

# Introducción a Scrum

- Introducción a Scrum: Historia, principios y valores de Scrum.
- ¿Qué es Scrum?
- Principios ágiles
- ¿Porqué emplear Scrum en el Aula?
- Pilares de Scrum
- Roles en Scrum
- Planificación y Ejecución de Sprints
- ▼ Reuniones de Scrum
  - Reunión diaria o Daily Scrum
  - Revisión del Sprint (Sprint Review)
  - Retrospectiva del Sprint (Sprint Retrospective)

## TO DO

- ☐ Intro Scrum.
- ☐ Dinámica
- ☐ Equipos
- ☐ Product Backlog Priorizado con las Historias de usuario en Aules

# Introducción a Scrum: Historia, principios y valores de Scrum.

## ¿Qué es Scrum?

Scrum es un marco que permite el **trabajo colaborativo** entre equipos. Al igual que un equipo de rugby (de donde proviene su nombre) cuando entrena para un gran partido, scrum anima a los equipos a aprender a través de las experiencias, a autoorganizarse mientras aborda un problema y a reflexionar sobre sus victorias y derrotas para mejorar continuamente.

Normalmente son los equipos de desarrollo de software los que utilizan con mayor frecuencia este tipo de scrum, pero sus principios y lecciones se pueden aplicar a todo tipo de trabajo en equipo.

Scrum incluye un conjunto de reuniones, herramientas y funciones que, de forma coordinada, ayudan a los equipos a estructurar y gestionar su trabajo. El objetivo es obtener mejores resultados cada vez, dándole agilidad al equipo, que mejor con cada experiencia, y la adaptación al cambio, al ser un proceso iterativo.



Scrum fue creado por Jeff Sutherland y Ken Schwaber a principios de los años 90. Ambos estaban trabajando en el desarrollo de software y se dieron cuenta de que los métodos tradicionales de

gestión de proyectos no eran efectivos. Así que se unieron para crear un marco de trabajo que permitiera a los equipos ser más ágiles y adaptarse a los cambios.

## Principios ágiles

Hoy en día definimos la filosofía Agile como la capacidad de crear y responder al cambio de forma rápida, es una forma de llevar a cabo un proyecto de forma exitosa en un entorno incierto y en constantes cambios (Agile Alliance, 2015).

Los principios ágiles son un conjunto de valores y principios que se utilizan para guiar el desarrollo de software.

Los principios ágiles se basan en el Manifiesto Ágil, que fue escrito por un grupo de desarrolladores de software en 2001. El manifiesto ágil establece cuatro valores fundamentales y doce principios que se utilizan para guiar el desarrollo de software.

Los cuatro valores fundamentales del manifiesto ágil son:

- **Individuos e interacciones sobre procesos y herramientas:** Los equipos ágiles valoran más a las personas y las relaciones entre ellas que a los procesos y las herramientas. Esto significa que los equipos ágiles se centran en la comunicación y la colaboración entre los miembros del equipo, en lugar de en los procesos y las herramientas que utilizan.
- **Software funcionando sobre documentación extensiva:** Los equipos ágiles valoran más el software que funciona que la documentación extensiva. Esto significa que los equipos ágiles se centran en crear software que funcione y que sea útil para los usuarios, en lugar de en crear una gran cantidad de documentación.
- **Colaboración con el cliente sobre negociación contractual:** Los equipos ágiles valoran más la colaboración con el cliente que la negociación contractual. Esto significa que los equipos ágiles trabajan en estrecha colaboración con los clientes para entender sus necesidades y crear software que satisfaga esas necesidades, en lugar de centrarse en la negociación de contratos.
- **Responder al cambio sobre seguir un plan:** Los equipos ágiles valoran más la capacidad de responder al cambio que seguir un plan. Esto significa que los equipos ágiles se centran en ser flexibles y adaptarse a los cambios en lugar de seguir un plan rígido.

De estos valores surgen los doce principios del manifiesto ágil.

Para realizarlos de forma más genérica y aplicado a nuestro contexto, los planteo adaptado al contexto general, y no solo al desarrollo de software:

1. La prioridad es satisfacer al cliente (docente) a través de la entrega temprana y continua con **valor**.
2. Aceptamos que los requisitos cambien.
3. Entregamos frecuentemente, cuanto más frecuente mejor.
  4. Los responsables de negocio (docentes) y los desarrolladores (alumnado) trabajamos juntos **de forma cotidiana durante todo el proyecto**. Docentes y alumnado tiene los mismos objetivos.
  5. Los proyectos se desarrollan en torno a **individuos motivados**. Hay que darles el entorno y el **apoyo** que necesitan, y confiarles la ejecución del trabajo.
4. El método más eficiente y efectivo de **comunicar** la información es **cara a cara**.
5. El **producto** que funciona es la medida principal del progreso.
6. Los procesos ágiles promueven el desarrollo sostenible. Debemos ser capaces de mantener un **ritmo constante** de forma indefinida.
7. La atención continua a la **excelencia técnica** y al buen diseño mejora la agilidad.
8. La **simplicidad**, o el arte de maximizar la cantidad de trabajo no realizado, es esencial.
9. Las mejores soluciones emergen de **equipos autoorganizados**. *Se les da la **libertad** para que tomen decisiones, y esto les hace estar más comprometidos.*
10. A **intervalos regulares** el equipo **reflexiona** sobre cómo ser más efectivo para a continuación ajustar y perfeccionar su comportamiento en consecuencia. *Hay que aprender de los errores.*

## ¿Porqué emplear Scrum en el Aula?

1. **Fomenta la autonomía y la responsabilidad**. El alumnado se organiza y planifica su trabajo, asume responsabilidades y toma decisiones.
2. **Fomenta la colaboración y el trabajo en equipo**. El alumnado trabaja en equipo, se apoya mutuamente y comparte conocimientos.
3. **Fomenta la creatividad y la innovación**. El alumnado tiene la oportunidad de proponer soluciones y de experimentar con nuevas ideas.
4. **Fomenta la comunicación y la empatía**. El alumnado aprende a comunicarse de forma efectiva y a ponerse en el lugar de los demás.
5. **Fomenta la adaptación al cambio**. El alumnado aprende a ser flexible y a adaptarse a las circunstancias cambiantes.
6. **Fomenta la reflexión y la mejora continua**. El alumnado reflexiona sobre su trabajo y busca maneras de mejorar.
7. **Fomenta la motivación y el compromiso**. El alumnado se siente motivado y comprometido con su trabajo.
8. **Fomenta la responsabilidad y la toma de decisiones**. El alumnado asume responsabilidades y toma decisiones sobre su trabajo.

9. **Fomenta la organización y la planificación.** El alumnado se organiza y planifica su trabajo de forma efectiva.
10. En el entorno de FP, fomenta la **empleabilidad**. El alumnado adquiere habilidades que son valoradas por las empresas.

## Pilares de Scrum

Scrum se basa en tres pilares fundamentales que son la transparencia, la inspección y la adaptación.

- **Transparencia:** Todos los aspectos del proceso deben ser visibles para todos los miembros del equipo. Esto significa que el equipo debe ser honesto y abierto acerca de su trabajo y de los problemas que puedan surgir.
- **Inspección:** El equipo debe inspeccionar regularmente su trabajo para identificar posibles problemas y oportunidades de mejora.
- **Adaptación:** El equipo debe adaptarse a los cambios en su entorno y en su trabajo. Esto significa que el equipo debe ser flexible y estar dispuesto a cambiar su forma de trabajar si es necesario.

Scrum por tanto se basa en realizar iteraciones cortas, llamadas sprints, en las que se planifica, se ejecuta y se revisa el trabajo realizado.

## Roles en Scrum

### Product Owner

Dentro del aula, el Product Owner es el/la docente.

Es la persona responsable de definir lo que se debe realizar a través de historias, priorizarlas y asegurarse de que el equipo de desarrollo conoce los requisitos del proyecto. El Product Owner es la persona responsable de maximizar el valor del trabajo del equipo de desarrollo.

### Scrum Master

Es el/la responsable de asegurarse de que el equipo sigue los valores y prácticas de Scrum. El/la Scrum Master es la persona responsable de ayudar al equipo a ser más efectivo y de eliminar los obstáculos que puedan estar impidiendo que el equipo tenga éxito.

Se trata de un/a facilitador/a, no de un/a “coordinador/a” o “jefe/a del equipo”. Se trata de un miembro igualitario del equipo.

El Scrum Master puede ser inicialmente el/la docente, en equipos con poca autonomía. En equipos con mayor autonomía, el/la Scrum Master puede ser un alumno o alumna, y que sea el propio equipo el que decida quién es el Scrum Master.

## Equipo de Desarrollo

El equipo de desarrollo es el responsable de crear el producto. El equipo de desarrollo es un equipo autoorganizado y multifuncional que trabaja en estrecha colaboración con el/la Product Owner para crear el producto. El equipo de desarrollo es responsable de determinar cómo se va a realizar el trabajo y de llevarlo a cabo.

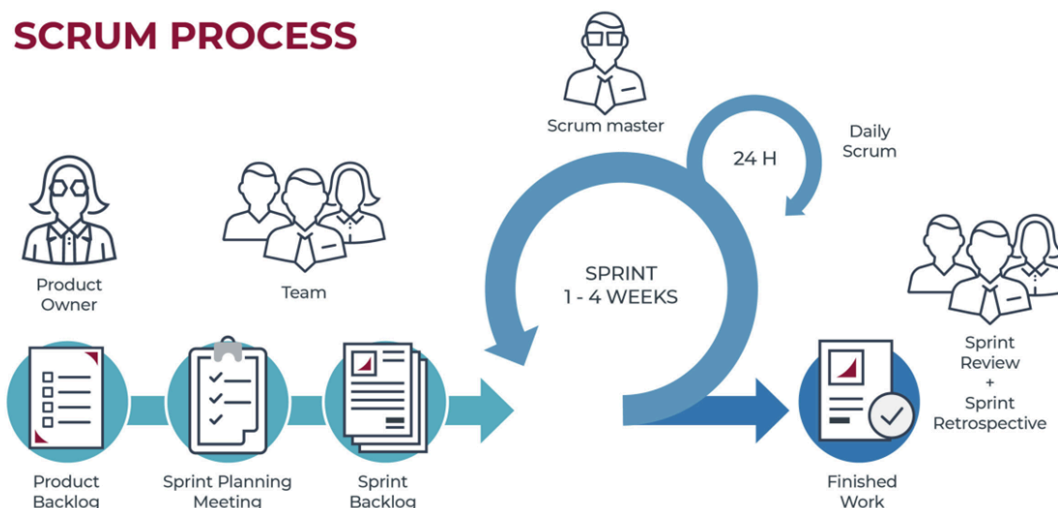
El equipo debe estar:

- Preparado
- **Comprometido**
- Con capacidad de obtener resultados
- **Autoorganizado**

Lo ideal son equipos de 4-6 personas, pero esto dependerá de las necesidades del proyecto.

¿Es aconsejable cambiarlos o mantenerlos? Normalmente es recomendable mantener el equipo estable. Pero dependerá de cómo hayan trabajado, a veces es recomendable cambiar el equipo para el siguiente proyecto.

## Planificación y Ejecución de Sprints



# Product Backlog

El Product Backlog es una lista de todas las tareas que se deben realizar para completar el proyecto. El Product Owner es el responsable de mantener el Product Backlog actualizado y de priorizar las tareas en función de su importancia. Es **dinámica**, y se va actualizando a lo largo del proyecto.

Será el primer paso en nuestro proyecto. El/la docente deberá definir las **historias de usuario** que se deben realizar para completar el proyecto.

Una historia de usuario es una descripción de una funcionalidad que se debe implementar en el proyecto. Las historias de usuario se utilizan para definir los requisitos del proyecto y para comunicar las necesidades al equipo.

Se escriben en un lenguaje sencillo y se centran en las necesidades. Por ejemplo, una historia de usuario podría ser: “Como usuario, quiero poder iniciar sesión en la aplicación para acceder a mi perfil”.

“Como [perfil], [quiero] [para].”

El listado con todas las **historias** se denomina **Product Backlog**. Y es el/la docente, ejerciendo su papel de Product Owner, la persona que debe priorizarlas.

Este listado se irá actualizando a lo largo del proyecto, añadiendo nuevas historias, modificando las existentes, o eliminando aquellas que ya no sean necesarias.

Cada historia de usuario puede tener:

- Nombre
- Descripción: en forma de historia de usuario
- Prioridad
- Estimación: en puntos de historia. No estimar en tiempo. Lo veremos más adelante.
- Criterios de aceptación: qué debe cumplir para que se considere terminada.

No es necesario escribir las historias de usuario con este formato. Se puede hacer una lista de requerimientos cómo mejor se adapte a la forma de trabajar del equipo.

Cada historia de usuario o requerimiento debe ir acompañado de sus **criterios de aceptación**. Los criterios de aceptación son una lista de condiciones que se deben cumplir para que la historia de usuario se considere terminada. Los criterios de aceptación se utilizan para asegurarse de que el equipo entiende lo que se espera de ellos.

Por ejemplo, este historia de usuario:

"Como usuario, quiero poder iniciar sesión en la aplicación para acceder a mi perfil"

Tendría los siguientes criterios de aceptación:

- El usuario debe poder introducir su nombre de usuario y su contraseña.
- El usuario debe poder hacer clic en un botón para iniciar sesión.
- El usuario debe ser redirigido a su perfil después de iniciar sesión.
- El usuario debe ver un mensaje de error si introduce un nombre de usuario o una contraseña incorrectos.

No hay que confundir los criterios de aceptación de cada tareas, con la **Definición de Hecho** ("Definition of Done") que se verá más adelante. La definición de hecho es la misma para todas las historias o requerimientos, y los criterios de aceptación son específicos de cada tarea.

TODO: Definición de Hecho

TODO: Definición de Divertido (más utilizada en eduScrum)

Ejemplo de definición de hecho:

- La tarea ha sido revisada y comprendida por todos los miembros del equipo.
- Cumple con los criterios de aceptación.
- Se ha documentado

TODO: producto mínimo viable

## Sprint

El Sprint es el corazón Scrum. Un sprint es un periodo de tiempo en el cuál se deben lograr una entrega que proporcione valor, utilizando todos los materiales y recursos que se ponen a disposición del equipo.

El Sprint tiene una duración fija, que suele ser de entre 1 y 4 semanas. Durante el Sprint, el equipo trabaja para completar las tareas que se han seleccionado del Product Backlog.

## Sprint Planning

Debe asistir el equipo al completo: el Product Owner, el Scrum Master y el equipo de desarrollo. ¿Qué vamos a acabar en este Sprint? Es el objetivo del Sprint. El Product Owner debe explicar qué es lo que se espera que se entregue al final del Sprint, y priorizar lo que es más importante.

El equipo de desarrollo selecciona las tareas del Product Backlog más prioritarias que considera va a poder realizar en el Sprint, y se planifica cómo se van a realizar. Debe estar claro que día va a finalizar el Sprint, y no se debe modificar.



El **Sprint Backlog** es una lista de todas las tareas que se deben realizar durante el Sprint. El equipo de desarrollo es el responsable de crear el Sprint Backlog y de asegurarse de que todas las tareas se completan a tiempo. Este listado de tareas no se puede cambiar durante el Sprint.

Para ver cuántas tareas se pueden realizar en un Sprint, hace falta saber cuántas personas hay en el equipo y cuanto tiempo se va a trabajar en el Sprint, si a tiempo completo, al 20% o al 50%.

El Scrum Master debe asegurarse de que el equipo entiende lo que se espera de ellos y de que tienen todo lo que necesitan para completar el trabajo. Debe asegurarse de que todo el equipo sale de la reunión sabiendo cuáles son las tareas que se deben realizar en el Sprint. Se pueden asignar que tareas va a realizar cada uno en esta reunión, pero no es necesario. Lo que si se debe hacer es estimar el esfuerzo que va a suponer cada tarea. Esto se puede hacer con la técnica de **poker planning**.

El Product Owner debe resolver preferiblemente en este momento las dudas que surjan sobre las tareas que se van a realizar.

Para que la planificación del Sprint sea efectiva, es importante desglosar las tareas en tareas más pequeñas, que se puedan realizar en un día, o máximo dos. También es importante saber si hay condicionantes, es decir, si para empezar una tarea debe estar terminada otra. Esto lo puede indicar el Product Owner, o lo puede indicar el equipo de desarrollo.

## Reuniones de Scrum

Scrum define las reuniones que se deben realizar: ¿Cuándo? ¿Quién debe asistir? ¿Qué se debe hacer en cada reunión?

### Reunión diaria o Daily Scrum

Tiene los siguientes objetivos:

- Comentar el trabajo realizado desde la última reunión.
- Definir el trabajo que se va a realizar hasta la siguiente sesión.
- Identificar los posibles problemas que han surgido

**NO** debe durar más del tiempo establecido, pero si puede durar menos. En Scrum son 15 min, pero en el aula dependerá del tiempo que tengamos de la sesión. Pensad que si son sesiones de 55 minutos, no puede durar más de 10 minutos. Es responsabilidad del Scrum Master que no se pase de ese tiempo.

Se recomienda hacerla en el mismo lugar y a la misma hora.

## **Preguntas**

- ¿Qué he hecho desde la última reunión?
- ¿Qué me comprometo a hacer hasta la próxima reunión?
- ¿Hay algún problema que me impida avanzar?

El Scrum Master es el responsable de informar de las acciones que ha realizado, y el estado en el que se encuentran, acerca de la resolución de los problemas que se han detectado en la reunión anterior.

## **Ventajas**

Mejora el conocimiento del estado del proyecto, y la comunicación entre los miembros del equipo.

Fomenta cierta competitividad, ya que se establece un compromiso con el resto del equipo, y nadie quiere decir que no ha avanzado nada.

Ayuda a generar vínculos con los miembros del equipo.

## **Revisión del Sprint (Sprint Review)**

Es el momento en el que el Product Owner valida el trabajo realizado por el equipo. Se debe mostrar el trabajo realizado, y el Product Owner debe validar que cumple con los requisitos que se habían establecido.

Deben venir todos, el Product Owner, el Scrum Master y el equipo de desarrollo. También pueden venir invitados, como otros profesores, o incluso el alumnado de otros equipos. O personas interesadas, que vayan a utilizar el producto.

El Product Owner puede proponer mejoras o rectificaciones. Esto nos va a modificar la lista de tareas pendientes.

Las ventajas de esta reunión son:

- Permite al equipo mostrar su trabajo
- Obtener el equipo feedback del Product Owner
- El Product Owner detecta desviaciones entre lo que deseaba obtener y lo que le muestran
- Permite preparar la replanificación del proyecto con la máxima información posible.

## **Lista de incremento de producto**

Se puede tener una lista con todas las tareas finalizadas. Es el Product Owner quien se encarga de mantenerla actualizada, y decidir si se da por terminada una tarea o no.

# Retrospectiva del Sprint (Sprint Retrospective)

El objetivo es autoevaluar la forma de trabajar. Se habla de la forma en la que el equipo ha actuado, de cómo ha afrontado los problemas, y de cómo debería haberlos afrontado.

Es recomendable hacerlo después de cada Sprint, pero no es obligatorio. Se puede hacer después de varios Sprints, pero no es recomendable hacerlo después de más de 3 Sprints.

Depende de si la duración del Sprint es corta.

¿Para que repetir errores antiguos habiendo tantos errores nuevos que cometer? Bertrand Russell

Cita siempre los errores propios antes de referirte a los ajenos. Así nunca parecerá que presumes. Noel Clarasó

Se debe asumir las críticas con profesionalidad, y no tomarlas como algo personal. Se debe buscar soluciones a los problemas que se han detectado.

El Scrum Master debe asegurarse de que las críticas son constructivas, y de que se buscan soluciones a los problemas que se han detectado. No se debe buscar culpables, ni usar frases como "tú has cometido este fallo". Se debe plantear con frases como "cómo podemos evitar que se cometa ese fallo". También debe encargarse de que no sólo se hable de los errores. Si hay cosas que se han hecho bien, hay que ponerlo en valor y procurar que se siga haciendo así.

Hay que sacar al *elefante* (problema grande) de la sala. El elefante es que todos ven el problema, pero nadie se atreve a decirlo. Si se consigue sacar al elefante de la sala, se habrá conseguido que el equipo se vea más capaz de resolver los conflictos.

Si se desea hablar de un problema hay que:

- Tener claro cual es el problema, y comprender a que miembros afecta y en que medida.
- Conocer a que miembros del equipo les puede afectar emocionalmente que se hable de ellos.
- Planificar adecuadamente la exposición del problema (indicando claramente lo que supone), así como propuestas de solución.