

METODOLOGÍA

Guía Agile: conceptos imprescindibles para conocer la Filosofía



En este artículo te voy a explicar qué es el Agile y el significado de las palabras imprescindibles que se utilizan en el día a día en los proyectos Agile.

En muchas ocasiones nos dejamos llevar por la emoción y empezamos a utilizar palabras muy técnicas o siglas. Creamos barreras que nos impiden llegar a todos nuestros interlocutores, es más complicado mantener la atención y que el mensaje llegue correctamente.




18 MIN. DE LECTURA

Glosario Agile [orden alfabético]:

A

-  Agile
-  Agile Coach
-  Artefactos

D

-  Definition of Done
-  Definition of Ready
-  Development Team

I

-  Incremento



T

-  Team agreements o Normas de equipo

B

-  Burn-down Chart

E

-  Épica
-  Estimación







P

-  Product Backlog
-  Product Owner

V

-  Velocidad
-  Visión

C

-  Ceremonias:
 -  Sprint Planning
 -  Daily
 -  Refinement
 -  Sprint Review
 -  Sprint Retrospective

H

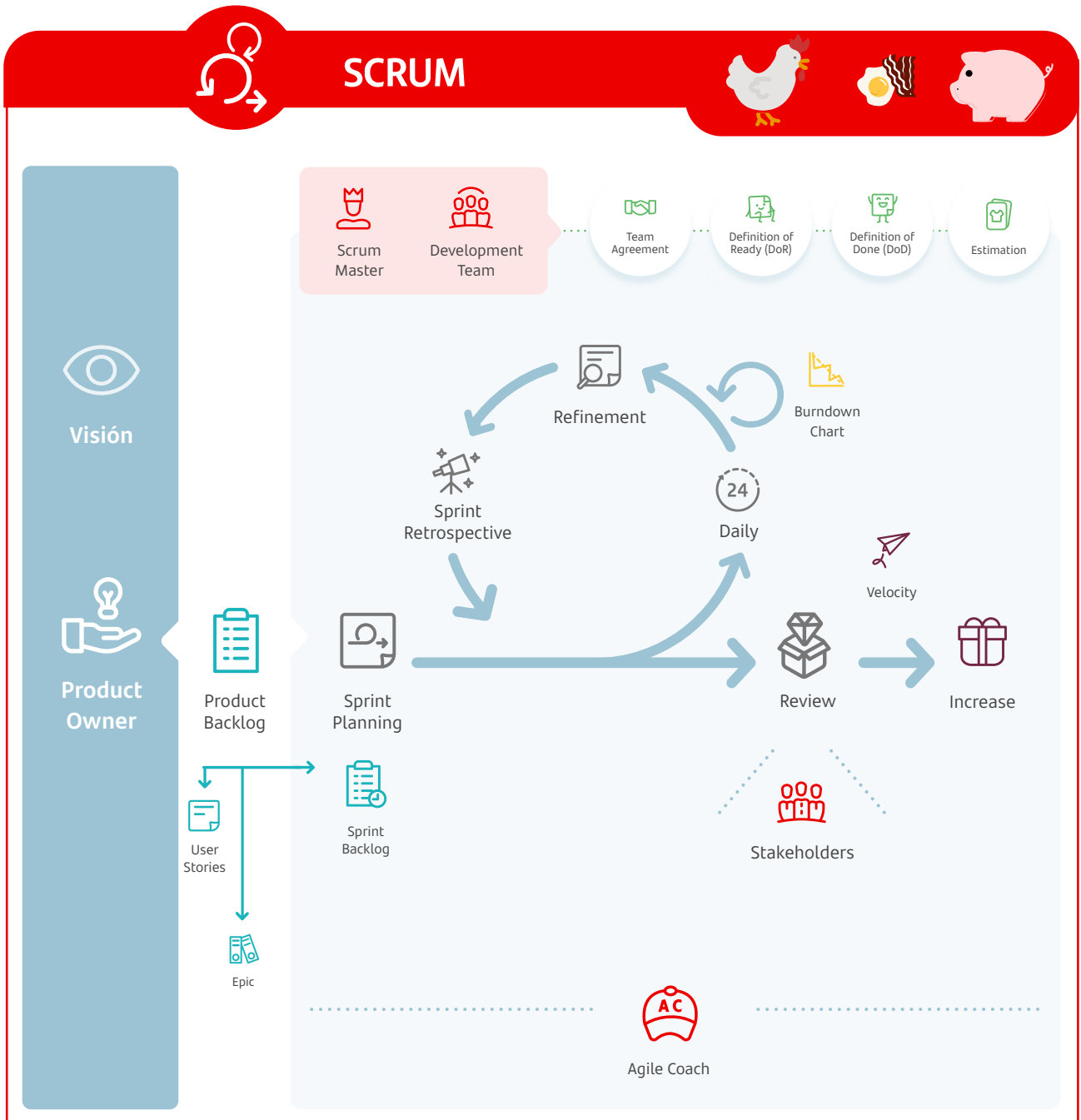
-  Historias de Usuario
-  Historias de Usuario de Referencia

S

-  Scrum
-  Scrum Master
-  Sprints
-  Stakeholder

Aquí tienes una visión general de todo lo que vas a conocer en este post: cómo se trabaja en Agile, quién interviene y los conceptos que necesitas para desarrollar un proyecto.

Puedes guardarte esta imagen, te aseguro que te ayudará a tener un esquema completo del trabajo en Agile.



Qué es (y qué no es) el Agile

Agile es una **filosofía** o **movimiento**, no es una metodología. Es una alternativa a la gestión de proyectos tradicional en *waterfall*.

Para los proyectos en Agile, se forma un **equipo** que colabora de forma muy intensa en **pequeñas iteraciones** o **Sprints**, que derivan en actualizaciones concretas del producto o servicio.

Proporciona un mecanismo de **retroalimentación constante** sobre la evolución. Y así, **aporta valor** al cliente desde la **1ª iteración**.

Abarca varios métodos, sobre los que podemos crear o adaptar nuestros propios procesos: Scrum, Lean, Kanban, XP, ...

La filosofía Agile se basa en 4 valores:

- **Individuos e interacciones** por encima de procesos y herramientas
- **Software funcionando** por encima de documentación exhaustiva
- **Colaboración con el cliente** por encima de la negociación contractual
- **Respuesta ante el cambio** por encima de seguir un plan

No se hace Agile para el cliente sino **CON el cliente**.



Además de los 4 valores, el **Manifiesto Agile** consta de 12 principios:

1. Nuestra principal prioridad es satisfacer al cliente a través de la entrega temprana y continua de software con valor.
2. Aceptamos que los requisitos cambien, incluso en etapas tardías del desarrollo. Los procesos ágiles aprovechan el cambio para proporcionar ventaja competitiva al cliente.
3. Entregamos software funcional frecuentemente, entre dos semanas y dos meses, con preferencia al período de tiempo más corto posible.
4. Los responsables del negocio y los desarrolladores trabajamos juntos de forma cotidiana durante todo el proyecto.
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.
6. El método más eficiente y efectivo de comunicar información al equipo de desarrollo y entre sus miembros es la conversación cara a cara.

7. El software funcionando es la medida principal de progreso.
8. Los procesos ágiles promueven el desarrollo sostenido. Los promotores, desarrolladores y usuarios debemos mantener un ritmo constante de forma indefinida.
9. La atención continua a la excelencia técnica y al buen diseño mejora la agilidad.
10. La simplicidad, o el arte de maximizar la cantidad de trabajo no realizado, es esencial.
11. Las mejores arquitecturas, requisitos y diseños emergen de equipos auto-organizados.
12. A intervalos regulares, el equipo reflexiona sobre cómo ser más efectivo para, a continuación, ajustar y perfeccionar su comportamiento en consecuencia.



Agile Coach

Es un **formador, mentor, entrenador o facilitador**. Ayuda a las personas a **repensar** y **cambiar** la forma en que se **desarrollan**. Motivan el cambio y lo hacen posible.

- Evaluar la **madurez** de la **organización** en **Agile** y guiarla a niveles más altos de agilidad y **efectividad**, a un **ritmo** que sea **sostenible**
- Hacer todo lo posible para que los **equipos** sean **auto-organizados** en su trabajo
- Implantar **prácticas** que promuevan la **comunicación** y que mejoren la toma de **decisiones** y la **resolución de conflictos**
- Promover el uso de **herramientas colaborativas** para facilitar la **transparencia** y la **sincronización** del trabajo.
- Trabajar para **eliminar impedimentos** para aumentar la **agilidad organizacional**
- Construir un **entorno seguro donde** los problemas se puedan plantear sin miedo, con énfasis en la **colaboración** y la **resolución**
- **Facilitar** el **trabajo sin imposición**, asignar o dictar cómo se entrega
- Ayudar con la **comunicación interna y externa**, mejorar la transparencia y compartir información
- **Apoyar y educar al Product Owner** de Transformación Agile, particularmente con respecto a refinar y gestionar el Product Backlog

- **Trabajar con Scrum Masters** / Facilitadores agile, para guiar a la organización en general sobre cómo usar prácticas y valores ágiles, y promover su uso
- Realizar **sesiones de capacitación** y **talleres** para equipos ágiles

Requiere alto nivel de **facilitación, training, leadership, resolución** de conflictos y tener **experiencia previa**: mínimo 3 años como Scrum Master.



Artefactos

Los medios y herramientas que utilizamos para **representar trabajo o valor**. Sirven para tener transparencia e identificar oportunidades de mejora: **Product Backlog**, **Sprint Backlog**, **Seguimiento del Progreso del Sprint** e **Incremento**.



Burn-down Chart: Trabajo pendiente

Gráfico para ver el **progreso del Sprint**: puntos del **trabajo realizado** y **pendiente** en una línea temporal que nos permite ver cuánto **tiempo** falta para terminar el **Sprint**.

Esta gráfica se puede consultar en la herramienta digital que tengas (Jira, Taiga,...) o se puede calcular con los datos de la pizarra física.


























Incluye:

- Diagrama que refleja el tiempo **disponible** para **completar** el trabajo del **Sprint**
- Eje **X**: días de **duración del Spint**
- Eje **Y**: cantidad de **trabajo comprometida**
- Si es inferior a la línea ideal, va en buen camino para alcanzar el Sprint Goal.

Existen otras gráficas para ver el progreso del equipo. El **Equipo de Desarrollo** hace seguimiento de este trabajo restante, por lo menos en cada **Daily** para gestionar su progreso y asegurar que se cumple el objetivo del **Sprint**.

Ceremonias

Las reuniones de los equipos Scrum en Agile se llaman Ceremonias. Son: Sprint Planning Meeting, Daily Scrum, Product Backlog Refinement, Sprint Review y Sprint Retrospective.

Sprint Backlog	Daily	Refinement	Sprint Review	Sprint Retrospective
				
¿Qué será desarrollado ? ¿Cuál es la prioridad ?	El equipo responde a 3 preguntas : ¿Qué hice? ¿Qué haré hoy? ¿Algún bloqueo?	Revisión y redifinición de las funcionalidades priorizadas.	Muestran el trabajo realizado , software funcionando	Impedimentos / mejoras , acciones a seguir.
 Product Owner  Development Team  Scrum Master	 Scrum Master  Development Team	 Product Owner  Development Team  Scrum Master	 Product Owner  Development Team  Scrum Master  Stakeholders	 Product Owner  Development Team  Scrum Master
 Máx: 2 h/semana del Sprint	 Máx: 15 min	 Máx: 10 % total Sprint	 Máx: 1 h/semana del Sprint	 Máx: 1 h/semana del Sprint

CEREMONIAS AGILE

Sprint Planning

Es la reunión **inicial** en cada comienzo de **Sprint**. El objetivo de la reunión es concretar dos cosas: ¿Qué será **desarrollado**? y ¿Cuál es la **prioridad**?

- Reunión inicial del Sprint, en la que el **Product Owner** discute el objetivo del Sprint con el **Scrum Team** y el **Scrum Master**
- Llegan a un pacto de **objetivos y trabajo**
- El **equipo estima el trabajo** que puede completar en el Sprint de los elementos priorizados en el **Product Backlog**

Duración: máximo **2 horas** por semana del sprint

Participantes: Product Owner, Scrum Master, Scrum Team



Daily

Breve **reunión diaria** del equipo + identificación de **impedimentos**.

Cada miembro del equipo responde a **tres preguntas**:

- ¿Qué **hice ayer** que ayudo a cumplir el objetivo del Sprint?
- ¿Qué **hare hoy** para ayudar a cumplir el objetivo del Sprint?
- ¿Veo **impedimentos** para que cumplamos el objetivo del Sprint?

Duración: máximo **15 min**

Participantes: Development Team, Scrum Master



Refinement

Revisión y redefinición de las **funcionalidades** priorizadas.

- Revisión y redefinición del **Product Backlog** a partir de nuevas informaciones / requerimientos
- **Re-estimación** de los elementos existentes basada en nueva información anteriormente desconocida
- **Añadir** nuevos ítems Product Backlog o **suprimir** algunos existentes
- **Dividir** ítems extensos en otros menores (**Epics** -> **User Stories**)
- **Estimación y priorización** de nuevos elementos
- **Revisar** si las historias de usuario priorizadas **cumplen Definition of Ready (DoR)**, con el objetivo de **trabajar** solo las **historias** que dan más **valor a negocio**.



Sprint Review

El Scrum Team **muestran** el **trabajo realizado**, es decir: software funcionando.

- Reunión estratégica para el análisis del **incremento completado** por el equipo.
- El equipo **muestra** el **incremento** resultado del **Sprint** a los **Stakeholders**, que ha sido validado previamente por el **Product Owner**
- Es imprescindible tener **software funcionando**

- El objetivo es obtener feedback del cliente y el usuario
- El Product Owner y los Stakeholders actualizan el **Product Backlog** para el siguiente Sprint
- Marcar la **estrategia** de negocio

Duración: máximo **1 hora** por semana del Sprint.

Participantes: Development Team, Scrum Master, Product Owner, Stakeholders (Opcional)



Sprint Retrospective

Es la reunión final de Sprint en la que **reflexionamos** como hemos trabajado, identificamos impedimentos / áreas de oportunidad y acciones a seguir.

- Reunión **gestión interna** del equipo donde se analiza el comportamiento y colaboración de los miembros y se buscan oportunidades de mejora
- Se discute qué ha salido **bien**, qué se podría haber hecho **mejor** y qué se puede **mejorar** en el próximo **Sprint**
- Todos los miembros deben hablar y expresar inquietudes
- Se busca **identificar mejoras** en el proceso de trabajo que se puedan poner en práctica de forma inmediata
- Método: recolectar **información + proponer ideas** que se trasladan en un **plan de acción** para evolucionar como equipo

Duración: máximo **1 hora** por semana del Sprint

Participantes: Development Team, Scrum Master, Product Owner



Definition of Done (DoD)

El equipo Scrum define unos **critérios** que se tienen que **cumplir** para poder para considerar un elemento (o ítem) del **Product Backlog** como **terminado**.

Se acuerda entre el **Product Owner** y el **Equipo de desarrollo**, antes de la primera interacción y se puede ir mejorando durante el proyecto.



Definition of Ready (DoR)

Un conjunto de condiciones que tiene que cumplir una **historia de usuario** para ser tomada en un **Sprint Planning**.

Se acuerda entre el **Product Owner** y el **Equipo de desarrollo**, antes de la primera interacción y se puede ir mejorando durante el proyecto.



Development Team o Scrum Team

Decide **CÓMO** será construido el producto y además lo construye. Se **auto gestiona, organiza y toma sus propias decisiones**.

- **Deciden** junto con el **Product Owner** el alcance de la release
- **Aceptan** los **elementos** del **Product Backlog** como “**Listos**” para coger
- Deciden lo que **entra** en el **Sprint**
- Se **auto-organizan** y trabajan **comprometidos** para conseguir los **objetivos** del **Sprint**
- Deciden **quién trabaja** en las distintas **tareas** del **Sprint**
- **Deciden cómo** construir el producto
- **Entregan incrementos** del **producto** en cada Sprint
- Hacen **visibles** los **impedimentos** que le **bloquean** para conseguir el **objetivo** del sprint
- **Inspección y adaptación** para **mejorar** la forma de **trabajar**
- **Construyen** el “**Definition of Done**”

- **Construyen** el “**Definition of Ready**”
- **Estiman** los **ítems** del **Product Backlog**
- **Crean** el **Sprint Backlog** y lo mantienen **actualizado**

El tamaño del equipo ideal es entre 3 y 9 miembros.



Épica

Es un nivel de **agrupación superior** a los **requisitos funcionales** que permite **clasificar** los requisitos (**historias de usuario**).



Estimación

Estimación de **esfuerzo relativo** que requiere cada **User Story**.

Existen diferentes modelos de estimación: Planning poker, Tallas de camiseta, PERT (Program Evaluation and Review Technique), Estimación en la pared,...

1. El **equipo** elige un **método de estimación** relativo.
2. Se **detalla** una **historia** (inventada por el equipo) y por consenso se decide su valor (**historia de referencia**), será la unidad de medida que tendrá el equipo.
3. Por **afinidad** en base a la historia de referencia de estiman el **resto**.

En la estimación **participarán todos los miembros del Development Team** y se tiene que estimar por **consenso**. Si hay discrepancias, los miembros del equipo argumentarán y volverán a estimar hasta llegar a un consenso.

Ejemplo: ¿Cuánto pesa una manzana? Igual no tengo ni idea, pero conozco lo que pesa un melocotón: en este caso 5. Ahora, por afinidad podría estimar que pesa lo mismo. Si me preguntan cuánto pesa una pera sería la mitad, y una cereza sería bastante menos... Sencillo, no?

Historias de Usuario o User Stories

Es el listado de requisitos, también llamados ítems, que quedan agrupados en el **Product Backlog**, que necesitamos para desarrollar nuestro producto / servicio.

Se suele utilizar esta plantilla:

Como --- Quiero --- Para --- + Criterios de Aceptación

Las historias de usuario más importantes (priorizadas) se definen con el máximo detalle posible en un lenguaje entendible por el Peticionario (Usuario).

Es una buena práctica seguir ciertas normas de calidad como:

- Tratar de **descomponer un requisito complejo en requisitos individuales más simples** (se tiene que poder construir en un sprint)
- **Evitar solapamientos**: cada requisito describe un aspecto concreto
- **Evitar subjetividades**: juicios de valor
- No **utilizar conjunciones** (y, o...), negaciones ni pronombres, porque pueden llevar a contradicciones. En esos casos simplificar más el requisito.
- **Evitar términos técnicos en la medida de lo posible** (estamos en un documento de nivel de Usuario)

Además podemos tener requerimientos del tipo operacionales, contabilidad, informes, formularios, de convivencias, datos o cualquier otro tipo de requisito.

Cuando están refinadas por el equipo se completan con: el diseño de las pantallas, los casos de prueba, el diagrama de secuencia y la arquitectura que soporta.

Historias de Usuario de referencia

Es la **Guía o Unidad** de medida que permitirá al equipo **estimar**.

¿Cómo se crea?

1. Pensamos en una **historia de usuario** que no tenga nada que ver con el producto que se está construyendo, tiene que ser algo **entendible** por **cualquiera**: ejemplo login, pantalla de consulta, ... además se tiene que **poder construir** en el intervalo de tiempo de un **sprint**. **Individualmente** cada persona del equipo **propone** una historia y se **decide** por **consenso**.

2. Esta historia se **detalla** para que cumpla con el **Definition of Ready (DoR)**.

Todo el equipo **estima** el **esfuerzo** que implica **realizar** la **historia**, contando con todo el trabajo que se tiene que hacer para considerar esa historia como terminada (**DoD**), utilizando el **modelo de estimación relativa** decidido por el **Scrum Team**.



Incremento

La entrega de cada **Sprint**, todo lo que ha desarrollado el equipo en el Sprint y se ha finalizado.

Este incremento tiene que estar '**terminado**' esto significa que cumple las **condiciones** de ser **utilizado** y **se ajusta** con la **definición de 'Terminado' del Scrum Team: la Definition of Done (DoD)**.

Aporta un valor de negocio al producto que se está construyendo.



Product Backlog

Listado de funcionalidades y/o requisitos que tenemos para desarrollar nuestro producto o servicio.

- Contiene los **requisitos** necesarios para lograr el **objetivo**
- Es el **inventario** de **cosas por hacer**
- **Transparente y vivo**
- El **Product Owner** es el responsable de gestionarlo
- Es la **única fuente** de trabajo para el **Development Team**
- Contiene toda la información necesaria para la **previsión** de la **planificación** y presentación de **informes**
- Ordenado, basado en:
 - ROI (Return On Investment), valor, dependencias, riesgos...



Product Owner

Es la persona que decide **QUE** será construido. Define las características del **PRODUCTO** y **PRIORIZA** el **Product Backlog**.

¿Qué hace?

- **Identifica, define y prioriza** con los **Stakeholders** las **necesidades** del producto
- Garantiza que se cumplan las **áreas de soporte** y **control** para **lanzar el producto**
- **Representa al usuario** y es el **nexo** con el **equipo**
- Es la **voz exterior** del equipo
- **Comprende y define** el **objetivo** de negocio y maximiza el valor del producto.
- **Cuantifica el valor** que aporta al negocio la solución
- **Define y prioriza** las **Historias de Usuario**
- **Valida las entregas** de cada **Sprint**
- Asegura el **escalado** y **soporte** al equipo

¿Qué no hace?

- **NO estima** historias de usuario
- **NO** es el **gestor / director** de la producción (dice **QUÉ** no el **CÓMO**)

Cada equipo Scrum tendrá un Product Owner.

Requiere: amplias habilidades de toma de decisiones y negociación, ser una persona organizada y con buena planificación.

Scrum

Es un **marco** de trabajo para **desarrollo ágil** de **software** o **proyectos complejos**.

Estos son los pilares:

- **Principios del Manifiesto Agile**
- Chiste del cerdo y la gallina



INVOLUCRACION EQUIPOS AGILE SCRUM

- **Melé de Rugby o Scrum:** esta es la formación típica en las que todos los **jugadores** están **enlazados** para disputar por la pelota. **Simboliza** muy bien estos **equipos de desarrollo**. No se lleva la pelota el equipo que tienen el jugador más fuerte sino los que **trabajan mejor en equipo**. De esto va trabajar en Scrum, de trabajo en equipo y por esto esta formación precisamente le da el nombre Scrum.



Es muy sencillo de entender pero difícil de dominar y mantener.



Scrum Master

Es un **Facilitador** y un **rol clave** dentro de un equipo Scrum.

¿Qué hace?

- Es **líder Servicial** del **Equipo SCRUM**
- Se asegura de que el marco **Scrum** se ha **entendido** y se ha **adoptado**
- **Elimina obstáculos** y guía al equipo en las practicas del Scrum
- **Facilita** las **reuniones del Scrum**
- Hace que el **equipo** sea **multifuncional, auto-organizado** y **autogestionado**
- **Guiar** al **Product Owner** con sus **tareas**
- Representa al equipo ante el resto de la organización
- **Protege al Equipo** de Desarrollo de la **interferencia** externa
- **Guiar** y **retar** al equipo de desarrollo para su mejora en la **calidad** y **productividad**
- Que la gente del **equipo** esté **motivada**

¿Qué no hace?

- **NO** es el guru de la **tecnología**
- **NO** es el **gestor / director de proyectos**

Cada equipo Scrum tendrá un Scrum Master.

Requiere: amplias **habilidades sociales** y tiene que ser ordenado, estructurado y metódico.



Sprints o Iteraciones

Los **equipos Scrum** trabajan en **bloques de tiempo** que oscilan entre **1 y 4 semanas**, estos periodos de tiempo son siempre **fijos** en el proyecto y pueden variar por equipo. En Agile, se denominan Sprints, de modo que cada bloque de tiempo es un Sprint.

Durante estos periodos de tiempo se **crea** el **incremento** de **producto** utilizable y potencialmente desplegable en un entorno Productivo.

Cada **nuevo Sprint** comienza inmediatamente **después** de la finalización del Sprint **anterior**.

Stakeholder

Cualquier parte **involucrada** que tenga **interés** en el **producto** que se está construyendo.

Participan de manera puntual en la identificación y validación de requisitos apoyando al **Product Owner**, guiarán en la definición del producto de manera más estratégica.

Tendrán acceso a toda la información del producto que se está construyendo.

El Product Owner los invitará a la **ceremonia** de revisión de Sprint (**Sprint Review**).

Es imprescindible identificarlos y contar con ellos desde minuto cero.

Team agreements o Normas de equipo

Los **equipos Scrum** definen **sus propias normas** para facilitar el arranque y **buena marcha** del **equipo**.

Cada uno tenemos una experiencia – vivencia – cultura y en muchos equipos incluso idiomas diferentes. Definir normas ayudan a **reducir conflictos** y son una base para la **auto-organización, compromiso** y facilita que se alcancen los **objetivos**.

Las podríamos agrupar en la línea de: respeto, participación, **ceremonias**, transparencia, bloqueos, compromiso. Se **definen** antes de la primera interacción y se puede ir **evolucionando** durante el proyecto.



Velocidad

La suma de los puntos de **historia** terminados en un **Sprint** y aceptados por el **Product Owner**.

Esta medida se suele comparar entre Sprints para poder hacer proyecciones. Por ejemplo: ¿Qué alcance cubriremos en Febrero? ¿Parece que mi siguiente Release la podría liberar dentro X Sprint?

Los equipos Agile suelen ir incrementando la velocidad conforme van madurando.



Visión

Tiene que **transmitir la esencia del futuro producto de una manera concisa** y establecer un **objetivo compartido**.

Características:

- Compartida y unificada
- Amplia y atractiva
- Corta
- Clara y estable