

---

# METODOLOGÍAS ÁGILES: *SCRUM*

---



# ÍNDICE

---

<b>INTRODUCCIÓN</b>	3
<b>OBJETIVOS</b>	4
<b>HISTORIA Y CONTEXTO</b>	5
Complejidad e incertidumbre	5
¿Por qué las metodologías ágiles están en auge en la actualidad?	6
<b>EL <i>FRAMEWORK</i> SCRUM</b>	7
El ciclo de Scrum	9
El equipo Scrum	11
Eventos de Scrum	16
Artefactos de Scrum	23
<b>RESUMEN</b>	28
<b>CONCEPTOS CLAVE</b>	29
<b>BIBLIOGRAFÍA</b>	30

# INTRODUCCIÓN

En esta unidad haremos una breve explicación del **contexto** en que se creó Scrum y de qué objetivo se buscaba cuando fue creado. Entender qué necesidades vino a cubrir en sus inicios te ayudará a entender por qué hoy en día este marco de trabajo (*framework*) está cada vez más extendido, ya que se adapta a las necesidades cambiantes del mundo en el que vivimos.

A continuación, veremos en detalle **las características de Scrum, sus valores, artefactos y eventos**. Además de conocer los elementos de Scrum, te explicaremos el objetivo que hay detrás de cada uno de ellos, para que puedas reflexionar acerca de su posible papel en tu contexto profesional.

También veremos cada uno de los **roles implicados en Scrum y sus responsabilidades y funciones**. Analizaremos los eventos o reuniones de Scrum que están pensados para maximizar el rendimiento del equipo, detallando qué actividades se desarrollan y qué objetivos y resultados se obtienen en ellos.

**En resumen, tendrás una idea clara de qué es Scrum, para qué sirve y cómo trabajar en un producto o proyecto con este *framework*.**

# OBJETIVOS

En esta unidad podrás:

1. **Analizar las necesidades y contextos** sobre los que se creó Scrum para entender en qué situaciones aplica mejor.
2. **Conocer las características de Scrum** y entender para qué sirve cada elemento del mismo.
3. **Conocer los retos** que pueden surgir a la hora de comenzar a trabajar con Scrum.

# HISTORIA Y CONTEXTO

Scrum fue presentado oficialmente en 1995 por Jeff Sutherland y Ken Schwaber, dos profesionales del mundo del software que, motivados por la necesidad de mejorar los procesos implicados en sus proyectos, decidieron crear un **marco de trabajo que se adaptara a proyectos con un alto grado de complejidad e impredecibilidad.**



El término *Scrum* tiene su origen en un estudio presentado en 1986 sobre cómo trabajaban los equipos de desarrollo de producto de las empresas que estaban teniendo éxito en la creación de nuevos productos en Estados Unidos y Japón (Fuji-Xerox, Canon, Honda, NEC, Epson, Brother, 3M, Xerox y Hewlett-Packard).

Este estudio compara las formas de trabajar de estos equipos multidisciplinares y altamente productivos con la unidad de los equipos de rugby que avanzan juntos detrás del balón y las características de la formación del *scrum* (*melé en español*).



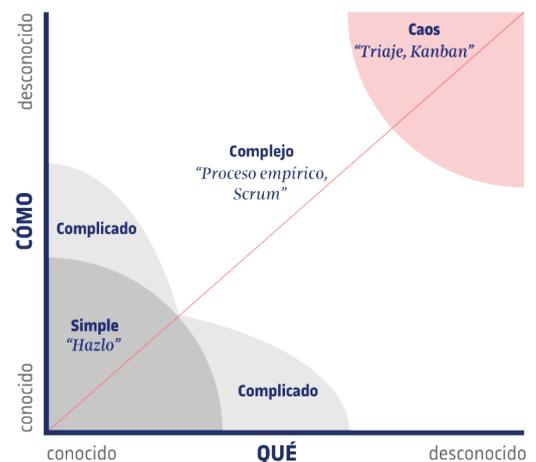
## Complejidad e incertidumbre

Scrum, como otros métodos ágiles, es un proceso empírico que busca adaptarse bien a entornos complejos.

La matriz de Stacey es una forma de representar estos entornos o proyectos complejos.

En un eje se sitúa la certeza que tenemos sobre el **Qué** (requerimientos, «¿Qué es necesario?») y en el otro, la certeza del **Cómo** (procedimiento «¿Cómo llegaremos al resultado deseado?»).

**Matriz de Stacey**



Eso nos permite clasificar los entornos o proyectos en:

- **Simples:** en los que sabemos exactamente qué debemos construir y cómo lo haremos.
- **Complicados:** en los que se añade incertidumbre a alguna de estas dimensiones.

A medida que haya más incertidumbre, entraremos en entornos o proyectos complejos, y si además tenemos en cuenta el tercer eje de las personas, mientras más individuos participen en un proyecto, mayor complejidad existirá.

## ¿Por qué las metodologías ágiles están en auge en la actualidad?

En entornos complejos es muy difícil predecir qué pasará a lo largo del proyecto: desconocemos mucho más de lo que conocemos.

En estas situaciones podemos predecir un rango de posibilidades, pero hay tantas variables en juego que nos resulta imposible tenerlas todas en cuenta. Por este motivo, hacer una gestión predictiva en un entorno complejo es prácticamente imposible.

**En los contextos complejos se necesitan respuestas emergentes que surjan como consecuencia de aprender de las acciones anteriores.** Además, es muy importante que las personas responsables de esas respuestas emergentes tengan capacidad para tomar decisiones.

Actualmente vivimos en un mundo donde los cambios se producen a gran velocidad, los proyectos que integran múltiples disciplinas son cada vez más comunes y el nivel de incertidumbre es muy elevado.

Gestionar proyectos con métodos empíricos pero que a la vez se adapten a esta complejidad es cada vez más necesario y por eso las metodologías ágiles están cada vez más presentes en diferentes entornos.

# EL *FRAMEWORK* SCRUM

*Scrum es un framework diseñado para abordar **problemas y proyectos complejos** de manera que nos permita una **alta productividad y creatividad** a medida que vamos **entregando producto de forma continua** con el **mayor valor posible**.*

Para conseguir esto, Scrum se basa en la teoría de control de procesos empíricos, donde se entiende que el conocimiento proviene de la experiencia y la toma de decisiones está basada en lo que se conoce.

Scrum tiene un enfoque iterativo e incremental para optimizar la previsibilidad y controlar el riesgo de manera implícita.

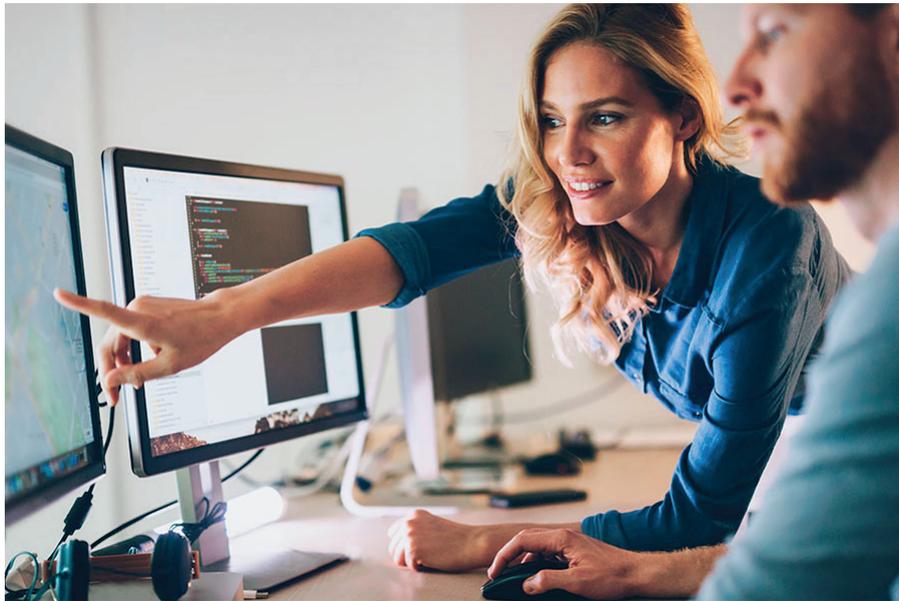
Los tres pilares del control de proceso empírico y que, en consecuencia, también son los **pilares de Scrum** son:

- **Transparencia**  
La información y aspectos relevantes del proceso deben estar a disposición y ser entendidos por todos los implicados.
- **Inspección**  
Debemos inspeccionar continuamente los artefactos de Scrum y cómo vamos progresando hacia el objetivo en cada iteración (o sprint) para evitar desviaciones o variaciones no deseadas. Estas inspecciones no deben interferir en el trabajo del equipo, sino estar integradas en el proceso de trabajo.
- **Adaptación**  
Si se identifican desviaciones fuera de lo esperado en aspectos del propio proceso o en el resultado del producto, debemos hacer los ajustes necesarios para minimizar el impacto.

Scrum es un método de trabajo ligero y fácil de entender, ya que solo define:

- **3 roles**
- **5 eventos**
- **3 + 1 artefactos**

Pero eso no significa que sea un método de trabajo fácil de dominar...



Los principales motivos que lo hacen **difícil de dominar** son los siguientes:

- Incorpora **un ritmo de trabajo y de entrega continuo** que saca a la luz las ineficiencias del equipo y de la organización.
- Supone aprender a **desarrollar roles nuevos** que generalmente son muy diferentes a los que se habían desempeñado hasta el momento.
- Obliga a **gestionar prioridades cambiantes** y asegurar una priorización que maximice el valor entregado. Este aspecto clave para Scrum es una actividad difícil.
- Requiere experiencia para **ganar fluidez** en los eventos de Scrum para que sean eficientes.

**Scrum** se considera un marco de trabajo (*framework*) porque **no es prescriptivo**, sino que solo describe una serie de roles, eventos y artefactos, y **deja libertad** en cómo organizar y llevar a cabo el trabajo del equipo.

A lo largo de los años han surgido muchas prácticas y herramientas que se usan en un contexto de Scrum pero que no son parte formal del *framework*.

## El ciclo de Scrum

El equipo Scrum está compuesto por tres roles: el **Product Owner (PO)**, el equipo de desarrollo y el **Scrum Master (SM)**.

El **Product Owner** es el dueño del **qué**. Es quien tiene la visión del producto y es el responsable de definir las prioridades para asegurar que en cada iteración se construya el producto que más valor aporte.

El **equipo de desarrollo** es el dueño del **cómo**. Es un equipo multidisciplinar de expertos que son los responsables de construir el producto con calidad y asegurando que cumpla las necesidades de sus usuarios.

El **Scrum Master** es quien vela por el proceso de Scrum, promoviendo su correcto funcionamiento y asegurando que todas las partes entiendan sus valores, principios y prácticas.

En Scrum, la definición del producto está reflejada en el **Product Backlog**, que contiene las necesidades y requerimientos del mismo.

El **Product Backlog** consiste en una lista de ítems ordenada de mayor a menor prioridad. Los ítems más prioritarios están más detallados, para poder comenzar a trabajar en ellos. A medida que avancemos en el proyecto, aparecerán nuevos requerimientos y se irán detallando los ítems en los que el equipo tenga que trabajar a continuación.

El proceso iterativo de Scrum está definido por *sprints*, que son iteraciones de una unidad de tiempo definida y constante (entre 1 y 4 semanas) que tienen como resultado un incremento del producto.

Un sprint comienza con una **planificación del sprint (Sprint Planning)** en la que el equipo, de manera colaborativa, define el trabajo que se va a realizar durante el sprint.

El objetivo de la **Sprint Planning** es definir qué incremento de producto se entregará al finalizar el *sprint* y cómo el equipo afrontará el trabajo necesario para conseguir dicho incremento.

Como resultado de la *Sprint Planning* queda definido el **Sprint Backlog**, que es el conjunto de ítems del *Product Backlog* que han sido seleccionados para el *sprint*, junto al plan de cómo el equipo construirá el incremento para conseguir el objetivo del *sprint*.

Una vez el objetivo del *sprint* está definido y la planificación está hecha, el equipo comienza a desarrollar el incremento del *sprint* de forma autoorganizada.

Para coordinarse y asegurarse de que no hay desviaciones y para adaptarse en caso de que las haya, cada día el equipo hace una reunión de sincronización de 15 minutos.

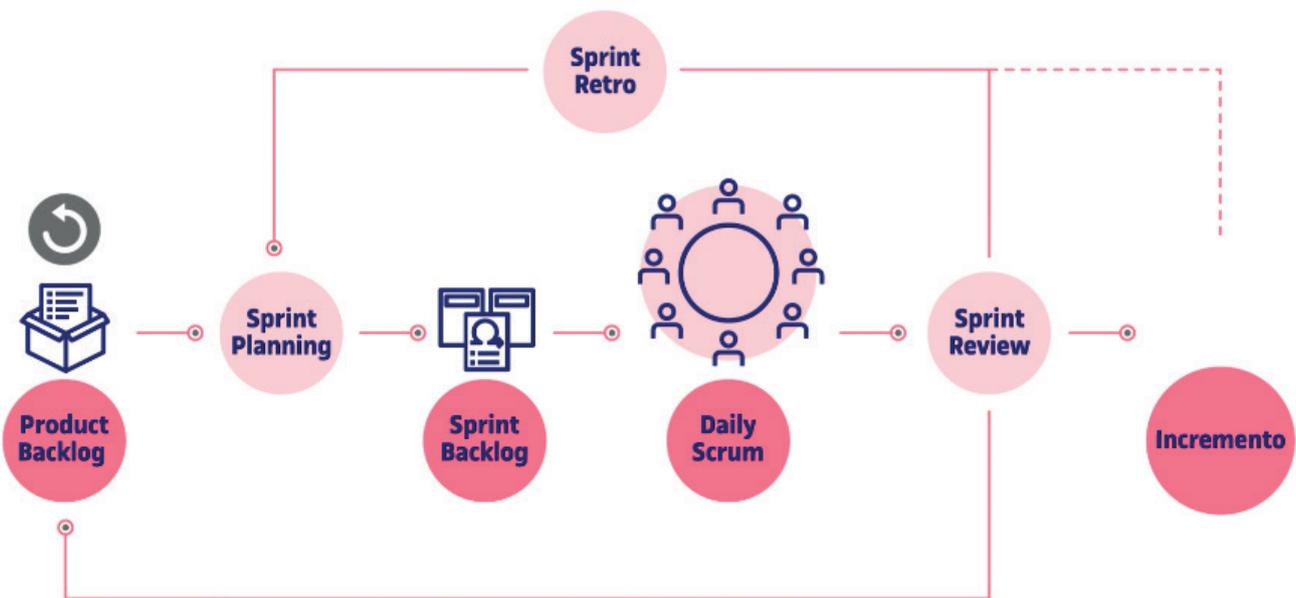
Esta reunión tiene el nombre de **Daily Scrum**.

El objetivo de la **Daily Scrum** es revisar el avance del sprint y tomar acciones necesarias ante desviaciones del plan, bloqueos, etc.

Al finalizar el *sprint* se hace su revisión. Este evento se denomina **Sprint Review** y tiene por objetivo revisar el resultado del *sprint*, el incremento del producto, y adaptar el *Product Backlog* si es necesario, en función del *feedback* recibido por los *stakeholders* que participen en la reunión.

Por último, antes de comenzar el siguiente *sprint*, se hace la **Sprint Retro**. Es un momento en que el equipo se inspecciona a sí mismo y crea un plan de mejora para el siguiente *sprint* y así mejorar de forma continua su método de trabajo.

Para resumir, el siguiente esquema es un buen resumen del ciclo de Scrum:

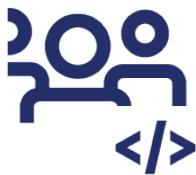


## El equipo Scrum

El equipo Scrum está formado por:



El *Product Owner* (PO)



El equipo de desarrollo



El *Scrum Master* (SM)

Los equipos scrum son **autoorganizados** y **multifuncionales**:

- **Equipos autoorganizados**  
Eligen la mejor manera de realizar su trabajo, en lugar de ser dirigidos por personas que no pertenezcan al equipo.
- **Equipos multifuncionales**  
Tienen todas las competencias necesarias para realizar el trabajo sin depender de otros que no forman parte del equipo.



Los equipos Scrum entregan productos de manera **iterativa e incremental**, maximizando las oportunidades de *feedback*.

Las entregas incrementales del producto acabado aseguran que una versión potencialmente útil del producto esté siempre disponible.

## Product Owner (PO)



El *Product Owner* (PO) es el responsable de maximizar el valor del producto que se obtiene como resultado del trabajo del equipo de desarrollo.

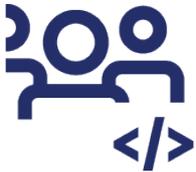
- **Tiene clara la visión del producto.** No tiene conocimiento de todas las características que tendrá el producto, pero sabe por qué se está construyendo, qué necesidad cubre.
- **Toma decisiones del producto.** Especialmente sabe y puede decir «no». Al construir un producto de forma iterativa y recibir *feedback* constante, hay aún más posibilidades de recibir nuevas ideas de las partes interesadas, pero para no colapsar el producto y el equipo es importante saber decidir qué ideas se implementarán y cuáles no.
- **Fomenta la comunicación entre el equipo y los *stakeholders*** para trabajar de forma colaborativa la definición del producto.
- **Gestiona las expectativas de los *stakeholders*** y las traslada al *backlog* si lo considera necesario.

El PO es el responsable de gestionar el *Product Backlog* y para ello se asegura de:

- **Detallar** de forma clara los ítems del *Product Backlog* a medida que estos son más prioritarios.
- **Priorizar** el *backlog* para asegurar que se cumplen los objetivos y se maximiza el ROI (retorno de la inversión).
- **Visibilizar:** asegurar que el *Product Backlog* es visible, transparente y claro para todos los interesados, y que muestra qué es lo próximo en lo que trabajará el equipo de desarrollo.
- **Acompañar:** asegurar que el equipo de desarrollo entiende los ítems del *backlog* al nivel necesario para poder trabajar en ellos.

Si bien el *Product Owner* es el responsable de la gestión del *backlog*, puede delegar parte del trabajo descrito en el equipo de desarrollo.

## Equipo de desarrollo



El Equipo de desarrollo está conformado por las personas que realizan el trabajo necesario para poder entregar un incremento de producto terminado (*done*) al finalizar el *sprint*.

Solo los miembros del equipo **crean un incremento**. Por eso es clave que el equipo sea **multidisciplinar**. Esto quiere decir que el equipo debe tener todos los conocimientos y habilidades necesarias para poder trabajar en una parte del producto de inicio a fin.



**El equipo de desarrollo Scrum es un equipo autoorganizado y multidisciplinar.**

Es **autoorganizado** porque son los responsables de decidir cómo construyen el incremento de producto del *sprint*. Y es **multidisciplinar** porque tiene todas las habilidades necesarias para entregar incrementos de su producto o proyecto.

El equipo de desarrollo interactúa con los *stakeholders* para compartir conocimiento de negocio, alinear prioridades y medir los resultados del producto entregado.

El tamaño del equipo debe ser lo **suficientemente pequeño para mantenerse ágil** y lo **suficientemente grande como para poder crear incrementos** del producto significativos en un *sprint*. Se recomienda que los equipos tengan entre tres y nueve personas. Con más de nueve se necesitará mucha coordinación y aumentará la complejidad.

En los equipos de desarrollo Scrum **no hay roles específicos**, independientemente del trabajo que haga cada uno. Tampoco hay subequipos, más allá de los dominios necesarios. Ciertas personas del equipo pueden tener habilidades o conocimientos específicos o un área de especialización, pero igualmente la responsabilidad recae en el equipo como un todo.

Es muy importante que un equipo de desarrollo Scrum trabaje realmente como un equipo cohesionado y con un objetivo común.

**En un equipo Scrum, este y el objetivo del *sprint* siempre están por encima de cada una de las personas.**

### **Scrum Master (SM)**



**El *Scrum Master* (SM) es responsable de promover y ayudar a entender el correcto funcionamiento de Scrum. Ayuda a que todos entiendan la teoría, prácticas, reglas y valores de Scrum.**

**El SM es un líder al servicio del equipo Scrum.**

**El principal objetivo del *Scrum Master* es conseguir un equipo de alto rendimiento en el desarrollo de producto iterativo e incremental.**

El Scrum Master se encarga de conseguir que el equipo funcione con los principios y valores de Agile, tanto dentro del equipo Scrum como en su relación con las personas o equipos con que interactúa. Tiene por objetivo asegurar que esas interacciones maximizan el valor entregado por el equipo.



Si hay un equipo Scrum que desarrolla una aplicación y tiene una dependencia puntual con un sistema cuyo equipo aún no trabaja con metodologías ágiles y tiene un ciclo de desarrollo de meses y unos requerimientos de análisis previos muy extensos, el SM deberá mediar entre ambos equipos y también con la organización para que se entiendan las necesidades del equipo Scrum, así como que, para poder funcionar bien, otros equipos deberán adaptarse a ciclos de desarrollo iterativos.

Para conseguir esto, el SM trabaja con una **visión sistémica del equipo** asegurando que el equipo de desarrollo rinda al máximo posible, que se haga una gestión ágil del producto y que la organización se adapte a los métodos de trabajo y necesidades de Scrum.

- **Funciones**

Las principales funciones de un SM son guiar al equipo y a la organización para que trabajen bajo los valores y principios de Agile, remover impedimentos, facilitar los eventos de Scrum y trabajar en la mejora continua del equipo.

- **Habilidades**

Es importante que un SM tenga buenas habilidades de liderazgo, comunicación, facilitación, *coaching* y también orientación a resultados. Estas habilidades ayudarán a que el trabajo que haga con el equipo sea permanente, ya que se hará propio al equipo.

- **Misión**

Su misión es conseguir que el equipo sea autónomo tanto en la creación de producto como también en la mentalidad de trabajo y la búsqueda de la excelencia. El SM debe buscar ser lo más prescindible posible y conseguir que, incluso en su ausencia, el equipo mantenga una mentalidad ágil de trabajo.

El *Scrum Master* sirve al *Product Owner*, al equipo de desarrollo y a la organización.

### El SM sirve al PO:

- Garantizando que el objetivo, alcance y dominio del producto son entendidos por el equipo Scrum.
- Encontrando técnicas para gestionar el *backlog* de manera efectiva.
- Asumiendo la necesidad de tener ítems del *Product Backlog* claros y concisos.
- Entendiendo la planificación de producto en un entorno empírico.
- Asegurando que el PO sabe cómo organizar el *Product Backlog* para maximizar el valor.

### El SM sirve al equipo de desarrollo:

- Ayudando al equipo a conseguir una autoorganización y multidisciplinariedad efectiva mediante enseñanza y *coaching*.
- Impulsando al equipo para crear productos de alto valor, poniendo el foco en la calidad técnica de su trabajo.
- Eliminando impedimentos que dificulten el progreso.

- Guiando en entornos organizacionales en los que Scrum aún no haya sido adoptado y entendido por completo.

### El SM sirve a la organización:

- Liderando y guiando a la organización en la adopción de Scrum.
- Generando cambios que incrementen la productividad del equipo Scrum.
- Trabajando con otros SM para aumentar la efectividad de la aplicación de Scrum en la organización.
- Ayudando a las personas y a la organización a entender y apoyar el desarrollo empírico de producto.

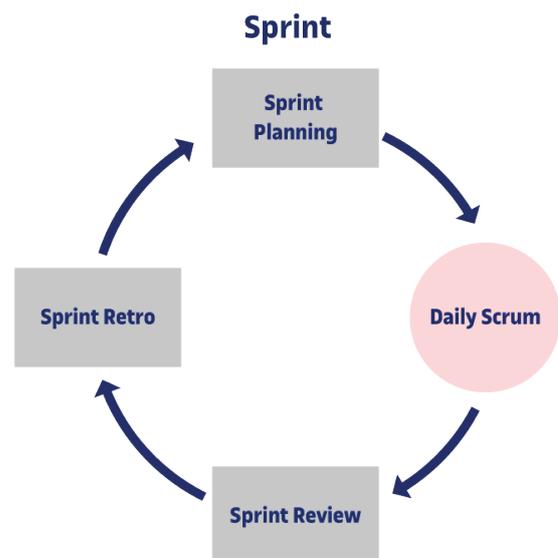
## Eventos de Scrum

Los eventos de Scrum se usan para conseguir regularidad y ritmo, a la vez que reducen la necesidad de otro tipo de reuniones. El *sprint* contiene el resto de eventos y cada uno de estos existe con el propósito de inspeccionar y adaptar alguna parte del proceso o producto.

**Los eventos están diseñados para conseguir la máxima transparencia e inspección.**

A continuación revisaremos los siguientes **eventos de Scrum**:

- El *sprint*
- *Sprint Planning* (planificación del *sprint*)
- *Sprint Goal* (objetivo del *sprint*)
- *Daily Scrum*
- *Sprint Review*
- *Sprint Retro* (retrospectiva)



## El *sprint*

*El núcleo de Scrum es el **sprint**, una duración de tiempo (iteración) de un mes o menos durante el cual se crea un **incremento del producto terminado** y que potencialmente pueda ser lanzado a mercado o entregado al cliente.*

Los *sprints* tienen una duración constante, no deben alargarse ni acortarse, ni hacer *sprints* de diferente duración. Esto impediría que el equipo consiguiera un ritmo de trabajo constante y dificultaría la predictibilidad y capacidad de planificación del equipo.

En Scrum se usa el *sprint* para fijar el tiempo durante el que trabajará el equipo. De esta forma, se consigue que **el equipo tenga claro el objetivo del *sprint* y consiga centrarse en conseguirlo**. Para esto hay que conseguir que no haya cambios durante ese período de tiempo que hagan peligrar el objetivo del *sprint* o su calidad.

Un nuevo *sprint* empezará inmediatamente después de que termine el anterior.

Cada *sprint* puede ser visto como un miniproyecto con un objetivo claro sobre qué construir, un plan flexible que nos guiará en la construcción (*Sprint Plan*), el trabajo propiamente dicho y el incremento de producto resultante.

### ***Sprint Planning* (planificación del *sprint*)**

*El **Sprint Planning** consiste en una reunión que tiene lugar antes del inicio del *sprint*, en la que el equipo acuerda el objetivo del *sprint* y planifica el trabajo que se va a realizar para conseguirlo.*

El *Sprint Planning* se divide en dos partes:

- En la primera se trata de **qué** se va a hacer en el siguiente *sprint*.
- En la segunda parte, se discute **cómo** organizará su trabajo el equipo para conseguir el objetivo del *sprint*.

## QUÉ

En la primera parte del *Sprint Planning*:

- El PO presenta al equipo el *Product Backlog* actualizado y priorizado y define el objetivo del *sprint*.
- El equipo examina la lista, clarifica las dudas con el PO y selecciona los elementos del *Product Backlog* que prevé completar en el *sprint*, teniendo en mente el objetivo del *sprint* propuesto por el PO.

## CÓMO

En la segunda parte del *Sprint Planning*, una vez el PO y el equipo de desarrollo acuerdan el alcance del *sprint*:

- El equipo de desarrollo define las tareas necesarias para poder completar cada ítem del *Backlog* que haya sido seleccionado para el *sprint*. Esto lo hace el equipo, que es el responsable de organizar su trabajo y quien mejor conoce cómo realizarlo.

Como **resultado** del *Sprint Planning*, además de tener un **objetivo de *sprint* claro**, tendremos el ***Sprint Backlog***, el conjunto de elementos del *backlog* que han sido elegidos para el *sprint* junto con la definición de tareas necesarias para construirlas.

### ***Sprint Goal* (objetivo del *sprint*)**

*El Sprint Goal es el objetivo fijado para el sprint que se conseguirá mediante la implementación de los elementos del Product Backlog.*

El *Sprint Goal*:

- Clarifica al equipo para qué está construyendo el incremento.
- Le da cierta flexibilidad a la hora de trabajar en las distintas funcionalidades o características del producto.
- Ayuda a que el equipo no piense en objetivos o tareas individuales

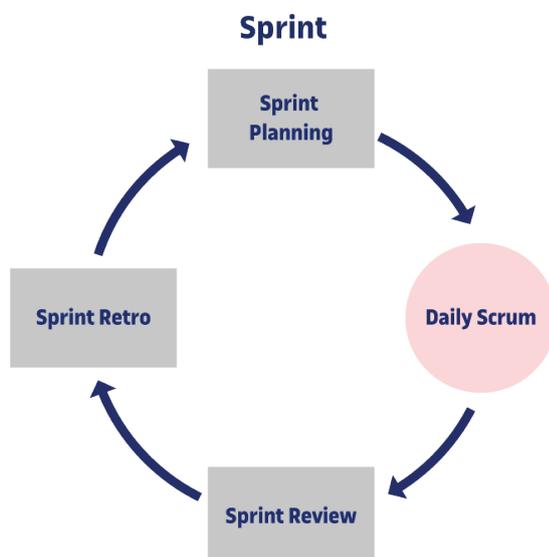
## ¿Cómo orientan los *Sprint Goals* el trabajo de un equipo Scrum?

Si tomamos como ejemplo un proyecto de diseño de un *e-commerce*, un objetivo de *sprint* podría ser «Recomendaciones de productos» para conseguir que la plataforma recomiende productos al usuario.

Para ello se deberán desarrollar varias funcionalidades, que serán el principal foco del equipo, pero no necesariamente serán el único trabajo del *sprint*. Seguramente habrá otras funcionalidades, mejoras o correcciones de errores de funcionalidades ya existentes en las que el equipo trabajará. Pero cada persona del equipo entenderá que si hay algo que puede hacer para ayudar a conseguir el objetivo, esto será prioritario. Incluso si es un tipo de trabajo en el que esa persona no es especialista.

### *Daily Scrum*

*La Daily Scrum es una reunión de sincronización del equipo de desarrollo, con una duración máxima de 15 minutos y que se hace a diario, a la misma hora y en el mismo lugar (generalmente delante del tablero de tareas que representa el Sprint Backlog).*



Esta reunión persigue **optimizar la colaboración y el desempeño del equipo**. Se revisa el trabajo realizado desde la *Daily Scrum* anterior y se planea cómo seguir adelante con el trabajo restante para conseguir el objetivo del *sprint*.

Cada día, el equipo debe entender cómo trabajará de forma autoorganizada para analizar todo el trabajo planeado para el *sprint* y entregar el incremento. Para conseguirlo, se expone lo que cada miembro del equipo hizo desde el día anterior y lo que hará durante el presente día, se identifican puntos de bloqueo y se coordina para trabajar juntos.

**Con esta reunión se consigue mejorar la comunicación, eliminar otras reuniones, identificar impedimentos y cómo superarlos, promover tomas de decisiones rápidas y mejorar el nivel de colaboración y conocimiento del equipo.**



El *Scrum Master* enseña al equipo a mantener la reunión en 15 minutos y a asegurar que sirva como punto de coordinación para inspeccionar cómo avanza el *sprint* y adaptar el plan del día para conseguir el objetivo.

Es importante centrarse en que la reunión sea rápida y fluida. Por esto, si en algún momento surge un tema que tiene que ser tratado con más profundidad entre dos o más miembros del equipo, el *Scrum Master* toma nota, ayuda a que la reunión siga su curso y una vez terminada asegura que todos los temas que han surgido sean comentados por las personas implicadas.

## ***Sprint Review***

*La Sprint Review es una reunión que tiene lugar al finalizar el sprint en la que el Product Owner y el equipo de desarrollo presentan a los stakeholders el resultado del sprint y, en especial, el incremento del producto para revisarlo y adaptar el Product Backlog si es necesario en función del feedback tras la revisión.*

*El incremento del producto es validado previamente por el Product Owner, que ha trabajado con el equipo durante el sprint para asegurarse de que cumple con los criterios definidos y con la definición de done del equipo.*

Durante la *Sprint Review*:

- Asisten el equipo Scrum y los *stakeholders* invitados por el PO.
- El PO explica el resultado del sprint, qué tareas han concluido y cuáles no, y si el objetivo del *sprint* se ha conseguido.
- El equipo de desarrollo comenta cómo ha ido el *sprint* (qué ha funcionado, qué problemas se han encontrado, cómo los han solucionado...) e informa a los *stakeholders* de la situación del equipo.
- Se hace una demostración del producto terminado con el objetivo de recibir *feedback* y responder dudas.

- Se revisa el estado del *Product Backlog* y de las entregas programadas.
- Todos los participantes colaboran para alinearse en cuáles serán los próximos pasos, para que la reunión proporcione al equipo *inputs* valiosos para el siguiente *Sprint Planning*.

### Beneficios de la *Sprint Review*

- **Facilita la gestión de expectativas y se obtiene una mayor calidad de *feedback***  
El cliente puede ver de manera objetiva si se están cumpliendo sus expectativas y puede dar *feedback* al respecto.
- **Mejora la toma de decisiones**  
Al ver un producto tangible, el cliente puede entender mejor sus necesidades y tomar mejores decisiones sobre futuros incrementos.
- **Ayuda a entender las necesidades**  
Al colaborar con el cliente, el equipo puede entender sus necesidades reales, y ver qué puntos de mejora tiene el producto y la comunicación con el cliente.
- **Aumenta el compromiso y satisfacción del equipo**  
El equipo se siente más comprometido y satisfecho cuando puede mostrar los resultados de su trabajo y recibir *feedback* directo del cliente.

### Importancia de las *Sprint Reviews*

Es importante **no mostrar un avance parcial** a los *stakeholders*. Si se muestra algo parcialmente terminado, el cliente no podrá apreciar fácilmente todo el trabajo que falta, y una vez esté realmente terminado, tendrá la sensación de que se ha avanzado poco. Incluso podría suceder que el trabajo restante tenga un riesgo asociado que podría cambiar el *feedback* del cliente sobre esa iteración.

Tomemos como ejemplo un proyecto de arquitectura de diseño de una casa. En la revisión solo le mostramos al cliente el diseño 3D con los avances pactados para el *sprint*, pero faltaría hacer el estudio de viabilidad de la estructura nueva y actualizar el presupuesto.

En el mejor de los casos, podría pasar que en el siguiente *sprint* no mostremos casi ningún avance, ya que el trabajo pendiente nos ocupará gran parte del *sprint*, y que no cumplamos las expectativas del cliente.

Y en el peor de los casos, incluso podría pasar que el análisis de viabilidad nos dé como resultado que el diseño no es viable o que, al actualizar el presupuesto con este nuevo diseño, se introduzca algún elemento que dispare el precio, lo que modificará lo que le habíamos mostrado al cliente en el sprint anterior, incluso si al cliente le había gustado.

### ***Sprint Retro (retrospectiva del sprint)***

*La Sprint Retro es una oportunidad para que el equipo Scrum se inspeccione a sí mismo y cree un plan para que se realicen mejoras durante el próximo sprint.*

El propósito de la retrospectiva del *sprint* es:

- Revisar cómo ha ido el último *sprint* en relación con las personas, relaciones, procesos y herramientas.
- Identificar los principales aspectos que han ido bien y potenciales mejoras.
- Crear un plan para activar las mejoras que ayuden al trabajo del equipo Scrum.

Las mejoras pueden estar centradas en obtener mejores resultados del producto, hacer más eficiente el proceso y las herramientas de trabajo o potenciar la motivación y relaciones del equipo. El SM promueve estas mejoras dentro del marco de Scrum con la intención de mejorar la satisfacción y efectividad del equipo.



El *Scrum Master* se asegura de que todos entiendan el propósito de la retrospectiva y de que sea una reunión positiva y productiva.

Aunque lo habitual es que el SM facilite esta reunión, también es positivo que otros miembros del equipo vayan rotando en el rol de facilitador, para que el SM pueda participar como miembro del equipo responsable del proceso de Scrum.

Si bien la mejora del equipo y su forma de trabajar debe suceder en todo momento, la retrospectiva es un momento formal para centrarse en la inspección y la adaptación.

## Artefactos de Scrum

Los **artefactos** son elementos físicos que se producen al aplicar Scrum. Representan el trabajo o valor que se está construyendo y proporcionan máxima transparencia de la información clave para que todos tengan el mismo entendimiento.

Los principales artefactos de Scrum que se describirán a continuación son:



Product Backlog



Sprint Backlog



Incremento

### ***Product Backlog***



*El Product Backlog es una lista ordenada de todos los requisitos que representan la visión del cliente y las necesidades del producto o proyecto. Es la única fuente de requerimientos para cualquier evolución y/o mejora que se deba hacer al producto.*

El orden de sus ítems está determinado por el **valor que aportan al cliente en relación con el riesgo y el coste estimado** para completarlo (priorización por valor de negocio y retorno de la inversión).

El PO es el responsable del *Product Backlog*, asegurando que esté disponible, priorizado y que su contenido esté debidamente detallado y entendido.

**El *Product Backlog* nunca está completo, es un artefacto vivo y dinámico.**

Al inicio es recomendable empezar solo con los requerimientos más prioritarios detallados y el resto de requerimientos que solo sean ideas identificadas, pero no detalladas. A medida que avancemos en el desarrollo del producto, iremos descubriendo más necesidades, priorizando y detallando al nivel necesario los ítems más prioritarios para poder trabajar en ellos.

### El *Product Backlog* y su proceso de refinamiento

El *Product Backlog* lista las características, funciones, requerimientos, mejoras y correcciones que representa el trabajo pendiente que debe ser entregado en futuros *sprints*. Es importante que estos ítems, además de tener una descripción, tengan unos criterios de aceptación claros que ayuden a comprobar que el trabajo está completo y cumple las expectativas cuando ya se haya hecho.

A su vez, estos ítems pasan por **procesos de refinamiento**. El refinamiento del *Product Backlog* consiste en añadir más detalle a los ítems más prioritarios, dividirlos en ítems más pequeños para que puedan ser realizados en un *sprint*, estimarlos y repriorizar.

Es responsabilidad del *Product Owner* que el *Product Backlog* esté bien refinado, pero es una actividad que se hace continuamente y en la que debe colaborar el equipo de desarrollo (no suele suponer más del 10% de su capacidad). Muchos equipos establecen como práctica hacer una **reunión de refinamiento** para asegurarse de que el refinamiento conjunto se hace y para interrumpir menos el desarrollo del *sprint*.

Una representación habitual del *Product Backlog* es la de un iceberg, donde los elementos que están en la superficie son más pequeños, claros y detallados, y a medida que entramos en la profundidad del *backlog* son más grandes y menos detallados, ya que son ideas para futuras entregas que deberán ser refinadas.



Los ítems más prioritarios que formarán parte del siguiente *sprint* están refinados hasta el punto en que estemos seguros que se podrán completar dentro de la duración del *sprint*. Estos ítems que tienen el suficiente detalle para poder trabajar en ellos y terminarlos dentro de un *sprint* se considera que están «listos» (*ready*).

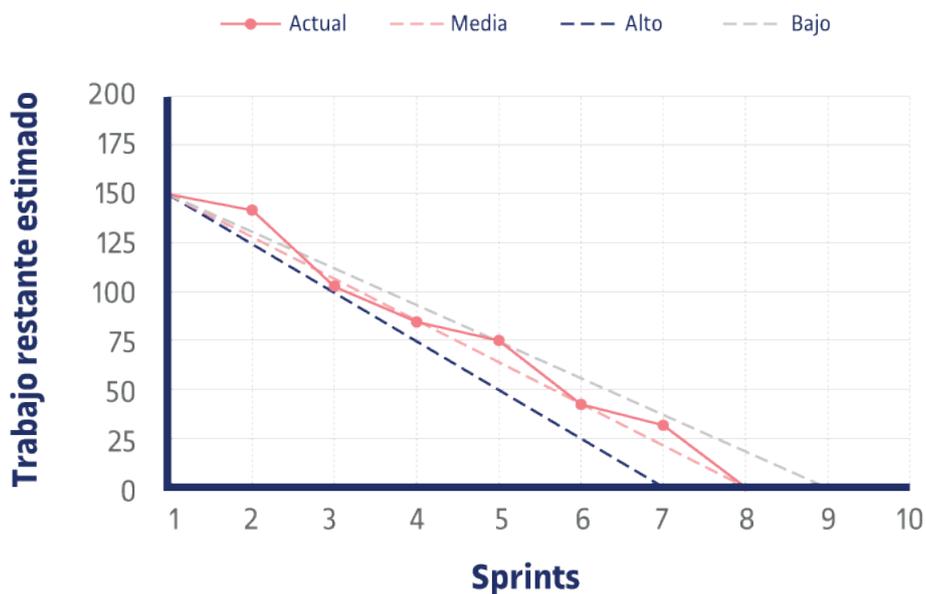
La estimación de los ítems que pueden quedar terminados en un *sprint* son responsabilidad del equipo de desarrollo, las personas que hacen el trabajo son las que lo estiman.



### Seguimiento del progreso del proyecto: gráfica *Release Burndown*

Dado que en todo momento del proyecto podemos sumar la cantidad de trabajo que queda pendiente en el *backlog*, el PO hace un seguimiento del trabajo pendiente como mínimo al finalizar cada *sprint* y compara esa cantidad con la de los *sprints* anteriores, para ver cómo la cantidad de trabajo pendiente va disminuyendo y tener una proyección de cuándo se terminará el proyecto o un hito en concreto. El PO hace visible esta información para todos los *stakeholders*.

Una de las formas de representar este seguimiento es con la **gráfica de *Release Burndown***, donde en el eje vertical está el trabajo restante (estimado habitualmente en puntos relativos) y en el eje horizontal los sucesivos *sprints*.



## Sprint Backlog



El *Sprint Backlog* es el conjunto de ítems del *Product Backlog* que el equipo seleccionó para el *sprint*, junto con el plan de tareas necesarias para completarlo y conseguir su objetivo.

El propósito del *Sprint Backlog* es mantener la **transparencia** dentro del desarrollo, proporcionando una visión del **trabajo que se va a realizar** durante el *sprint* actual.

Está gestionado por el equipo de desarrollo, que se encarga de mantenerlo actualizado y transparente durante toda la iteración.

Para mantener esta transparencia y ayudar al seguimiento del plan del *sprint* y la coordinación del equipo, el *Sprint Backlog* se gestiona de manera visual con un **tablero de tareas** (o **tablero kanban**). Este tablero proporciona una visión en tiempo real del trabajo del *sprint*, tanto para el equipo como para cualquier otra persona relacionada con el proyecto.



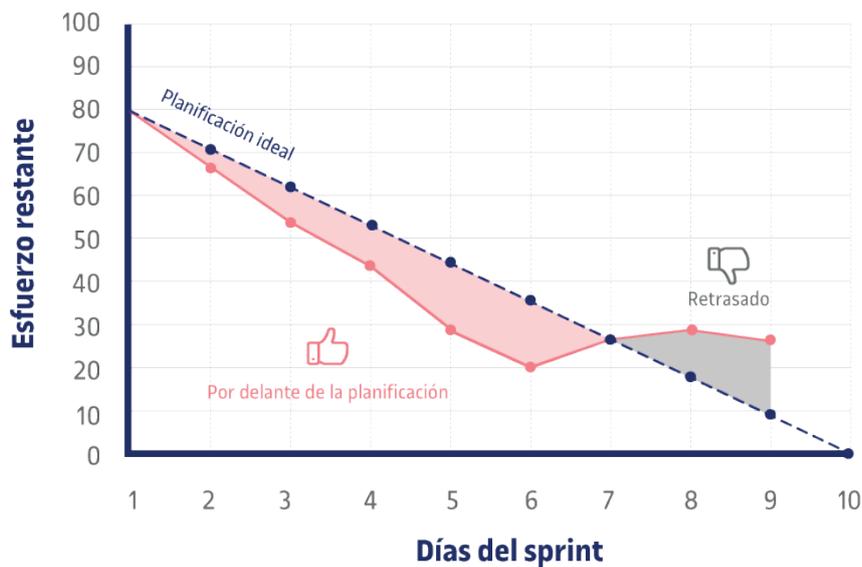
A medida que el equipo avanza en el *sprint*, el *Sprint Backlog* puede modificarse, ya sea porque el plan de tareas no estaba completamente detallado o porque surjan nuevas tareas que se deben realizar que no se habían contemplado en el *Sprint Backlog* durante el *Sprint Planning*.



### Seguimiento del progreso del sprint: gráfica *Sprint Burndown*

Como en cualquier momento del *sprint* la cantidad de trabajo pendiente puede ser sumada, el equipo de desarrollo hace un seguimiento de esta cantidad de trabajo pendiente por lo menos en cada *Daily Scrum* para poder proyectar la probabilidad de conseguir el objetivo del *sprint*.

La forma de visualizar esta información es con la *Sprint Burndown Chart*, que muestra la cantidad de trabajo pendiente en el eje vertical y los días del *sprint* en el eje horizontal.



### Incremento



*El incremento es la suma de todos los ítems del Product Backlog completados durante el sprint añadida a los incrementos de los sprints anteriores.*

*Es el resultado del sprint que se ve reflejado en un producto terminado que nos ayuda a construir nuestro producto o proyecto de manera iterativa e incremental.*

Cuando se dice que un incremento del producto o un ítem del *Product Backlog* está «hecho» (*done*), todos deben tener la misma idea de qué significa «hecho».

Este entendimiento compartido de qué significa que un ítem del *Product Backlog* esté hecho garantiza la transparencia, ayuda al equipo a estimar la cantidad de trabajo que podrá acometer en un *sprint* y sirve para evaluar si un incremento de producto está realmente «hecho».

# RESUMEN

Scrum es un **framework de trabajo** que se basa en el control de procesos empíricos y en los pilares de **transparencia, inspección y adaptación**.

Para reducir riesgos y poder gestionar el proyecto de forma empírica, se rige por una creación de producto iterativa e incremental. En cada iteración tendremos un incremento completo de producto que es candidato a ser entregado. Esta forma de trabajar reduce el riesgo del proyecto de forma implícita y da la posibilidad de adaptarse a las necesidades cambiantes.

Se definen **tres roles**: el *Product Owner*, el equipo de desarrollo y el *Scrum Master*. El PO es el «dueño del **qué**». El equipo de desarrollo es el «dueño del **cómo**», es un equipo multidisciplinar y autoorganizado de expertos, que son los responsables de construir el producto. El SM es quien vela por el proceso de Scrum, asegurando que todas las partes entiendan los valores, principios y prácticas de Scrum.

En las iteraciones (*sprints*), hay una serie de reuniones para planificar el trabajo de la iteración, coordinarse como equipo, revisar el resultado de la iteración y adaptar el plan, y para trabajar la mejora continua del equipo.

Cada elemento del *framework* Scrum está pensado para ayudar al equipo a centrarse en su trabajo.

# CONCEPTOS CLAVE

- ~ **Proceso empírico:** se entiende que el conocimiento proviene de la experiencia y la toma de decisiones está basada en lo que se conoce.
- ~ **Equipo multidisciplinar:** posee los conocimientos y habilidades necesarias para poder trabajar en una parte del producto de inicio a fin.
- ~ **Equipo autoorganizado:** elige la mejor manera de realizar su trabajo, en lugar de ser dirigidos por otros fuera del equipo.
- ~ **Desarrollo iterativo incremental:** incremento del producto terminado en una duración de tiempo determinada (*sprint*).
- ~ **Complejidad:** característica de proyectos o entornos con un alto grado de incertidumbre.
- ~ **Product Owner:** responsable de maximizar el valor del producto que se obtiene como resultado del trabajo del equipo de desarrollo.
- ~ **Scrum Master:** responsable de promover y ayudar a entender el correcto funcionamiento de Scrum, su teoría, prácticas, reglas y valores.
- ~ **Equipo de desarrollo:** conformado por las personas que realizan el trabajo necesario para poder entregar un incremento de producto terminado al finalizar el *sprint*.
- ~ **Sprint Planning:** definición del trabajo que se va a realizar durante el *sprint*.
- ~ **Sprint Review:** revisión del resultado del *sprint*.
- ~ **Retrospectiva:** momento en que el equipo se inspecciona a sí mismo y crea un plan de mejoras para accionar en el siguiente *sprint*.
- ~ **Daily Scrum:** reunión diaria de sincronización del equipo de 15 minutos.
- ~ **Product Backlog:** lista de ítems que contiene las necesidades y/o requerimientos del producto ordenados de mayor a menor prioridad.
- ~ **Sprint Backlog:** conjunto de ítems del *Product Backlog* que han sido seleccionados para el *sprint*, más el plan de cómo el equipo construirá el incremento para conseguir su objetivo.

# BIBLIOGRAFÍA

**Takeuchi, Hirotaka; Nonaka, Ikujiro** (1986). «[The New New Product Development Game](#)». *Harvard Business Review*.

**Schwaber, Ken** (1997). «[SCRUM Development Process](#)». En: J. Sutherland; C. Casanave; J. Miller; P. Patel; G. Hollowell (eds.). *Business Object Design and Implementation* (págs. 117-134). Londres: Springer.

[Scrum guide](#)

[Proyectos agiles.org](#)

[Scrum la guía definitiva](#)