprint

Esta última reunión de cada iteración es de **carácter eminentemente técnico**, y en ella participa el equipo de desarrollo para analizar su propio trabajo y plantear posibilidades de mejora sobre los procesos.

El *Scrum Master* se encarga una vez más de controlar los tiempos y asegurarse de que el diálogo se desarrolla bajo un espíritu constructivo.

Tras analizar el último sprint, evaluando las personas, procesos y herramientas, se crea un plan de mejoras que se incluirá en el siguiente sprint.

8. Las historias de usuario

Un elemento fundamental en Scrum, como en cualquier otro proceso de desarrollo de software, son los requisitos. En el caso de Scrum los requisitos se materializan en **historias de usuario (*User Stories (US)*)**, que son uno de los elementos integrantes del Product Backlog (PB) que mantiene el Product Owner (PO).

En un sentido amplio, el **_Product Backlog** contiene **PBI (*Product Backlog Items*)**, que son las historias de usuario y cualquier otra característica asociada al sistema que no necesariamente esté asociada a una funcionalidad o beneficio concreto para el usuario (por ejemplo, requisitos no funcionales).

La mejor manera para enunciar las historias de usuario (y en general cualquier PBI) es la siguiente:

como rol quiero funcionalidad o evento para beneficio

8. Las historias de usuario

Ejemplos de historias de usuario

US-001

COMO: usuario de la tienda online
QUIERO: un botón para ordenar por precio los artículos
PARA: poder localizar los más económicos.

US-002

COMO: usuario de la tienda online
QUIERO: un listado de opciones de pago
PARA: poder elegir la más conveniente.

US-003

COMO: administrador de la tienda online
QUIERO: obtener un listado mensual de pedidos
PARA: hacer un seguimiento de las ventas.

8. Las historias de usuario

Ejemplo tabla de historias de usuario

ID	Historia	Descripción	Criterios de aceptación	Prioridad	Esfuerzo
HU1					
HU2					
HU3					

8. Las historias de usuario

Para que un **PBI** sea correcto es conveniente que siga la regla **INVEST** (Agile Alliance, 2015), que fue acuñada por Bill Wake, en 2003, ofreciendo una serie de criterios que permiten evaluar la calidad de los PBI (Wake, 2003).

- **Independent (independiente).** Los PBI deberían ser independientes entre sí y no presentar solapamientos conceptuales. Esto garantiza que su definición no impone un orden de implementación, y que las estimaciones sean más sencillas.
- **Negotiable (negociable).** La tarjeta que contiene cada historia ofrece una descripción general lo suficientemente detallada para que resulte comprensible. Los detalles deben ser negociados con el PO en tiempo de desarrollo (la historia captura la esencia, no los detalles).
- **Valuable (valiosa).** La historia debe resultar valiosa para el PO (y en última instancia, para el cliente y usuarios).

8. Las historias de usuario

- **Estimable (estimable).** Debe ser fácil para el equipo la estimación de esfuerzo asociado a cada historia. Las mayores dificultades para ello son la falta de conocimiento sobre el producto (lo cual requiere mayor negociación) o su excesivo tamaño (y en este caso debemos descomponer la historia).
- **Small (pequeña).** Idealmente cada historia debería ser implementada por dos o tres personas trabajando una semana.
- **Testable (testable o verifiable).** Como cualquier requisito, cada historia debería estar asociada a algún tipo de prueba que permita verificar que ha sido correctamente implementada. Reflexionar sobre las pruebas puede ayudar a comprender el verdadero alcance del requisito o, incluso, a identificar historias que no son necesarias. Los requisitos no funcionales también deberían ser verificables.

8. Las historias de usuario

Como sabemos, en cada iteración los **PBI** se descomponen en **tareas concretas**, más manejables, y que pueden ser abordadas por desarrolladores específicos en función de su experiencia.

Wake también ofrece algunas indicaciones sobre las características deseables de cada una de estas tareas, que deben ser en este caso **SMART**:

- ▶ **Specific (específica).** Debe ser enunciada en términos muy concretos y comprensibles, y garantizar que no tiene solapamientos con otras tareas.
- ▶ **Measurable (medible).** Debe ir asociada con algún tipo de métrica que permita garantizar que la tarea está completada.
- ▶ **Achievable (alcanzable).** Las tareas son asignadas a miembros concretos del equipo (o a parejas, en el caso de la programación en parejas). Completarlas debe suponer un objetivo realista teniendo en cuenta los recursos disponibles y la formación y experiencia del equipo.

8. Las historias de usuario

- ▶ **Relevant (relevante).** Aunque la descomposición en tareas es importante desde el punto de vista técnico, cada una de las tareas debe ser justificable a ojos del PO, representando algún tipo de contribución necesaria y significativa para el PBI.
- ▶ **Time-boxed (acotada en el tiempo).** La duración estimada de la tarea es un indicador de su complejidad, y permite al desarrollador asignado solicitar ayuda en caso de que observe que no cumplirá los plazos.

9. Extensiones de Scrum

Scrum ofrece un marco de trabajo general, define roles, artefactos y actividades que guían el proceso de desarrollo a través de iteraciones de duración constante.

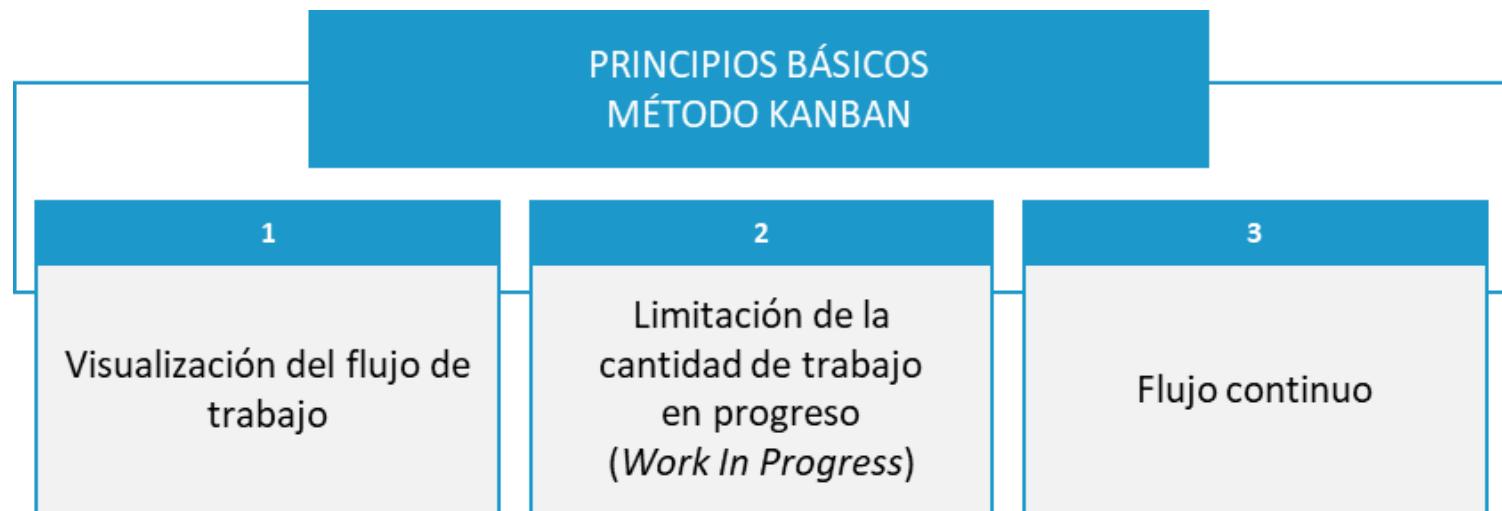
Este marco es apropiado para combinarse con elementos de otras metodologías, por ejemplo:

- Scrumban: combinación de Scrum y Kanban
- Combinación de Scrum y XP

9. Extensiones de Scrum

Combinación de Scrum y Kanban (Scrumban)

El método **Kanban** hace énfasis en la entrega continua de resultados, teniendo en cuenta que se deben evitar saturaciones y cuellos de botella según las capacidades del equipo de desarrollo. Se basa en tres principios (McLaughlin, 2018):



9. Extensiones de Scrum

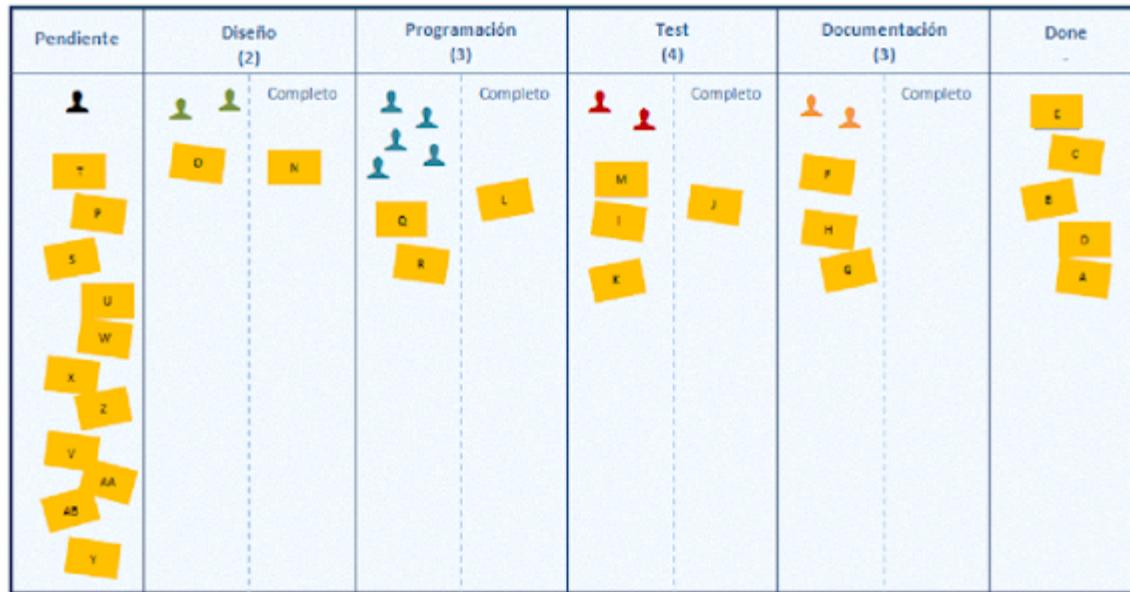
Combinación de Scrum y Kanban (Scrumban)

Visualización del flujo de trabajo.

- Ofrece una herramienta fundamental, el tablero Kanban, que permite obtener información visual sobre cada tarea y sus relaciones con un solo golpe de vista.
- Se descompone el trabajo en tareas que se desplazan de izquierda a derecha, y en columnas se muestra el estado del trabajo (pendiente, en progreso, finalizado, en la configuración más básica).

9. Extensiones de Scrum

Combinación de Scrum y Kanban (Scrumban)



9. Extensiones de Scrum

Combinación de Scrum y Kanban (Scrumban)

Limitación de la cantidad de trabajo en progreso (Work In Progress (WIP)).

- Busca equilibrar la carga de trabajo en cada momento, de manera que el equipo no quede saturado.
- En cada columna del tablero no se puede superar un volumen de trabajo prefijado.

Flujo continuo.

- Cuando una tarea se finaliza, la siguiente más prioritaria se pone de inmediato en desarrollo.
- No se trabaja por lotes de trabajo, como en Scrum con sus sprints, sino que todas las tareas pendientes son susceptibles de ser comenzadas.

9. Extensiones de Scrum

Combinación de Scrum y Kanban (**Scrumban**)

La manera más simple de combinar ambas metodologías consiste en aprovechar las capacidades de **visualización** del proceso que ofrece Kanban, con el tablero Kanban.

En **Scrumban** realmente se tiende a adoptar un modelo de proceso con flujo continuo, como propone Kanban, de manera que la pila del sprint ya no es necesaria y todas las tareas pendientes están agrupadas en la columna inicial del tablero (Vila, 2016).

9. Extensiones de Scrum

Combinación de Scrum y XP

Scrum no prescribe la utilización de prácticas de ingeniería concretas y se reduce a proporcionar un marco de trabajo general para gestionar el proceso.

Es habitual que muchos equipos adopten progresivamente prácticas concretas, como las **12 prácticas propuestas por la Programación Extrema** (Beck, 2000).

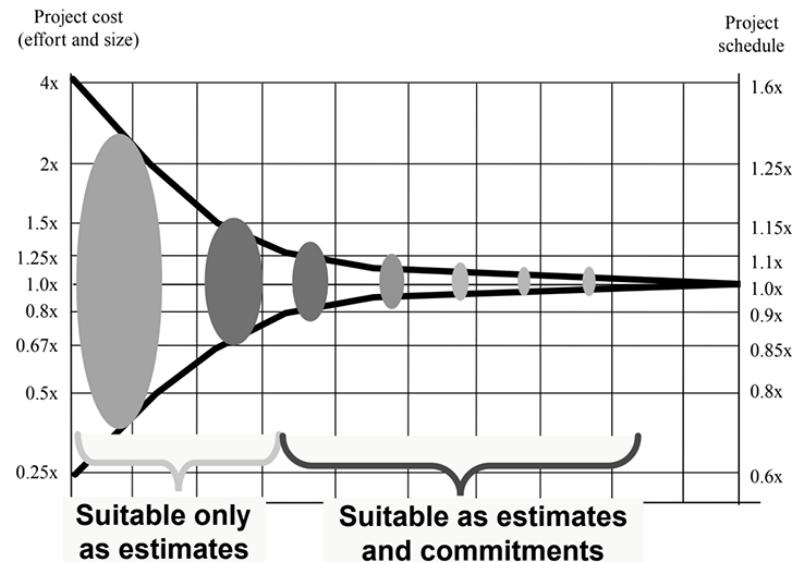
Entre ellas podemos citar:

- › La programación en parejas
- › El empleo de estándares de programación
- › La propiedad del código compartida.

10. Estimaciones ágiles

Uno de los factores críticos en la planificación de cada sprint es la **definición de su objetivo**, dependiente de la carga de trabajo que el equipo puede asumir.

En proyectos de desarrollo de software la **incertidumbre inicial es muy grande**, y se va reduciendo a medida que avanza el proyecto siguiendo una secuencia denominada «**cono de incertidumbre de las estimaciones**».



10. Estimaciones ágiles

Estimaciones de los PBI y las tareas

Es evidente que realizar estimaciones precisas en horas de trabajo es complejo, más aún en proyectos complejos con gran incertidumbre. Para ello, es habitual seguir la siguiente técnica:

Estimar la duración de las historias de usuario en puntos de historia (Story Points).

- Para ello, el equipo de desarrollo dialoga con el PO para comprender a fondo cada historia y, entre todos, asignan a cada una de ellas una puntuación tomando valores del conjunto {0, 1, 3, 5, 8, 13, 21, 40, 100} (Alaimo, 2013).
- Esto da una idea al menos de la complejidad relativa de las historias del PB.

10. Estimaciones ágiles

Estimaciones de los PBI y las tareas

Estimar las tareas en horas de trabajo.

- Durante la reunión de planificación del sprint, se realiza una descomposición de las historias del Sprint Backlog en tareas concretas de implementación.
- En este momento es más fácil realizar estimaciones más precisas para cada una de las tareas o actividades individuales y, así, establecer la carga de trabajo concreta para cada sprint individual.

10. Estimaciones ágiles

Estimaciones de los PBI y las tareas

La información obtenida complementa las tarjetas de historia de usuario.



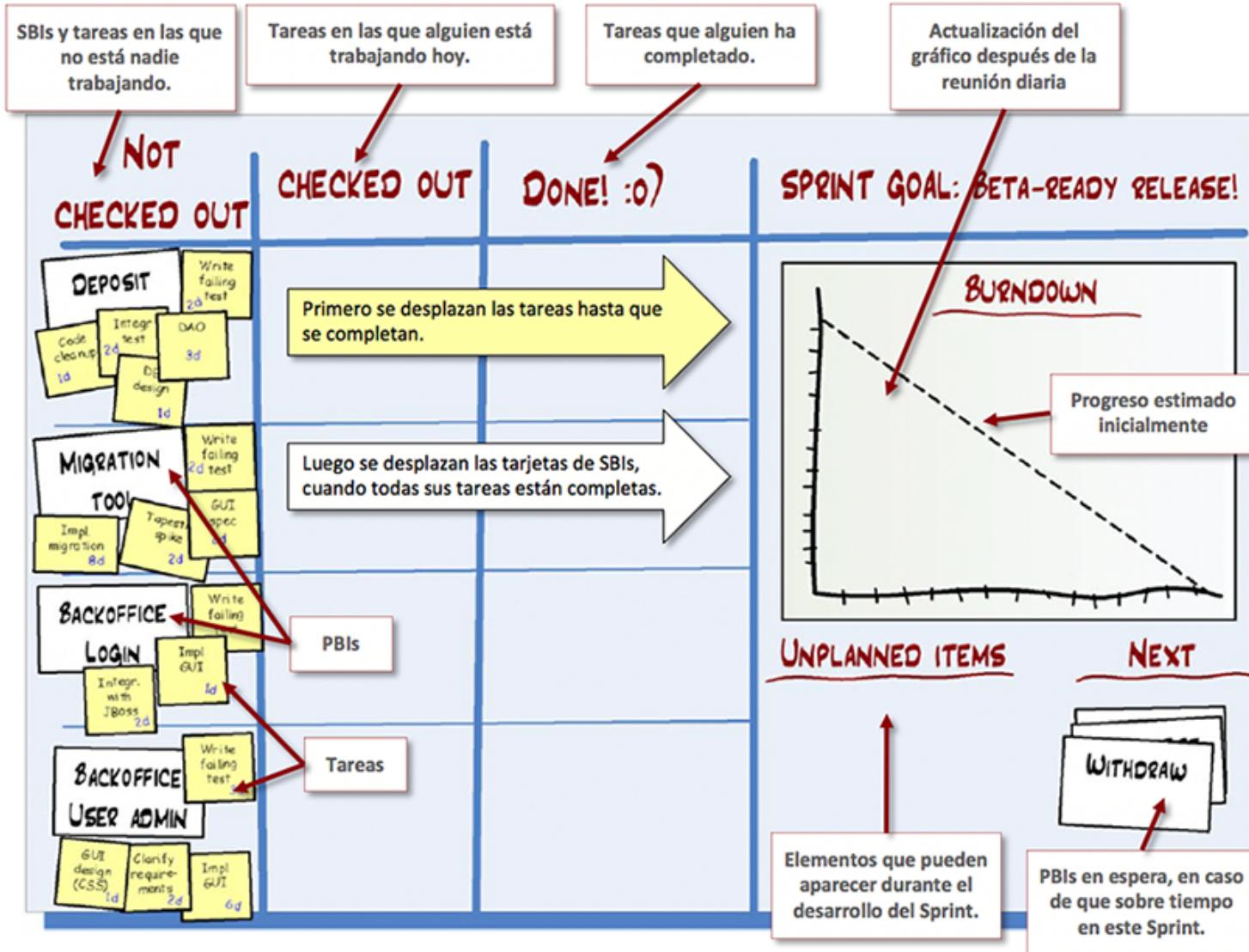
10. Estimaciones ágiles

Seguimiento del sprint

La suma de horas de trabajo asignadas a todas las tareas que forman parte de un sprint permite determinar **la carga total de trabajo**, teniendo en cuenta las capacidades del equipo de desarrollo (número de integrantes, horas de dedicación, etc.).

El seguimiento del progreso de desarrollo se realiza habitualmente empleando un artefacto adicional: el **Burndown Chart** (Kniberg, 2015).

Este gráfico muestra en abscisas (eje horizontal) los días que componen el sprint, y en ordenadas (eje vertical) las horas de trabajo correspondientes a las tareas que restan por implementar.



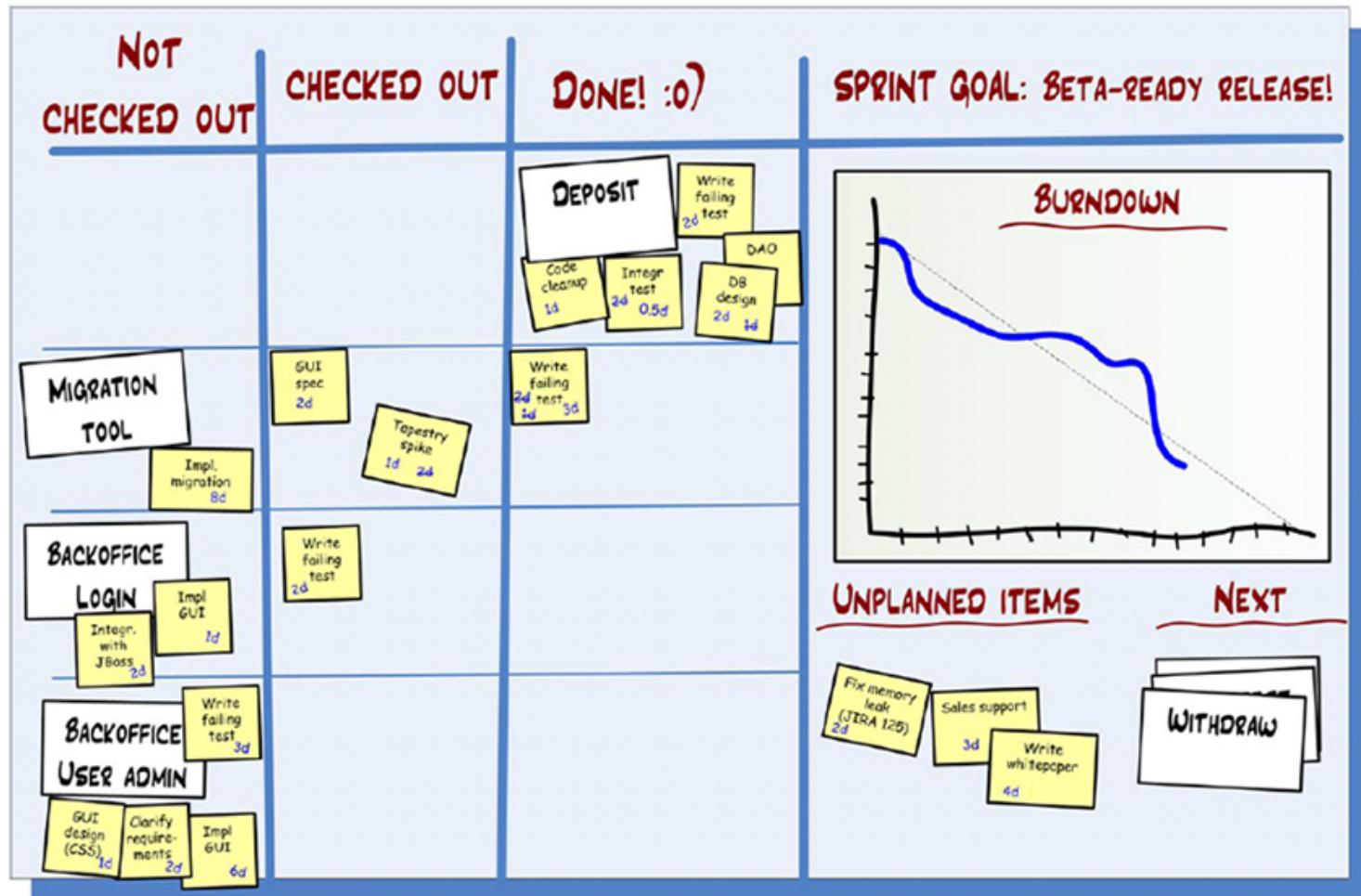
10. Estimaciones ágiles

Seguimiento del sprint

Al final de cada **Scrum diario**, el equipo modifica el panel con las tareas que finalizó el día anterior, y es el momento de actualizar el gráfico de progreso con la suma de horas correspondientes a todas estas tareas.

Una posible situación en **mitad de un sprint** puede presentar que:

- Se ha completado una historia junto con todas sus tareas.
- Otras tres historias están en progreso, y en el momento actual solo hay en desarrollo 3 tareas.
- Además, han aparecido 3 tareas imprevistas, y el gráfico de avance sigue aproximadamente la evolución planificada.



11. Escalado de Scrum

Scrum es una metodología pensada para equipos de desarrollo pequeños, donde la transparencia de la información y la comunicación habitual y fluida son posibles.

Existen proyectos de grandes dimensiones donde es necesaria la participación de equipos más voluminosos.

En estos casos, el enfoque general es la distribución del personal en **varios equipos de Scrum**, definiendo cauces de comunicación adicionales que favorecen la aplicación del método general.

La distribución de los equipos

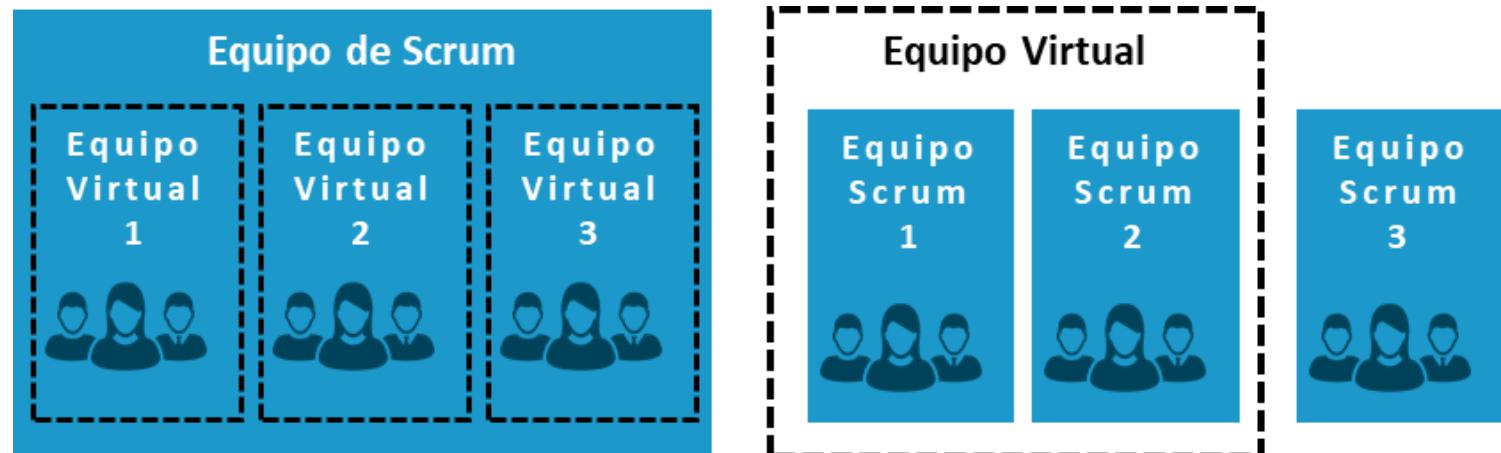
Trabajar con equipos de gran tamaño introduce dificultades, como la excesiva duración de las reuniones diarias, o la falta de visión general de los miembros del equipo por falta de comunicación.

11. Escalado de Scrum

La distribución de los equipos

Cuando trabajamos con **equipos experimentados** y es factible dividir el desarrollo en varias líneas diferenciadas que no comparten código, **es posible la división del equipo**. En otro caso suele ser preferible mantener un único equipo.

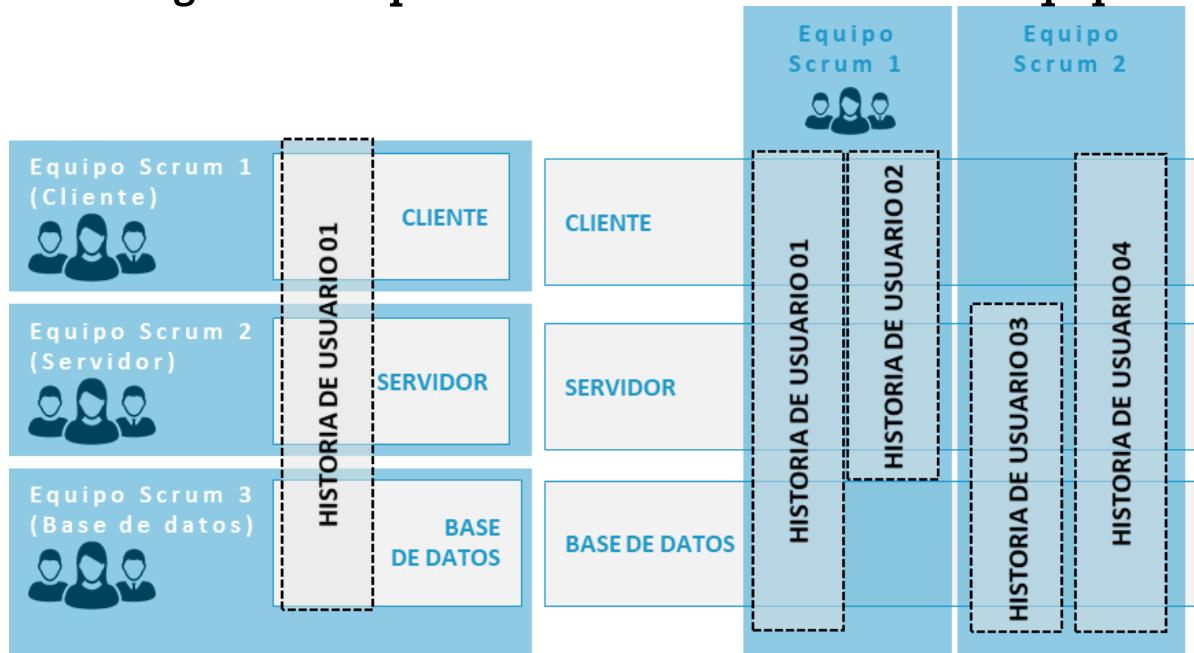
Dada una conformación de equipos, el Scrum Master debe observar cómo se establecen los **cauces de comunicación** entre sus miembros.



11. Escalado de Scrum

La distribución de los equipos

Otro factor es el **grado de especialización** de cada uno de los equipos.



11. Escalado de Scrum

La sincronización de los sprints

Cuando tengamos varios equipos trabajando en paralelo en el mismo proyecto, es necesario determinar una manera de sincronizar sus flujos de trabajo.

La recomendación es que trabajemos con sprints sincronizados, con todos los equipos comenzando y finalizando sus iteraciones al mismo tiempo.

Las ventajas de este enfoque son las siguientes:

- **Disponemos de un punto natural en el que realizar redistribuciones de los equipos**, sin que ello entorpezca el normal funcionamiento de uno de ellos.
- **Se fomenta la colaboración entre equipos**, que pueden comunicarse entre sí a la hora de planificar los objetivos de cada sprint.
- **Se reduce el número de reuniones y la carga administrativa**. Las reuniones de revisión se pueden tener de manera conjunta, realizando una única demostración.

11. Escalado de Scrum

Scrum de Scrums

Cuando trabajamos en **proyectos grandes con varios equipos Scrum trabajando simultáneamente**, es necesario establecer cauces de comunicación y sincronización entre los diferentes equipos.

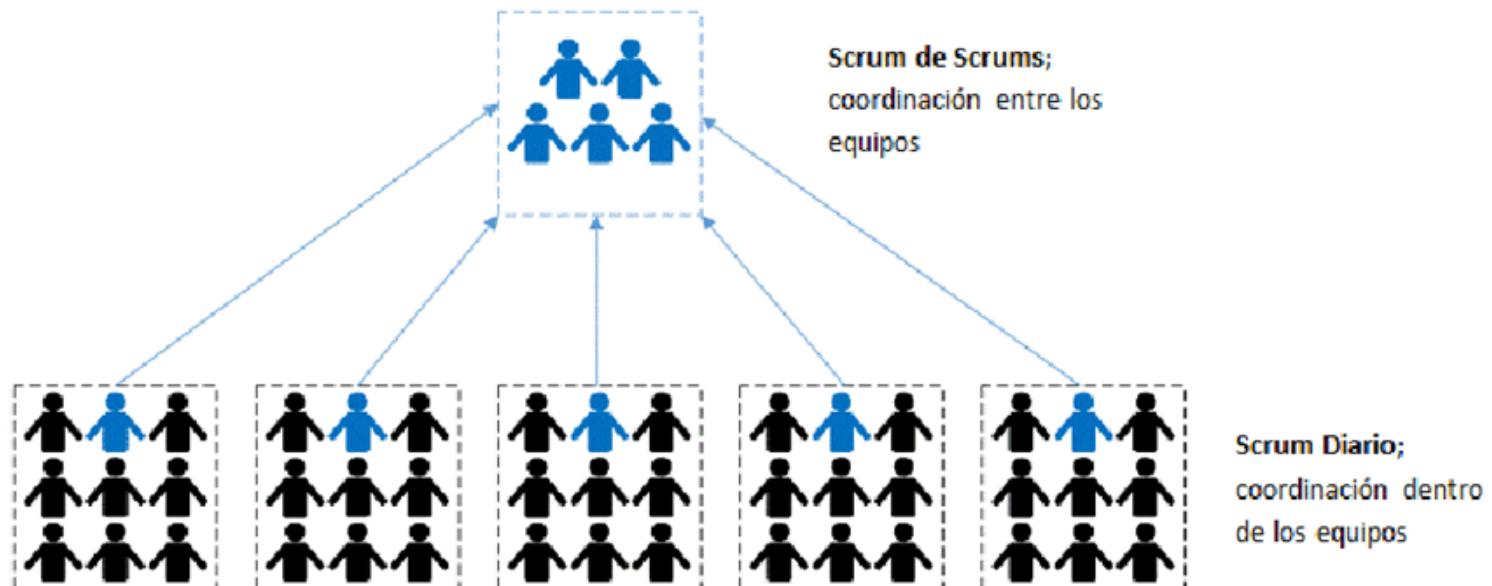
El Scrum de Scrums es una reunión periódica, frecuentemente diaria y con posterioridad al Scrum diario de cada equipo, en la que uno o dos representantes de cada uno de ellos acude para compartir información.

Es preferible que se trate de representantes con perfil técnico, con capacidad para decidir sobre el proceso de desarrollo y asumir compromisos, y con una visión general de la iteración actual.

En algunos casos el enviado es precisamente el Scrum Master de cada equipo, aunque no sea lo idóneo.

11. Escalado de Scrum

Scrum de Scrums



11. Escalado de Scrum

Scrum de Scrums

En estas reuniones, los representantes de los equipos deben responder a las siguientes preguntas (tres de ellas similares a las planteadas en el Daily Scrum):

- ¿Qué tareas ha realizado tu equipo desde la última reunión?
- ¿Qué tareas va a realizar tu equipo hasta la siguiente reunión para lograr el objetivo del sprint?
- ¿Existe algún impedimento que afecte de manera negativa a tu equipo?
- ¿Existe algún factor con origen en tu equipo que puede afectar a los objetivos del resto de equipos?

11. Escalado de Scrum

La gestión del Product Backlog

Cuando tenemos varios equipos de Scrum, cada uno de ellos con su Scrum Master, existen tres estrategias a la hora de gestionar los requisitos del sistema en su totalidad:

Un PO y un PB.

- Este es el enfoque preferido.
- Toda la información sobre requisitos del producto está centralizada, y el Product Owner tiene una visión global del proceso de desarrollo.
- Las reuniones de planificación de cada sprint se suelen realizar en el mismo lugar y momento, pero en espacios diferenciados para los diferentes equipos.
- Es un trabajo intenso, pero al mismo tiempo se fomenta la comunicación entre todos los participantes.

11. Escalado de Scrum

La gestión del Product Backlog

Un PO y varios PB.

- El mayor problema de este enfoque consiste en la asignación de historias a los diferentes equipos.
- Puesto que el PB es gestionado por el PO, se espera que sea este rol quien haga la asignación, pero esta es una tarea que requiere en general de un perfil técnico.

Varios PO, cada uno con su PB.

- Este enfoque es el más problemático, pues se diluye la visión general del producto entre varios PO que deben coordinarse entre sí.
- Además, en el caso de que los diferentes PB tengan bases de código comunes, se puede dificultar el proceso de desarrollo.

11. Escalado de Scrum

La gestión del Product Backlog

Varios PO, cada uno con su PB.

- En un caso ideal en el que los diferentes equipos estuvieran asignados a componentes con distintas bases de código, estaríamos próximos a un escenario en el que el producto se descompone en varios subproyectos relativamente independientes, y solo en este caso este enfoque sería adecuado.

11. Referencias

- Agile Alliance. (2015, diciembre 17). What does INVEST Stand For? [Blog corporativo]. Recuperado de <https://www.agilealliance.org/glossary/invest/>
- Alaimo, D. M. (2013). Proyectos ágiles con Scrum. Buenos Aires: Ediciones Kleer.
- Albaladejo, X. (2009). Introducción a la estimación y planificación ágil [Web]. Recuperado de <https://proyectosagiles.org/2009/06/08/introduccion-estimacion-planificacion-agil/>
- Beck, K. (2000). Extreme Programming Explained. Embrace Change. Boston: Addison-Wesley.
- CollabNet. (2012). ScrumMaster Checklist [Web]. Recuperado de <https://www.collab.net/resources/scrummaster-checklist>
- Ferrer, M. (2016). Escalando equipos Scrum. Scrum de Scrums de Scrums [Web]. Recuperado de <http://managementplaza.es/blog/escalando-equipos-scrum/>

11. Referencias

- ▶ James, M. (2012). The ScrumMaster Checklist [Web]. Recuperado de <http://scrummasterchecklist.org/>
- ▶ Kniberg, H. (2010). Scrum Checklist [Web]. Recuperado de <https://www.crisp.se/gratis-material-och-guider/scrum-checklist>
- ▶ Kniberg, H. (2015). Scrum and XP from the trenches: how we do Scrum (2^a ed.). C4Media.
- ▶ McConnell, S. (2008). 10 Deadly Sins of Software Estimation [presentación]. Recuperado de <http://www.austin-cs.org/archive/10-Deadly-Sins-of-Software-Estimation>
- ▶ McLaughlin, M. (2018). What is Agile Methodology? [Web]. Recuperado de <https://www.versionone.com/agile-101/agile-methodologies/>
- ▶ Schwaber, K. y Sutherland, J. (2017). La Guía Definitiva de Scrum: Las Reglas del Juego [Web]. Recuperado de <https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-Spanish-European.pdf#zoom=100>

11. Referencias

- Sutherland, J. y Schwaber, K. (2012). The Scrum Papers: Nuts, Bolts, and Origins of an Agile Process (Version 1.1). Massachusetts: Scrum Inc.
- Vila, J. L. (2016). Scrumban [Web]. Recuperado de <http://managementplaza.es/blog/scrumban/>
- Wake, B. (2003). INVEST in Good Stories, and SMART Tasks [Blog personal]. Recuperado de <https://xp123.com/articles/invest-in-good-stories-and-smart-tasks/>