



Scrum Master

Su objetivo es actuar como **facilitador, un eliminador de impedimentos**, un **líder que sirve tanto a la Organización, como al Equipo Scrum**. Asegura que el proceso Scrum se usa como es debido.



AYUDA A LA ORGANIZACIÓN A

- Liderando y guiando la **adopción de la Agilidad**
- Apoyando a **entender y adoptar Scrum** conforme su teoría, prácticas y reglas.
- **Motivando cambios** que incrementen la productividad del Equipo Scrum.
- Facilitando a las personas externas el **cómo interactuar con el Equipo Scrum**.



AYUDA AL PRODUCT OWNER A

- Encontrar técnicas para gestionar el **Product Backlog** maximizando el valor.
- Entender cómo **crear elementos** de Product Backlog **claros y concisos**.
- **Entender la planificación** del producto en un entorno empírico.
- Asegurar que el **valor aportado** es el **máximo posible**.
- **Entender y practicar el agilismo**.
- Facilitar las **dinámicas Scrum** que sea necesario.



AYUDA AL EQUIPO A

- Ser **auto-organizado**.
- Y **multifuncional**.
- Crear **productos de alto valor**.
- **Eliminar impedimentos**.
- Facilitar las **dinámicas Scrum**.



¿QUÉ NO ES?

- Un policía del Scrum Team.
- Un escriba o secretario.
- El que trae el café al equipo.
- El que preside la reunión.
- Un administrador.
- El líder del equipo.
- El jefe jerárquico del resto.



HABILIDADES NECESARIAS

- Habilidades estratégicas como **visión de la organización**, capacidad de **influencia** y habilidades de **negociación** y **gestión de conflictos**.
- Habilidades de **comunicación** como escucha activa, asertividad, dar y recibir **feedback**.
- Habilidades para la **construcción** y evolución de **equipos de trabajo**.



CEREMONIAS

EN LAS QUE PARTICIPA ACTIVAMENTE

- SPRINT PLANNING
- DAILY
- SPRINT REVIEW
- RETRO



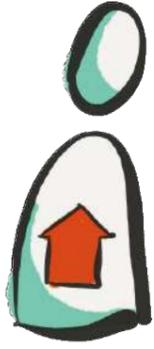
PREGUNTAS QUE DEBO HACERME

1. ¿Soy accesible para todos los miembros del equipo?
2. ¿He revisado qué bloqueos son recurrentes?
3. ¿El equipo se coordina y conversan entre ellos o están reportando a alguien? ¿Estoy consiguiendo que el equipo se auto-organice?
4. ¿Estoy sugiriendo utilizar todos los mecanismos para que el equipo evolucione a nivel técnico y a nivel de habilidades?
5. ¿Estoy en contacto y coordinándome con otros Scrum Masters y con la práctica ágil de otros equipos de la organización?
6. ¿Estoy contribuyendo con mi práctica Agile al cambio de la organización?
7. ¿Me mantengo actualizado en cuanto a nuevos modelos, herramientas Agile?



STOPPERS

- **Carencia de influencia** y liderazgo hacia otras posiciones jerárquicas.
- Bloqueos con otros equipos que no sean ágiles.
- Que los miembros del equipo **NO entiendan su rol** y sus relación con los demás.



Product Owner

Es el responsable de **mantener la visión del producto** que se va a construir **maximizando la cantidad de valor entregado** al finalizar cada iteración. Para mantener esa visión el dueño de producto tiene diferentes interlocutores a los que llamamos *stakeholders* o interesados en el proyecto.



¿CUÁLES SON SUS FUNCIONES?

- Obtener la información necesaria de los stakeholders para garantizar que el **producto desarrollado es el solicitado**.
- Transmite a los **stakeholders** las motivaciones de la metodología y **promueve su implicación**.
- Gestionar el backlog del producto, **definiendo los ítem del Backlog** a nivel de Negocio.
- Es responsable de transmitir de una forma efectiva la visión de producto a todos los implicados y **priorizar el trabajo**, de acuerdo al máximo valor aportado.
- Debe garantizar que el equipo tiene acceso al Backlog, aportando transparencia y claridad sobre cuál es la próxima funcionalidad a desarrollar.
- Debe velar porque todo el **equipo comprenda los ítems del Backlog**.
- Ha de **validar la funcionalidad** desarrollada de acuerdo a los criterios de aceptación que ha definido.
- Debe **conocer los impedimentos** que afecten al equipo y estar informado del desarrollo del Sprint.
- Es el responsable de **controlar** toda la **inversión** relacionada con el proyecto.



¿QUÉ NO ES?

- Un Analista Funcional.
- Un Jefe de Proyecto.
- El jefe del resto.
- Un dictador.



HABILIDADES NECESARIAS

- **Identificación de riesgos, priorización**, levantamiento de métricas.
- **Story Telling** (asociado a Historias de usuario o no).
- Habilidades estratégicas como **visión** de la organización, **conocimiento del producto**, de las tendencias de la **industria/mercado y tecnología**.
- Habilidades de **negociación y toma de decisiones**.
- Habilidades de **comunicación** para transmitir ideas y alternativas al cliente, al equipo y a todos los implicados.



CEREMONIAS

EN LAS QUE PARTICIPA ACTIVAMENTE

- SPRINT PLANNING
- SPRINT REVIEW
- RETRO



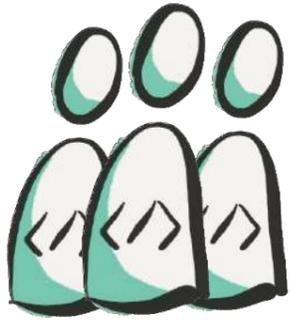
PREGUNTAS QUE DEBO HACERME

1. ¿Conozco el valor que aporta cada funcionalidad del producto para la organización?
2. ¿Están las necesidades definidas de una forma clara?
3. ¿Están bien definidos los criterios de aceptación?
4. ¿He revisado mi priorización teniendo en cuenta además el feedback del equipo?
5. ¿El backlog es lo suficientemente claro y transparente para el equipo?
6. ¿Estoy transmitiendo los objetivos para cada sprint?
7. ¿Conozco todos los impedimentos que pueden afectar al desarrollo del sprint?
8. ¿Una vez he cerrado una iteración dispongo de todo el feedback y las métricas para priorizar la siguiente?



STOPPERS

- Estar en una **posición** que me **condicione** la **priorización** del producto
- No conseguir que los stakeholders **diferencien entre importante y urgente**
- Que los stakeholders no se involucren de forma que pueda comprender claramente los **beneficios para la organización**
- No tener **capacidad de decisión** sobre varias peticiones que coincidan en importancia y en el tiempo



Development Team

El *development team* son los profesionales encargados de **construir el producto o servicio**, la pieza de valor. Los Equipos de Desarrollo son multifuncionales, contando como equipo con todas las habilidades necesarias para crear un Incremento de producto. Estos equipos deben ser **pequeños** (de entre 3 y 9 personas), **multidisciplinares** -que entre todos los miembros puedan dar servicio al proyecto- y **auto organizados** -que ellos mismos decidan la mejor manera de desarrollar su trabajo.



¿CUÁLES SON SUS FUNCIONES?

- Es **auto-organizado**. Nadie debe especificar cómo convertir el backlog en incrementos de funcionalidad.
- Es **multifuncional**. Cuenta con todas las habilidades necesarias para crear un incremento del producto, además, **no debe haber especialistas**.
- **El trabajo de cada uno de ellos aporta de forma directa a la entrega de valor.**
- Aunque puede haber especialistas, **el equipo es responsable como un ente completo**.
- Es deseable que se comparta el conocimiento para **evitar centralizarlo** en una única persona.
- Es el **responsable** de la **construcción** del producto.
- **Presenta el resultado** de la iteración al Product Owner.



¿QUÉ NO ES?

- Un grupo de esclavos o robots
- Calculadores de la fecha exacta de entrega
- Un grupo sin objetivos comunes elegido al azar



HABILIDADES NECESARIAS

- Habilidades para **trabajar de forma coordinada, autónoma y responsable.**
- Habilidades **especializadas** según el rol en el equipo.
- Habilidades de **comunicación y feedback.**



CEREMONIAS

EN LAS QUE PARTICIPA ACTIVAMENTE

- SPRINT PLANNING
- DAILY
- SPRINT REVIEW
- RETRO



PREGUNTAS QUE DEBO HACERME

1. ¿Cómo estamos trabajando como equipo? ¿Nos estamos coordinando bien?
2. ¿Tenemos acuerdos de equipo, son explícitos y los cumplimos?
3. Del 1 al 10, ¿cuán de autónomos somos para llevar el producto end-to-end?
4. ¿Cuán de participativos somos en cada una de las ceremonias? ¿Qué necesitamos para serlo más?



STOPPERS

- Que el equipo no cuente con recursos para la entrega de valor.
- Que haya un exceso de dependencias externas.
- Que el Product Owner no pueda resolver las dudas del equipo.

Errores comunes en las ceremonias

SPRINT REVIEW

- Querer usarlas para justificar un avance parcial.
- Se eternizan.
- No preparadas por el equipo.
- Que cada uno presente su parte.
- No pedir feedback.

SPRINT PLANNING

- Definición de las necesidades no READY.
- Product Owner no disponible.
- Criterio de DONE no completo.

STAND-UP DAILY

- Baja energía / perseguir gente.
- Que se use como una herramienta de supervisión o reporte.
- Hablar de tareas que no estén reflejadas en el panel.
- Entrar demasiado al detalle de los temas.
- Crear conversaciones paralelas.

SPRINT RETROSPECTIVE

- Monotonía.
- No guiadas cada vez por una persona.
- No fijar acciones concretas y responsables para mejorar.
- Que solo se traten temas técnicos o que solo se traten las relaciones entre el equipo.
- No revisar los aspectos identificados en la retro anterior.

NO cumplir los Time-box establecidos para cada una de las ceremonias.

Sprint Planning

Es el evento en el que se planifica el trabajo a realizar durante el siguiente sprint. Deben responderse las dos siguientes preguntas: ¿**qué puede entregarse** en el Incremento resultante del Sprint que comienza? ¿**Cómo se conseguirá** hacer el trabajo necesario para entregar el Incremento?



OBJETIVOS

- Fijar un **objetivo para el sprint** (Sprint Goal)
- **Seleccionar** y **estimar** (esfuerzo) los elementos sobre los que va a trabajar el equipo de desarrollo (Sprint Backlog).



¿CUANTO DURA (Time-box)?

- Máx. **8 h** en un sprint de 1 mes
- Máx. **4 h** en un sprint de 2 semanas



MOTIVACIONES

- Repasar y **comprender los ítems del Product Backlog** a desarrollar en esta iteración, y que pasarán a formar parte del **Sprint Backlog**.
- Asegurarse de que lo priorizado cumple con la definición de **Ready**.



¿QUIÉN DEBE PARTICIPAR?

- El **Product Owner**
- El **Equipo de Desarrollo**
- El **Scrum Master**

Daily Scrum o Stand-Up Daily

Es la ceremonia/reunión de **planificación y sincronización diaria** entre los miembros del Equipo de Desarrollo, en la que además subyace el plan de trabajo para las **próximas 24 horas**. Para ello se responden las tres preguntas desde el punto de vista del objetivo a lograr en el Sprint: **¿qué hice ayer? ¿Qué haré hoy? ¿Tengo o veo algún bloqueo o impedimento?**



OBJETIVOS

- **Comunicar y hacer explícitos los impedimentos** que existan o que vayan apareciendo.
- Hacer **partícipe a todo el equipo** de la situación actual del sprint y el avance planificado para las siguientes 24 horas. (Un panel físico refuerza este punto).



¿CUANTO DURA (Time-box)?

- Máx. **15 min** todos de pie.



MOTIVACIONES

- Mejorar la **sincronización** y el flujo de información entre los miembros del equipo.
- **Fomentar la comunicación** de manera activa.
- Adquirir un compromiso y **disciplina de equipo**.



¿QUIÉN DEBE PARTICIPAR?

- El **Equipo de Desarrollo**
- Recomendable que asistan (no participen) periódicamente el **Product Owner y el Scrum Master**

Sprint Review

El Equipo de Desarrollo explica **qué funcionalidades no se han terminado** y muestra al Product Owner las **funcionalidades que se han terminado y que componen el incremento**. El Product Owner aporta la **visión del mercado y del uso del producto**. El Equipo Scrum determina si se ha alcanzado el **objetivo del Sprint**.



OBJETIVOS

- **Inspeccionar** el Incremento y las mejoras construidas en esta iteración.
- Recibir **feedback** que permita al equipo orientarse hacia el producto final requerido por el usuario.



¿CUANTO DURA (Time-box)?

- Máx. **4h** en un sprint de 1 mes
- Máx. **2h** en un sprint de 2 semanas



MOTIVACIONES

- Comprobar que la funcionalidad solicitada se ha **desarrollado según los criterios de aceptación** propuestos.
- **Adaptar el Product Backlog** si fuese necesario.



¿QUIÉN DEBE PARTICIPAR?

- El **Product Owner**
- El **Equipo de Desarrollo**, y en caso necesario, los **stakeholders** pueden ser invitados a participar.
- El **Scrum Master** puede asistir.

Refinement (no oficial de Scrum)

El Refinamiento se consideraba previamente una **ceremonia** en la que se depuran los *Product Backlog Items* para que sean completamente claros y cuenten con toda la información necesaria para su desarrollo. Lo ideal es que este refinamiento **se produzca de forma continua** durante la iteración, sin recurrir a fijar una ceremonia específica. Originalmente se denominaba como *Grooming*, pero debido a las connotaciones negativas de este término se ha sustituido en recientes revisiones por *Refinement*.



OBJETIVOS

- Obtener un sprint backlog **priorizado**.
- Alcanzar el estado de **READY** para cada PBI.



¿CUANTO DURA (Time-box)?

- Máx. **10%** del tiempo del equipo en una iteración.



MOTIVACIONES

- Detectar los posibles **impedimentos o dependencias** que puedan existir, y madurar los ítems del backlog para que no se produzcan paradas o bloqueos durante el desarrollo.



¿QUIÉN DEBE PARTICIPAR?

- El **Product Owner**
- El **Equipo de Desarrollo**.

Sprint Retrospective

La *Sprint Retrospective* tiene lugar después de la *Sprint Review* y antes de la siguiente *Sprint Planning*. Es una oportunidad muy valiosa para el **Equipo Scrum** de **inspeccionarse a sí mismo** y crear un **plan de mejora** para el siguiente Sprint, es decir está **enfocada en el equipo y su proceso**, y no en el producto como otras ceremonias.



OBJETIVOS

- Identificar las **fortalezas** y las **áreas de mejora** tanto en los aspectos técnicos como en las relaciones del propio equipo.
- Obtener **acciones concretas** y responsables de su seguimiento.



¿CUANTO DURA (Time-box)?

- Máx. **3 h** en un sprint de 1 mes
- Máx. **1,5 h** en un sprint de 2 semanas



MOTIVACIONES

- Realizar un ciclo de **inspección y adaptación** sobre el propio equipo y el proceso.
- Fomentar la **mejora continua** como parte del proceso natural de un equipo ágil.



¿QUIÉN DEBE PARTICIPAR?

- El **Product Owner**
- El **Equipo de Desarrollo**.
- El **Scrum Master**.

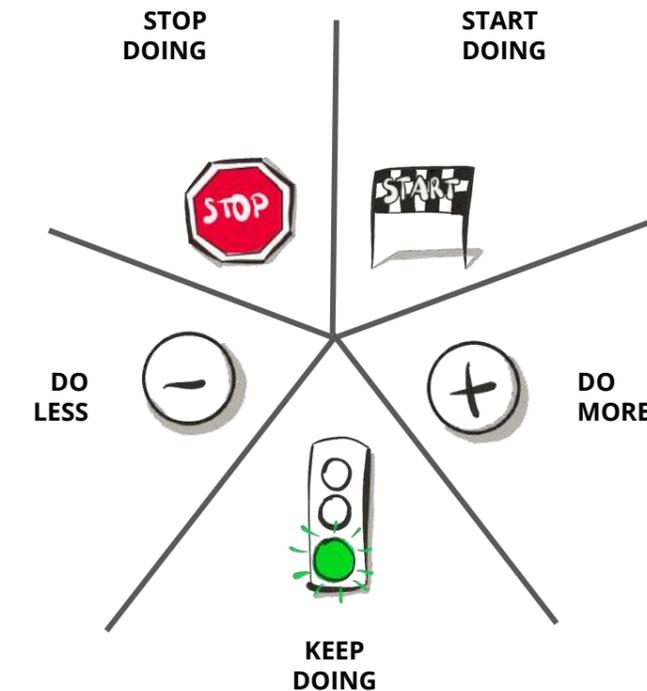
Hay múltiples modelos para llevar a cabo una retrospectiva, y es importante emplear diferentes para que no se convierta en algo rutinario y monótono para el equipo. Podemos usar fuentes de referencia para realizarlas como **Retromat** o **Fun Retrospective**.



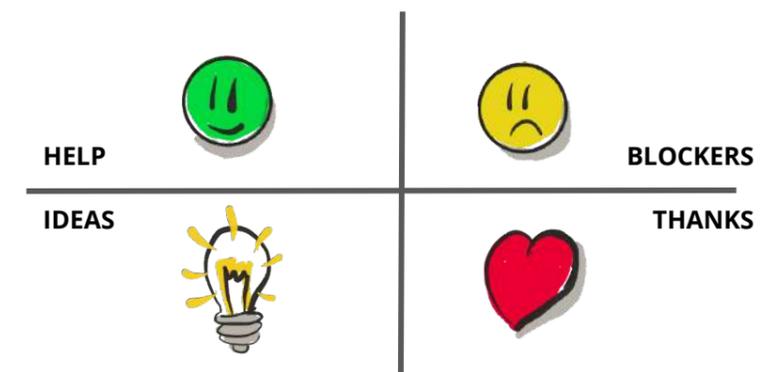
FASES

- 1. Preparar el escenario:** recordar el objetivo de la reunión, y el tiempo máximo que puede durar.
- 2. Recabar datos:** analizar el estado de ánimo del equipo, presentar hechos relevantes que han ocurrido en la iteración, identificar items terminados, no terminados, bloqueos, decisiones relevantes tomadas, etc.
- 3. Generar entendimiento:** analizar los hechos y responder a las preguntas ¿qué ha pasado? ¿Por qué ha pasado? ¿Cómo lo podemos mejorar?
- 4. Decidir qué hacer:** de todo lo identificado, elegir aquello sobre lo que queremos trabajar para incorporarlo en la siguiente iteración.
- 5. Cerrar la retrospectiva:** marcar responsables para seguir cada una de las acciones que hemos elegido. Motivar al equipo agradeciendo el trabajo, la participación y la transparencia.

MODELO ESTRELLA DE MAR



MODELO CLÁSICO





Chore

Los **Chores** son **items** que son **necesarios** pero que **no aportan** un **valor de negocio directo** u obvio al cliente.



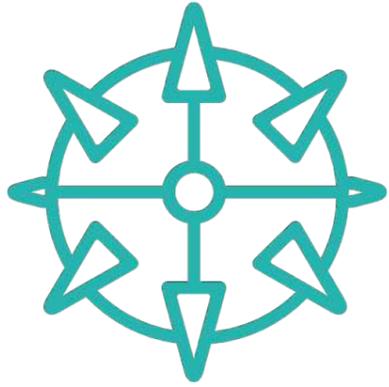
DEFINICIÓN

Un Chore es **algo que se necesita hacer antes o durante el desarrollo de un item** de funcionalidad que puede afectar a una o a varios product backlog items al mismo tiempo, **trabajos esenciales** para establecer entornos o de soporte para las funcionalidades, no aportan funcionalidad directamente, pero la construcción de esta **depende de ellos**. Algunos ejemplos podrían ser: “preparar un entorno de desarrollo” o investigar “por qué un determinado conjunto de pruebas tarda demasiado”.

Los Chores son un **coste para el negocio**. Utilizan recursos sin contribuir a la velocidad del equipo pero son necesarios. Los Chores tampoco son trabajos básicos, en general se necesita una capacidad y especialización similar a la necesaria para desarrollar cualquier otra funcionalidad.

Dicho esto, la importancia y necesidad es obvia pero:

- Los Chores pueden representar **deuda técnica** o **puntos de dependencia** con otros equipos. Pueden afectar a muchas funcionalidades y ser una fuente de una **importante carga de trabajo**.
- Los Chores **emergen a lo largo del tiempo** según la necesidad y ese es el momento de acometerlos **no son actividades que se puedan o deban anticipar**.



Spike

Spike, es un término que respondería a la **pregunta**: ¿Qué es lo **más simple** que podemos **programar** para **asegurarnos** que vamos por el **buen camino**?



DEFINICIÓN

Es decir, un Spike sería el programa más sencillo que nos permite explorar diferentes opciones o soluciones. Es una tarea orientada a contestar una pregunta o conseguir una información más que a producir un producto entregable.

A veces un *product backlog item* no se puede estimar correctamente hasta que el equipo de desarrollo haga algún trabajo de investigación para resolver o contestar una pregunta técnica o un problema de diseño. La solución es crear un "spike", un trabajo o actividad, cuyo objetivo es conseguir esa respuesta o solución.



AVANZADO

En SAFe, por ejemplo, los Spikes se definen como un *item* facilitador de un tipo de exploración. Sus orígenes están en XP y representan actividades del tipo de diseño, investigación, exploración y prototipado. El objetivo no es otro que ganar el conocimiento necesario para reducir el riesgo de un planteamiento técnico, entender mejor un requerimiento o mejorar la fiabilidad de la estimación de un funcionalidad.

Como cualquier otro *item*, los Spikes deben ser estimados y demostrados al final de su Sprint.



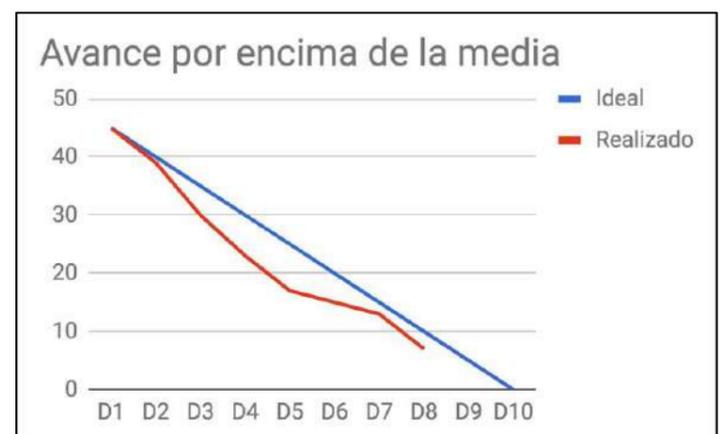
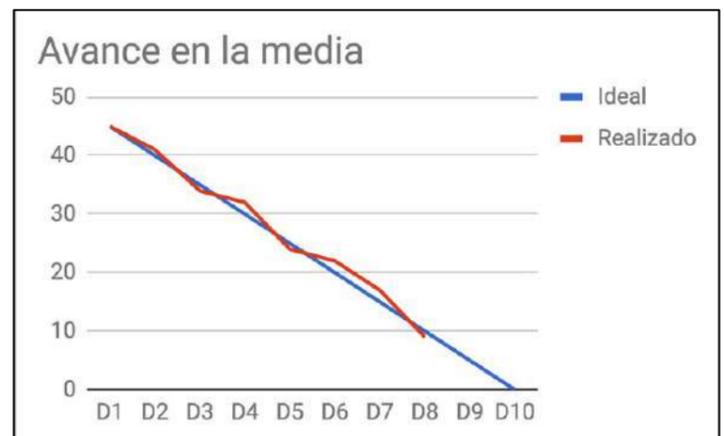
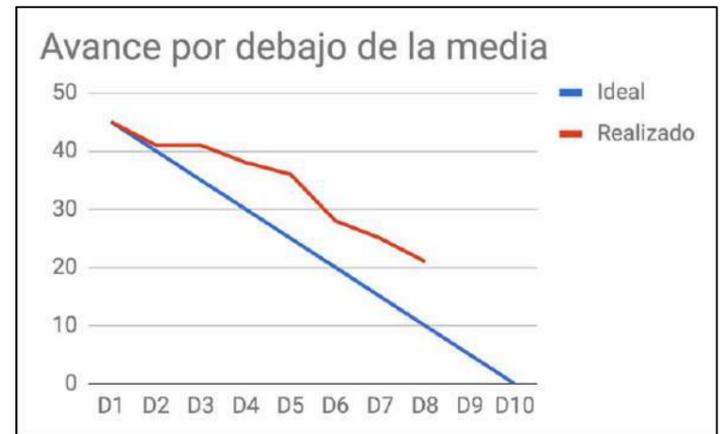
Burn down chart

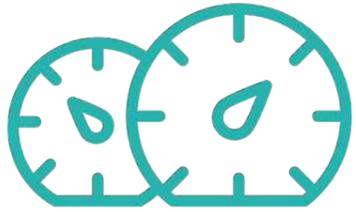
El objetivo de este gráfico es mostrar el **trabajo** (esfuerzo) **que queda por realizar** en un periodo de tiempo concreto (iteración). En los equipos de desarrollo ágiles, el gráfico Burn Down se suele **aplicar por sprint** y ayuda al equipo a la **visualización** del **esfuerzo realizado** y el que queda por realizar hasta la finalización del sprint.



UTILIZACIÓN

- En el eje **X se representa el tiempo**, bien en semanas, en días, etc, dependiendo de la duración de la iteración.
- En el eje **Y se representa el esfuerzo** a realizar, la mejor unidad para hacerlo son los puntos de esfuerzo.
- Si el gráfico representa un sprint, entonces la Y debería ser los **puntos totales de esfuerzo** para este Sprint, y la X la **duración del Sprint**.
- Se han tomado como referencia para el ejemplo un Sprint de 2 semanas (10 días) y una estimación de 45 puntos de esfuerzo a realizar por el equipo.
- Uniendo ambos puntos, obtendremos la pendiente que representa el **esfuerzo medio a realizar por día**. Es decir, el avance teórico del consumo de *items* y tareas a lo largo del sprint (ideal).
- El gráfico se usa colocando una marca **(un punto) en la coordenada** que corresponde al **día** dentro **del sprint** y a los **puntos** de esfuerzo **que quedan** al restar, los realizados hasta el momento, menos el total de puntos del sprint. Tras esta acción, se debe unir esta coordenada con la coordenada representada anteriormente con una línea.
- Si estamos **por encima de la pendiente**, el sprint va avanzando **por debajo de la media**, si estamos **por debajo de la pendiente**, el avance es **por encima de la media**.





Velocity

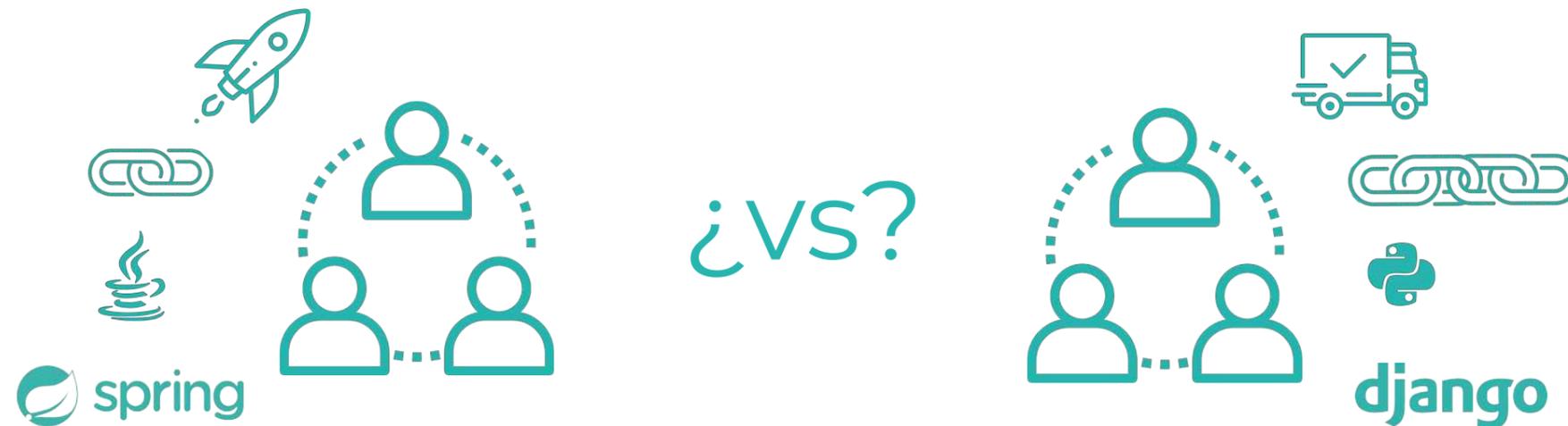
La velocidad **no es un objetivo, es una consecuencia**, se puede denominar como la cantidad de puntos de esfuerzo que el equipo ha sido capaz de finalizar en un sprint.



¿PARA QUÉ NOS SIRVE?

- La Velocidad **no es un valor que permita comparar la productividad entre equipos**, ya que depende de numerosos factores, como las **tecnologías** que se emplee (un equipo puede trabajar con Java y otro con Python), la complejidad del **producto** en el que trabajen (uno puede ser un producto de terceros que adaptamos, otro un desarrollo desde cero in-house), la complejidad del **negocio** (no es lo mismo calcular una órbita de reentrada, que la franja de entrega de un paquete), las **dependencias** (hay desarrollos fuertemente dependientes de terceros que pueden impactar)...

Básicamente estaríamos comparando “peras” con “manzanas”.



- Dentro de un mismo equipo sí que nos puede ayudar para **analizar la tendencia** que tiene y aunque sea de forma estimativa, comparar la velocidad, con puntos de esfuerzo del backlog para hacer una proyección (con todas las consideraciones que una proyección puede tener) en un mismo equipo.
- Debe ser uno de nuestros objetivos **modificar el criterio de medición**, de productividad a **valor entregado**.