

# 3. Scrum Master:

## Líder al servicio del equipo Scrum

En este capítulo, nos adentraremos en la función vital que desempeña el Scrum Master en la implementación de Scrum. El Scrum Master no solo asegura una adopción precisa de Scrum según lo delineado en la Guía de Scrum, sino que también facilita la comprensión y el respeto de su teoría y prácticas a todos los niveles organizacionales. A lo largo de este segmento, desplegaremos un análisis detallado de las diversas actividades, habilidades y responsabilidades que caracterizan esta función, ampliando lo ya explorado en el modelo core de Scrum. Puntos Clave de este Capítulo

- Profundizar en la responsabilidad crítica del Scrum Master en la adopción y mejora de Scrum.
- Analizar las actividades clave que el Scrum Master facilita dentro del equipo Scrum.
- Estudiar las habilidades esenciales que un Scrum Master debe cultivar para liderar con eficacia.
- Examinar las responsabilidades a nivel organizacional que incumben al Scrum Master.

- [3.1 - Habilidades del Scrum Master](#)
- [3.2 - Los valores del equipo](#)
- [3.3 - Los Pilares de Scrum](#)
- [3.4 - El Scrum Master y los eventos](#)
- [3.5 - Técnicas útiles para el Scrum Master](#)
- [3.6 - Emisores de información útiles para el Scrum Master](#)

# 3.1 - Habilidades del Scrum Master

Un Scrum Master debería tener un conocimiento profundo del marco Scrum por varias razones:

- Para poder aplicarlo de manera coherente y consistente
- Para poder resolver problemas y superar obstáculos
- Para poder adaptar Scrum a las necesidades del equipo
- Para poder enseñar y orientar a otros

Como scrum master (SM) es necesario brindar soporte y soluciones al Equipo Scrum y a la organización en general; para esto debe contar con aptitudes y actitudes que contribuyan al liderazgo, a continuación se muestran algunas habilidades que tiene un SM:

- Habilidades de liderazgo
- Habilidades de facilitación
- Habilidades de comunicación
- Habilidades para la resolución de problemas
- Habilidades de enseñanza

## Habilidades de liderazgo

Las habilidades de liderazgo en un Scrum Master se refieren a la capacidad de liderar y guiar al equipo para que alcance sus metas y objetivos. Algunas de las habilidades de liderazgo esenciales para un Scrum Master incluyen:

- Fomentar la colaboración y el trabajo en equipo
- Ayudar a los miembros del equipo a superar obstáculos
- Ser una fuente de apoyo y orientación
- Establecer objetivos y expectativas claras
- Inspirar y motivar al equipo

**Ejemplo:** Durante un proyecto, el equipo se siente inseguro respecto a la nueva tecnología que están utilizando. El Scrum Master, reconociendo la necesidad de liderazgo, organiza una serie de sesiones de capacitación con expertos para familiarizar al equipo con la tecnología, estableciendo así objetivos claros y

proporcionando apoyo y orientación.

#### Tip

Recuerda que un buen líder no sólo dirige, sino también inspira y motiva. Ser un líder en Scrum significa servir al equipo, eliminando obstáculos y ayudando a cada miembro a dar lo mejor de sí.

# Habilidades de facilitación

El SM logra que todas las acciones, procesos y tareas sean fáciles de entender para el equipo, además de asegurar que se cuente con el entorno y herramientas necesarias para trabajar. Es necesario que actúe de manera:

- Colaborativa: para guiar al equipo en sus funciones
- Neutral: para entender las necesidades de todas las partes sin tener preferencia alguna
- Responsable: garantizando la disciplina y la efectividad del equipo

**Ejemplo:** Durante la retrospectiva, algunos miembros del equipo discuten sobre qué metodología de trabajo es la más eficiente. El Scrum Master, como facilitador, interviene para asegurar que todos tengan una oportunidad de expresarse y que la discusión se mantenga centrada en la mejora continua, evitando desviarse hacia debates improductivos.

## El Scrum Master facilitando los eventos

El SM como facilitador debe tener en cuenta los siguientes aspectos al momento de moderar un evento:

- Gestionar un evento: es necesario que previamente se cuente con la información necesaria, tal como: objetivos, duración, participantes y demás.
- Asegurar las intervenciones necesarias: para que todos puedan participar y tomar el liderazgo.
- Controlar el ritmo: para que no se generen bucles o discusiones innecesarias, y se cumpla el objetivo del evento.

# Habilidades de comunicación

Las habilidades de comunicación en un Scrum Master se refieren a la capacidad de comunicarse de manera efectiva con todas las personas con las que trabaja. Algunas de las habilidades de comunicación esenciales para un Scrum Master incluyen:

- Transmitir de manera clara y concisa los objetivos y expectativas del producto
- Brindar feedback constructivo
- Escuchar activamente
- Comunicar cambios y decisiones de manera clara
- Resolver conflictos

**Ejemplo:** Uno de los desarrolladores, Juan, ha estado teniendo dificultades con una tarea específica. En vez de señalarlo abiertamente y ponerlo en una situación incómoda, el Scrum Master se acerca de manera privada, ofrece feedback constructivo sobre lo que ha observado y escucha activamente las preocupaciones de Juan, buscando soluciones conjuntas.

#### Tip

Recuerda que la comunicación no es solo hablar, sino también escuchar. Establece canales de comunicación abiertos, promueve la retroalimentación constante y verifica siempre que los mensajes hayan sido entendidos correctamente por todos los miembros del equipo.

# Habilidades para la resolución de problemas

Las habilidades de resolución de problemas en un Scrum Master se refieren a la capacidad de identificar y resolver problemas que puedan surgir durante el trabajo con el equipo, esto considerando que No puedes impedir que aparezcan conflictos o problemas, pero puedes gestionarlos para minimizar el impacto al equipo, para esto debes desarrollar:

- Inteligencia emocional: individual y colectiva
- Entender las necesidades: como persona, más que como empleado y así abordar los conflictos oportunamente.
- Analizar la disposición de las personas y enfocarte en el problema.
- Hacer acuerdos para avanzar.
- Mantener una comunicación asertiva para superar el conflicto con madurez.

#### Tip

En lugar de brindar soluciones inmediatas, guía al equipo a través del proceso de descubrimiento de soluciones. Esto empodera al equipo y les ayuda a desarrollar habilidades de resolución de problemas por sí mismos.

# El Scrum Master como Eliminador de impedimentos

**Un impedimento es cualquier cosa que obstruye el flujo de trabajo generando retrasos.**

Es importante que el SM pueda remover los impedimentos del equipo para garantizar el flujo y a su vez el éxito del Sprint.

**Ejemplo:** El equipo enfrenta un bloqueo técnico que ha detenido el progreso. El Scrum Master organiza una sesión de tormenta de ideas (brainstorming) donde todos pueden compartir posibles soluciones. Se anima al equipo a pensar de manera creativa, y juntos, logran encontrar un enfoque innovador que resuelve el problema.

## Desafíos para el Scrum Master al eliminar impedimentos:

Desafío	¿Cómo lo puedes abordar?
<b>El equipo Scrum no reporta el impedimento.</b> Por lo tanto el SM desconoce la situación real del ambiente y no puede solucionar oportunamente.	<ul style="list-style-type: none"><li>- Preguntar para identificar falencias del equipo.</li><li>- Trabajar en conjunto para solucionar</li><li>- Fortalecer las habilidades de comunicación</li></ul>
<b>El equipo se estanca con un impedimento fácil de resolver.</b> El SM puede actuar oportunamente, hablando con el equipo sin afectar el flujo del proceso, brindando soluciones en tiempo real.	<ul style="list-style-type: none"><li>- Encontrar soluciones aplicables</li><li>- Hacer una pausa para despejar la mente</li><li>- Analizar el impedimento de manera creativa con el equipo</li></ul>
<b>Falta capacidad y autoorganización en el equipo.</b> El SM debe conocer la cultura y el contexto organizacional, procesos, proyectos y otros equipos de trabajo para apoyarse y buscar una solución al impedimento.	<ul style="list-style-type: none"><li>- Buscar personal idóneo para completar la capacidad del equipo</li><li>- Apoyarse en otros equipos para encontrar la solución al impedimento</li></ul>

# Habilidades de enseñanza

El SM debe animar al equipo para aprender nuevos conocimientos, esto permite que el equipo sea multidisciplinario y no se estanque fácilmente. Es necesario que el Scrum Master:

- Tenga seguridad para guiar al equipo
- Desarrollar comunicación abierta y transparente
- Establecer reglas claras con el equipo
- Involucrarse en todas las actividades (no como supervisor, sino como apoyo)
- Recordar que trabajas con personas no robots

Aprovecha los momentos informales para enseñar y compartir conocimientos. En ocasiones, una conversación casual puede ser una oportunidad dorada para impartir una lección valiosa. Recuerda, el aprendizaje no siempre tiene que ser estructurado o formal; a veces, los momentos espontáneos dejan las enseñanzas más duraderas.

## El Scrum Master como coach y mentor

Como SM sueles entrenar a todo el equipo según sus habilidades y su estilo de trabajo.

**Como coach:** Su enfoque está en enseñar a los demás a mejorar. Un coach no necesariamente sabe más, busca ayudarte a sacar lo mejor de ti, para que encuentres mejores soluciones con tus propias habilidades.

**Como mentor:** Su enfoque está en ayudar y guiar a quienes tienen menos experiencia. Un mentor sí es alguien que sabe más que los demás sobre un determinado dominio y usa su conocimiento para guiar a los demás.

**Ejemplo:** Sara, una nueva integrante del equipo, no está familiarizada con las prácticas de Scrum. El Scrum Master organiza sesiones individuales y grupales de capacitación para enseñarle los conceptos clave. A través de ejercicios prácticos y discusiones, el Scrum Master asegura que Sara comprenda no sólo los procesos sino también los valores y principios detrás de Scrum.

## Habilidades complementarias

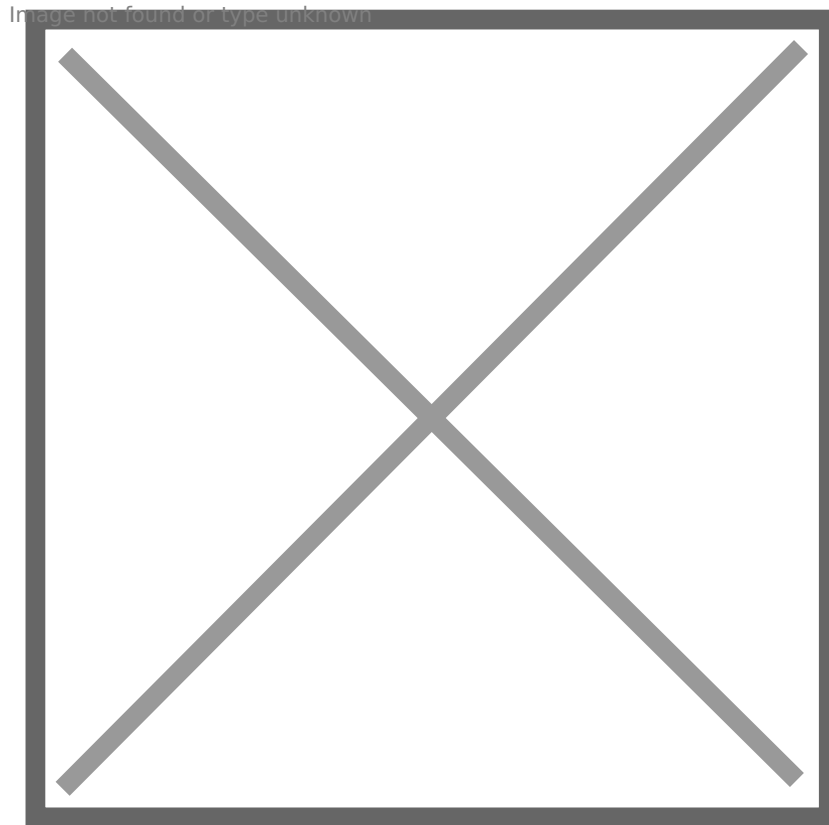
- **Habilidades de escucha:** Es importante resaltar la capacidad de escuchar, no sólo oír. Escuchar activamente implica comprender las preocupaciones, inquietudes y necesidades del equipo.
- **Capacidad de autoaprendizaje y adaptabilidad:** El mundo de la agilidad está en constante evolución. Un Scrum Master debe estar dispuesto a aprender y adaptarse a los cambios.
- **Integridad:** Es vital que un Scrum Master sea honesto y ético en todas sus acciones. Esto construye confianza y establece un ejemplo para el resto del equipo.
- **Empatía:** Poder ponerse en el lugar de los demás es fundamental para comprender las necesidades y preocupaciones del equipo.
- **Visión estratégica:** Aunque el SM no es el responsable del producto, es útil que entienda la visión a largo plazo para guiar mejor al equipo.
- **Gestión del tiempo:** Dado que los Sprints son ciclos de tiempo limitado, la habilidad de gestionar el tiempo eficientemente es crucial.

#### **Tip**

Como Scrum Master, predica con el ejemplo en cuanto a la gestión del tiempo. Utiliza herramientas de planificación, establece prioridades claras, aprende a decir "no" cuando sea necesario para asegurarte de que las tareas cruciales se cumplan dentro de los plazos. ¡Recuerda, trabajar inteligentemente es tan importante como trabajar arduamente!

## 3.2 - Los valores del equipo

Es importante que entre los miembros del equipo se genere y mantenga una filosofía basada en valores que fomenten la confianza, la comunicación y la entrega de resultados. A continuación, se relacionan los valores que se deben promover en un equipo Scrum:



- Compromiso
- Enfoque
- Apertura
- Respeto
- Coraje

**Nota:** El Scrum Master promueve los valores en el equipo.

En un equipo Scrum, los valores juegan un papel central en la construcción de una base sólida para la colaboración y la entrega efectiva.

- El **Compromiso** se refleja en la responsabilidad compartida, donde cada miembro se dedica plenamente a alcanzar los objetivos del equipo.





**Ejemplo:** Tras la planificación de un Sprint, el equipo se da cuenta de que una de las historias de usuario comprometidas es más compleja de lo que inicialmente pensaban. En lugar de relegarla o dividirla, el equipo decide organizarse de forma diferente, redistribuyendo algunas tareas para asegurarse de que todos los elementos comprometidos se entreguen a tiempo.

- El **Enfoque** significa mantener una visión clara de las metas, priorizando tareas y asegurándose de que el esfuerzo se invierta en lo que realmente importa.

**Ejemplo:** A mitad de un sprint, la dirección de la empresa sugiere una nueva funcionalidad que considera urgente. El equipo decide mantenerse enfocado en las tareas comprometidas para ese sprint y propone discutir la nueva solicitud en la próxima reunión de planificación, asegurando que la dirección actual del sprint no se vea comprometida.

- La **Apertura** implica una comunicación transparente y honesta, creando un ambiente donde las ideas y preocupaciones puedan ser compartidas libremente, y donde el feedback sea recibido como una oportunidad de mejora.

**Ejemplo:** Al final de un sprint, una característica desarrollada no cumple con los estándares de calidad esperados. En lugar de intentar "maquillar" los resultados o posponer la revisión, el equipo comunica abiertamente la situación en la Revisión del Sprint, expone las causas del problema y propone una estrategia de corrección.

- El **Respeto** se manifiesta en el reconocimiento de las habilidades y opiniones de cada miembro, fomentando una cultura de empatía y consideración mutua, asegurando que cada voz se escuche y valorando la diversidad en el equipo.

**Ejemplo:** Un miembro nuevo se une al equipo con una forma diferente de trabajar y con herramientas con las que el equipo no está familiarizado. En lugar de resistirse o desestimar su enfoque, el equipo respeta y se toma el tiempo para entender las ventajas que podría traer este nuevo método, valorando la diversidad de experiencia que el nuevo miembro aporta.

- Finalmente, el **Coraje** es esencial para enfrentar desafíos y cambios, permitiendo al equipo tomar riesgos calculados y aprender de los errores, y abordar situaciones difíciles con determinación.

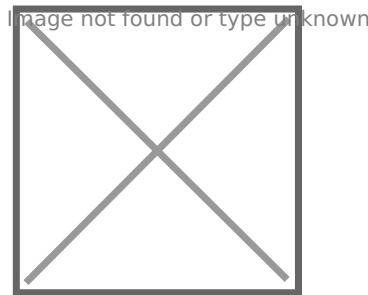
**Ejemplo:** Después de varios días trabajando en una característica, el equipo se da cuenta de que la solución que estaban desarrollando no es óptima. Aunque revertir y cambiar de rumbo podría parecer desalentador, el equipo decide correr el riesgo, ya que creen que es lo mejor para el producto final. Hacen frente al desafío y trabajan juntos para encontrar una solución más adecuada.

Estos valores no sólo fortalecen la relación del equipo sino que también potencian su rendimiento, creando una cultura de excelencia. Cuando estos valores son asimilados por el equipo de Scrum (reforzados continuamente) y con las personas con las que trabajan, la teoría de Scrum que enfatiza en la transparencia, inspección y adaptación se materializa construyendo confianza.

## 3.3 - Los Pilares de Scrum

Si bien, los 12 principios del manifiesto ágil describen una orientación sobre los principios que se siguen en Scrum, hay 5 elementos que son clave para el buen funcionamiento de este marco de trabajo.

Estos pilares junto a los 12 principios del manifiesto ágil se deberían conocer y aplicar por los miembros del Equipo Scrum. El Scrum Master suele ser el responsable de que estos pilares hagan parte de la cultura del equipo.



### 1. Auto-gestión en el Equipo Scrum

Para Scrum, es altamente recomendable que el equipo tenga la capacidad de auto-gestionarse, esto significa que son los miembros del equipo quienes eligen la mejor opción de llevar a cabo su trabajo sin ser dirigidos por personas externas al equipo.

La auto-gestión es una característica definitoria de los equipos Scrum, centrada en la confianza, autonomía y responsabilidad de sus miembros. Aquí es donde los equipos no sólo toman decisiones sobre cómo abordar su trabajo, sino que también asumen la responsabilidad de esas decisiones.

A continuación, se listan algunos elementos clave que hace posible la Auto-Gestión del Equipo Scrum:

- Los desarrolladores deberían autoasignarse el trabajo a realizar en los diferentes Sprints, nadie, ni siquiera el Product Owner debe imponer el trabajo al Equipo.
- El equipo decide la mejor distribución del trabajo, garantizando la equidad y el trabajo en equipo.
- El Equipo debe conocer muy bien sus límites de decisión para así poder tener mayor autonomía (delegación).

- El Equipo debe tener espacios que le permitan realizar jornadas de investigación y capacitación.
- Se debe trabajar continuamente en la motivación del equipo.
- No se deben cambiar los miembros del equipo (a menos que sea inevitable).

#### Tip para el Scrum Master

Las mejores prácticas y herramientas evolucionan con el tiempo. Dedica tiempo para la formación continua y asegúrate de que el equipo esté al día con las últimas tendencias y técnicas en Scrum.

## Propiedad y responsabilidad colectiva del producto

Para garantizar una colaboración constante y evitar con el tiempo la aparición de una cultura de la culpa, es importante fomentar una propiedad colectiva del producto, esto significa que todo el Equipo Scrum es dueño del producto, y por tanto cualquiera de sus integrantes podría contribuir al desarrollo de cualquier parte del producto aun cuando no haya sido quien lo desarrolló inicialmente.

Así mismo no deberían existir reconocimientos individuales a los miembros del equipo por sus contribuciones al producto.

#### Tip para el equipo

**Evita la cultura de la culpa:** En lugar de buscar a quién culpar, enfócate en cómo resolver el problema.

## Colaboración de equipo

La colaboración se da gracias a la constante comunicación que existe en los Equipos Scrum, tanto entre sus miembros como con las Partes interesadas del Proyecto, este concepto es parte integral del Manifiesto Ágil "**La forma más eficiente y efectiva de transmitir información hacia y dentro del Equipo de Desarrollo es la conversación cara a cara**".

El Scrum Master es el rol responsable de garantizar una sana comunicación entre todas las partes interesadas del proyecto, en especial de su Equipo de Desarrollo.

#### Tip para el equipo

**Fomentar una cultura de feedback** constante puede simplificar muchos aspectos del trabajo. Las correcciones tempranas y la orientación adecuada evitan la complejidad de corregir errores más adelante.

Se debe considerar que, según la naturaleza del proyecto, las necesidades de la organización e incluso factores externos, se determinan la ubicación de los miembros del Equipo:

- **Equipos Centralizados:**
  - Los miembros del Equipo se encuentran en la misma ubicación, lo que les permite comunicarse con gran facilidad.

- La resolución de problemas es prácticamente inmediata, ya que al estar ubicados en el mismo lugar es fácil realizar sesiones de diálogo.
- **Equipos Distribuidos:** Es aquel en el que sus miembros no se encuentran en una misma ubicación, por lo general está disperso debido a factores como:
  - La subcontratación (outsourcing – freelance).
  - Oficinas ubicadas en diferentes ubicaciones físicas.
  - Trabajo desde casa.

Para garantizar la comunicación permanente en este tipo de equipos se hacen necesarias las siguientes herramientas:

- Groupware.
- Software Videollamadas o chat.
- Software de gestión de proyectos ágiles.
- Herramientas de software que simulan la funcionalidad de Scrum boards.

## Equipos multifuncionales

Los equipos multifuncionales (también llamadas células ágiles) tienen todas las competencias y habilidades necesarias para llevar a cabo el trabajo sin depender de otras personas que no formen parte del equipo.

Contrario a lo que piensa un equipo multifuncional, no se trata de que todos sus integrantes hagan de todo, se trata de que los integrantes adquieran conocimiento en distintas disciplinas (aplicables a los proyectos de la organización) y así puedan contribuir eficazmente con la colaboración.

La realidad es que incluso cuando un equipo sea experto técnico, siempre necesitarán capacitación adicional, así es que el Product Owner deberá decidir si aprobará el dinero y el tiempo para capacitarse o por el contrario serán los miembros del equipo quienes se encargarán del tema.

## Colaboración con el cliente

En los proyectos tradicionales, los clientes por lo general se mantenían a distancia y solo se involucraban al principio y al final del proyecto. En Scrum es altamente recomendable que el cliente participe de las revisiones del producto y brinde retroalimentación en todos los puntos de "inspección y adaptación". Esto minimiza el riesgo y le brinda más opciones al cliente y a las partes interesadas.

Por ejemplo, en otros Marcos Ágiles como XP, es obligatorio que el Cliente forme parte del equipo.

El cliente (o sus representantes) deberían trabajar junto al Product Owner para definir las historias de usuario y detallar dichas historias antes o durante las reuniones de planificación.

El cliente y las partes interesadas por lo general participan en la Reunión de Revisión de los Sprint y, dependiendo de la relación entre el cliente y el Product Owner, el Cliente incluso podría participar de algunas reuniones de Retrospectiva de los Sprint.

## Gestionar el conocimiento

Dado que los equipos autogestionados se responsabilizan de su aprendizaje y crecimiento, la gestión del conocimiento se convierte en un aspecto crucial. Esto puede tomar la forma de compartir habilidades, aprender nuevas tecnologías o técnicas, o incluso la formación entre pares.

Gestionar el conocimiento permitirá identificar, recopilar, organizar, transferir y retener el conocimiento necesario para dar soporte a todo el personal en sus actividades laborales, para la toma de decisiones bien fundadas y para aumentar la productividad.

#### Tip para el Developer

Si bien es esencial documentar, es fundamental preguntarte: "**¿Esta documentación agrega valor?**". Evita la creación de documentos extensos y en su lugar, apunta a documentaciones claras y concisas que sean realmente útiles.

## Motivación del equipo

Los equipos Scrum se caracterizan por mantener un enfoque hacia la entrega frecuente de resultados; y aunque los miembros del equipo son conscientes de la responsabilidad que esto implica, existe un factor de fondo que facilita el impulso y el esfuerzo para cumplir con los objetivos: la motivación.

La motivación hace referencia a que los miembros del equipo mantengan determinada conducta y estado de ánimo que propicien las interacciones sanas y el alto rendimiento en el proyecto.

- **Motivación Intrínseca:** este tipo de motivación es propio de cada persona, es decir que por su propia voluntad e inspiración es capaz de mantener una conducta específica y el impulso necesario para cumplir con una meta que brinda satisfacción interna y realización personal. (Ref: CHAMPFROGS – Moving Motivators)
- **Motivación Extrínseca:** este tipo de motivación hace referencia a mantener una conducta específica para responder a un impulso externo, es decir que en este caso la voluntad e inspiración de la persona se ven influenciadas por una recompensa externa (que puede ser algo físico, monetario o psicológico).

Según cómo se maneje la motivación en el equipo Scrum, eventualmente la motivación extrínseca tiende a convertirse en motivación intrínseca, pues los miembros del equipo van adaptando su conducta y mejorando su rendimiento para cumplir los objetivos sin necesidad de que todo el tiempo estén recibiendo recompensas o algo a cambio.

---

## 2. Simplicidad

Scrum no sería considerada una metodología ágil de no ser por su simplicidad, es por ello que se intenta al máximo reducir la burocracia en sus prácticas, se trabaja con los artefactos que son esenciales para el equipo y se sigue un flujo de prácticas simple, sin descuidar todos los elementos

necesarios para la correcta gestión del producto, eliminando cualquier trabajo superfluo o que no aporte valor.

#### Tip para el Product Owner

**Simplicidad en la Arquitectura del Producto:** Una arquitectura bien diseñada y simple facilita la adición de nuevas características y la corrección de errores. La sobrecarga arquitectónica puede ralentizar considerablemente el desarrollo.

A través de la simplicidad, se puede acelerar la entrega, mejorar la comunicación y reducir los riesgos asociados con la complejidad.

Algunos elementos que hacen posible la Simplicidad en Scrum son:

## Maximizar el trabajo no realizado

La simplicidad no solo trata de hacer las cosas de manera simple, sino también de identificar y eliminar cualquier tarea que no sea esencial para alcanzar los objetivos del producto. Es una forma de ser eficiente y focalizado, dedicando recursos solo a lo que realmente importa.

## Clara definición de "Hecho" (terminado)

Una definición clara y concisa de lo que significa "hecho" para un elemento del producto ayuda a evitar el trabajo innecesario. Si el equipo entiende perfectamente cuándo un ítem está completo, se evita el esfuerzo extra y las revisiones innecesarias.

## Priorización

La simplicidad se logra también a través de una adecuada priorización de las tareas. Al centrarse en lo que realmente aporta valor al cliente o usuario final, el equipo puede evitar la complejidad y concentrarse en lo esencial.

## Comunicación efectiva

Las soluciones simples a menudo emergen de una comunicación clara y efectiva. Al fomentar un diálogo abierto y honesto entre los miembros del equipo y con los interesados, se pueden identificar y eliminar obstáculos, malentendidos y redundancias.

#### Tip para el equipo

Cuanto más dependencias existan entre tareas o equipos, más complejo se vuelve el trabajo. Al minimizar estas dependencias, no solo se simplifica el proceso, sino que se mejora la velocidad y fluidez del trabajo.

## Refactorización continua

La simplicidad también está en el código. Refactorizar regularmente el código para mantenerlo limpio y manejable es esencial. Un código más simple es más fácil de entender, modificar y mantener.

## Limitar el trabajo en progreso (WIP)

Al limitar la cantidad de trabajo en progreso, los equipos pueden centrarse en finalizar tareas antes de comenzar nuevas. Esto reduce la complejidad y facilita una entrega más rápida y eficiente.

## Usar herramientas de software

Las herramientas de software dedicadas a la gestión de proyectos ágiles se han consolidado como elementos cruciales para simplificar procesos y tareas. Estas herramientas ofrecen diversas ventajas, entre las cuales destacan:

- **Centralización de Datos:** Agrupan toda la información de los proyectos en un solo lugar, lo que facilita el acceso, control y análisis de la información.
- **Automatización de Tareas:** Las herramientas permiten la automatización de diversas actividades, tales como:
  - Realizar estimaciones basadas en datos históricos.
  - Calcular métricas como la velocidad del equipo.
  - Generar gráficos que permiten el seguimiento del presupuesto, el progreso del Sprint (Burndown) y el avance global del proyecto (Diagrama de flujo acumulado), entre otros.
  - Crear registros detallados de las reuniones.
- **Alertas Proactivas:** Envían notificaciones sobre elementos del Product Backlog que podrían estar retrasados o requieran atención.
- **Colaboración Remota:** Facilitan la coordinación y colaboración entre equipos que están distribuidos geográficamente, asegurando que todos estén alineados en sus objetivos y tareas.
- **Especialización en Desarrollo:** Para proyectos específicos de desarrollo de software, estas herramientas ofrecen funciones adicionales como:
  - La integración continua.
  - Ejecución de pruebas automatizadas.
  - Trazabilidad completa entre el código y las historias de usuario.

### Tip para el equipo

No todas las herramientas sofisticadas son las mejores. A veces, las herramientas más simples, pero funcionales, pueden ser más eficaces y fáciles de usar para el equipo.

# 3. Enfoque en el valor para los interesados



Dentro del marco Scrum, centrarse en entregar valor constante y significativo para los interesados es primordial. Este pilar se fundamenta en la premisa de que la satisfacción del cliente debe ser prioritaria y continua.

Para asegurarse de que el equipo está en la dirección correcta, es esencial tener en cuenta los siguientes puntos:

- **Entregar incrementos "Terminados":** Asegurarse de que cada incremento del producto cumpla con la Definición de Terminado (DoD) garantiza que los resultados no sólo son completos, sino que también son de alta calidad y aptos para su uso.
- **Priorizar los elementos de mayor valor:** La priorización efectiva del Product Backlog, garantiza que el equipo se enfoque en las características y funciones que ofrecen el mayor retorno de inversión y satisfacción para los interesados.
- **Validar los prototipos con los interesados:** Antes de iniciar el desarrollo completo, es crucial obtener feedback de los interesados sobre los prototipos. Esta práctica asegura que el producto esté alineado con las expectativas y necesidades del cliente desde una etapa temprana.
- **Validación continua en la Revisión del Sprint:** Presentar y validar cada incremento con los interesados durante la Revisión del Sprint permite recolectar retroalimentación directa y realizar ajustes en tiempo real, asegurando así que el producto final cumpla con sus expectativas.

#### Tip para el Scrum Master

Reconoce y celebra cuando el equipo entregue valor real a los interesados. Esto motiva al equipo y refuerza la cultura centrada en el valor.

#### Tip para el Product Owner

Después de la entrega de un incremento o producto completo, realice encuestas para comprender qué tan satisfechos están los interesados y qué áreas se pueden mejorar.

## 4. Disciplina

La disciplina es esencial en Scrum para garantizar que el equipo trabaje armoniosamente y cumpla con los objetivos trazados. Esta rigurosidad no sólo asegura la satisfacción del cliente sino también el funcionamiento fluido del equipo.

Lineamientos que forman parte de la disciplina del equipo:

- Lineamientos establecidos por el Equipo Scrum.
- Lineamientos establecidos por la organización (reglas de negocio).
- Bloques de tiempo asignados a los eventos Scrum (time-box).

- Definición de terminado (DoD) y criterios de aceptación.

#### Tip para el Scrum Master

**La disciplina no significa rigidez:** Si un lineamiento o práctica no está funcionando, el equipo debe sentirse empoderado para discutirlo y hacer los ajustes necesarios.

## Lineamientos establecidos por el equipo Scrum

Estos son los acuerdos o reglas internas que el Equipo Scrum define para garantizar una colaboración y comunicación eficaces entre sus miembros. Por ser autoimpuestos, estos lineamientos refuerzan el pilar de auto-gestión y aseguran que el equipo se adueñe de su proceso. Gracias al pilar de la auto-gestión, son los miembros del Equipo Scrum, quienes establecen sus propios lineamientos, claro, considerando el cumplimiento de las reglas establecidas por la organización.

Los lineamientos del equipo suelen registrarse en un “Plan de Colaboración del equipo Scrum”. Estos lineamientos pueden abordar aspectos como los métodos y herramientas de comunicación preferidos, los horarios de las reuniones, la asignación de roles específicos dentro del equipo, penalizaciones por incumplimientos, los valores del equipo, o las normas sobre cómo y cuándo se realizarán ciertas tareas.

#### Tip para el Scrum Master

Los lineamientos pueden requerir ajustes a medida que el equipo madura. Dedique tiempo en las retrospectivas para revisar y actualizar estos acuerdos.

## Lineamientos establecidos por la organización (reglas de negocio):

Estas son las directrices o reglas que la organización establece, y que todos los equipos, incluidos los Equipos Scrum, deben seguir. Pueden abordar áreas como los estándares de codificación, las políticas de seguridad de la información, los procedimientos de aprobación y revisión, entre otros. Estas reglas aseguran que, aunque el equipo tenga autonomía, su trabajo sigue estando alineado con los objetivos y requisitos más amplios de la organización.

## Bloques de tiempo (time-box)

El marco de Scrum hace un uso extensivo del concepto de time-boxing, donde cada evento tiene una duración máxima asignada. Este enfoque garantiza que las ceremonias o eventos no se prolonguen innecesariamente y que el trabajo se realice de manera puntual y eficiente. Respetar estos bloques de tiempo es crucial para mantener la disciplina y el ritmo del equipo.

Algunas ventajas de establecer y cumplir los Bloques de Tiempo asignado son los siguientes:

- Se evita que el equipo pierda motivación.

- Menos gastos generales en el equipo.
- Se garantiza una alta velocidad para los equipos.
- Las prácticas relacionadas con el desarrollo de entregables son más eficientes.

#### **Recuerda los bloques de tiempo de los eventos de Scrum:**

- Sprint: 1-4 Semanas
- Planificación: 8 Horas para sprints de 1 mes
- Daily: 15 Minutos máximo
- Revisión: 4 Horas para sprints de 1 mes
- Retrospectiva: 3 Horas para sprints de 1 mes

## Definición de terminado (DoD) y criterios de aceptación

La Definición de Terminado es una lista acordada de criterios que deben cumplirse para que una tarea o característica se considere "terminada". Asegura que todo el trabajo cumpla con un estándar mínimo de calidad y esté listo para ser entregado al cliente o puesto en producción.

Los criterios de aceptación, por otro lado, son requisitos específicos que una tarea o característica debe cumplir para ser aceptada por el cliente. Estos criterios ayudan a aclarar las expectativas y garantizar que el equipo esté trabajando hacia el mismo objetivo.

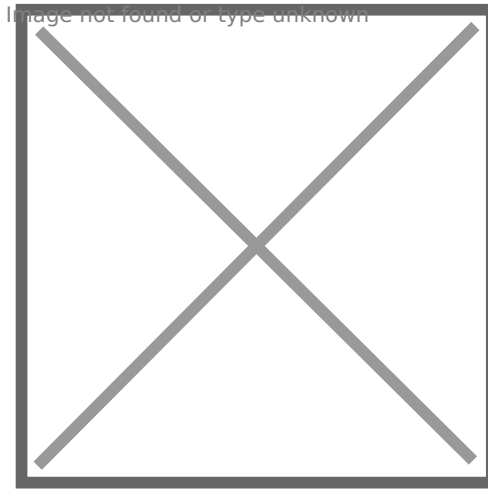
#### **💡 Tip para el Developer**

Aunque puede ser tentador manejar múltiples tareas a la vez, la multitarea puede disminuir la calidad y eficiencia del trabajo. Enfócate en una tarea a la vez y asegúrate de que se complete según la Definición de Terminado.

## 5. Desarrollo iterativo

Dentro de Scrum, el proceso de trabajo se lleva a cabo mediante ciclos repetitivos conocidos como Sprints. Estos Sprints son el núcleo del enfoque iterativo de Scrum, permitiendo flexibilidad y adaptabilidad a lo largo del proyecto.

A diferencia de un enfoque de desarrollo tradicional, que busca tener un plan detallado desde el comienzo, el enfoque iterativo reconoce la naturaleza cambiante del desarrollo de productos y servicios. Al adoptar un enfoque de desarrollo iterativo, Scrum permite a los equipos responder a los cambios y adaptarse a las necesidades emergentes de los clientes, el mercado y la organización.



### **Ventajas del Desarrollo Iterativo:**

- **Entregas Incrementales:** El desarrollo iterativo da lugar a entregas de producto por etapas, ofreciendo valor a los usuarios y partes interesadas desde fases iniciales del desarrollo.
- **Adaptabilidad:** Dada su estructura cíclica, permite ajustes y refinamientos en cada Sprint basados en el feedback obtenido, haciendo que el producto final esté más alineado con las necesidades reales del usuario.
- **Aprendizaje Continuo:** Las retrospectivas al final de cada Sprint brindan al equipo la oportunidad de reflexionar sobre su desempeño y hacer mejoras continuas en su proceso.
- **Estabilidad y Control:** La estructura de Sprint proporciona hitos regulares para revisar y ajustar el progreso, ofreciendo puntos de control y estabilidad al equipo.
- **Transparencia:** Las revisiones frecuentes de Sprint aseguran que las partes interesadas estén constantemente informadas sobre el progreso y estado del producto, reduciendo la incertidumbre y construyendo confianza.

#### **Tip para el Scrum Master**

Promueve una cultura donde cada miembro del equipo esté comprometido con la mejora continua, no sólo en términos del producto sino también en términos de procesos y colaboración.

# 3.4 - El Scrum Master y los eventos

En Scrum cada evento (ceremonia) tiene un propósito específico, diseñado para maximizar la transparencia, la colaboración y la eficiencia. **El Scrum Master**, a menudo considerado como el guardián de la agilidad del equipo, juega un papel esencial en estos eventos. Su responsabilidad no se limita a la simple organización o facilitación; va mucho más allá, actuando como un catalizador que asegura que cada ceremonia genere el máximo valor, alineando al equipo con los valores de Scrum.

En esta sección, exploraremos en detalle cómo el Scrum Master navega por cada evento, qué se espera de él y cómo su participación activa puede marcar la diferencia en la dinámica y resultados del equipo Scrum.

## El Scrum Master en la Planificación del Sprint

- Apoya al Product Owner a moderar la participación de todos
- Cuida que se cumpla con la agenda y duración del evento
- Asiste al Product Owner en la definición del trabajo a realizar
- Apoya en la resolución de dudas que tengan los Developers
- Apoya a los Developers al momento de realizar las estimaciones del trabajo

**Caso práctico:** Durante una sesión de planificación del sprint, algunos desarrolladores muestran confusión sobre una historia de usuario. El Scrum Master, Pablo, interviene: "Vamos a aclarar esta historia antes de continuar". Facilita una sesión de preguntas rápidas con el Product Owner, María, eliminando las dudas. Cuando llega el momento de las estimaciones, Pablo propone un "Planning Poker", logrando consenso. Al finalizar, recalca la importancia de mantener la comunicación abierta durante el Sprint.

### Tip para el Scrum Master

Asegúrate de que todos los elementos seleccionados para el Sprint estén claros y bien definidos. Si notas que hay dudas o incertidumbre, tómate un momento para clarificarlas con el Product Owner antes de que el equipo empiece a estimar.

## El Scrum Master en el Daily

- Apoya a los Developers en la realización del evento, y que se respete su duración
- Identifica los impedimentos que puedan encontrar los Developers
- Asegura que el Daily se lleve a cabo a la misma hora y en el mismo lugar
- Si el Scrum Master está trabajando activamente en los elementos del Sprint, entonces participa como un Developer

**Caso práctico:** Al tercer día del Sprint, los Developers se reúnen puntualmente para el Daily Scrum. Pablo, el Scrum Master, se asegura de que todos estén presentes y de que se inicie a tiempo. Cuando Sofía, una Developer, menciona que está teniendo problemas técnicos con una herramienta, Pablo toma nota del impedimento para tratarlo después del Daily. Aunque Pablo está trabajando en una tarea específica de este Sprint, mantiene su participación breve y centrada, como si fuera otro Developer. Al finalizar, Pablo recuerda a todos la importancia de ser consistentes con la hora y el lugar del Daily, reforzando la rutina y previsibilidad.

#### Tip para el Scrum Master

Si durante el Daily un Developer empieza a profundizar demasiado en un impedimento, tú como Scrum Master debes intervenir amablemente y recordar al equipo que el Daily tiene una duración limitada. Sugiere abordar el impedimento en detalle justo después del Daily o programar un momento específico para tratarlo, asegurando que el foco del Daily se mantenga en la actualización rápida de cada miembro del equipo.

## El Scrum Master durante el desarrollo del Sprint

- Lleva el seguimiento al progreso de los Developers con respecto a lo planificado
- Gestiona y resuelve impedimentos que puedan aparecer (en su mayoría identificados en el Daily)
- Gestiona la resolución de conflictos que puedan surgir entre los Developers
- Trabaja en estrategias que aumenten la motivación de los Developers
- Asegura que los Developers no son interrumpidos durante el Sprint
- Cuida que los Developers cumplan con los criterios de aceptación y terminado (DoD)

**Caso práctico:** A mitad del Sprint, Pablo, el Scrum Master, nota que el equipo está un poco detrás de lo planificado. Un impedimento principal es la falta de acceso a un servidor que se identificó en el Daily. Pablo toma la iniciativa, habla con el equipo de infraestructura y resuelve el acceso al servidor al día siguiente.

Días después, observa una discusión entre Sofía y Jorge (dos Developers) sobre la mejor manera de implementar una función. Pablo facilita una breve sesión para comprender los puntos de vista y logra que ambos lleguen a un acuerdo, manteniendo el ambiente armonioso.

Finalmente, al aproximarse el final del Sprint, Pablo realiza pequeñas revisiones con los Developers para asegurarse de que todos los elementos del trabajo estén alineados con los criterios de aceptación y la Definición de Terminado (DoD).

#### Tip para el Scrum Master

Mantente disponible y accesible. Los impedimentos pueden surgir en cualquier momento, y tu capacidad para actuar rápidamente puede hacer la diferencia entre un Sprint exitoso y uno problemático.

## El Scrum Master en la Revisión del Sprint

- Apoya al Product Owner a moderar la participación de todos
- Cuida que se cumpla con la agenda y duración del evento
- Asegura que se presenten los resultados logrados en el Sprint
- Asegura que el progreso respecto al Objetivo de Producto es discutido en esta sesión
- Recopila información que permita medir el desempeño del equipo

**Caso práctico:** Al finalizar el Sprint, es momento de la revisión. Pablo, el Scrum Master, asegura que la sala de reuniones está lista y que todos los recursos tecnológicos funcionan correctamente para evitar interrupciones.

Mientras Carla, el Product Owner, se prepara para revisar las características que completaron los Developers durante el Sprint, Pablo se mantiene alerta para moderar y garantizar que se respeten los tiempos y que todos los miembros tengan la oportunidad de participar.

Pablo también se encarga de recopilar comentarios y preguntas que surgen durante la presentación, y se asegura de que, al finalizar, el equipo y los stakeholders (si los hay) discutan el progreso en relación con el Objetivo de Producto.

Una vez terminada la revisión, Pablo recoge feedback sobre la revisión, buscando oportunidades de mejora para las próximas sesiones.

#### Tip para el Scrum Master

No te enfoques únicamente en lo que se completó: Asegúrate de celebrar los logros del equipo y reconoce los esfuerzos. Al mismo tiempo, fomenta una discusión abierta sobre lo que no se logró y las razones detrás de ello, para crear un ambiente de aprendizaje continuo.

## El Scrum Master en la Retrospectiva del Sprint

- Guía y modera el evento
- Aplica estrategias y técnicas para llevar a cabo la retrospectiva
- Brinda la confianza y seguridad para que el equipo sea sincero con respecto a la reflexión sobre los resultados del Sprint
- Asiste el equipo en la identificación de lecciones aprendidas
- Guía la reflexión del equipo acerca de su rendimiento (tal como la velocidad y el progreso)

**Caso práctico:** Tras la Revisión del Sprint, Pablo, el Scrum Master, organiza la Retrospectiva. Pablo comienza recordando la importancia de la sinceridad y el respeto mutuo durante el evento, y decide emplear la técnica "Estrella de Retro" para abordar distintas facetas del trabajo del equipo. Como facilitador, cuando un punto particularmente delicado sale a la luz, Pablo interviene para asegurarse de que la discusión sea productiva y enfocada en las soluciones, y va tomando notas de lecciones aprendidas y áreas de mejora. Al final, se asegura de que todos los acuerdos se registren y sean visibles para el próximo Sprint. Agradece al equipo por su compromiso y les anima a implementar las mejoras acordadas.

#### Tip para el Scrum Master

Cultiva un ambiente seguro y de confianza: Recuerda que la Retrospectiva no es un espacio para señalar errores individuales, sino para identificar áreas de mejora como equipo. Fomenta la empatía y escucha activa, evitando juicios prematuros o asumir posturas defensivas.



# 3.5 - Técnicas útiles para el Scrum Master

## Para guiar el Daily Scrum

Una de las dudas al momento de llevar a cabo el daily es saber quién se supone que debe hablar primero y en qué orden, cómo mantener la atención, por lo que se proponen las siguientes estrategias:

### Pasa la ficha (token de habla)

Esta técnica busca estructurar la comunicación y asegurar una participación activa y ordenada de todos los miembros. Se utiliza un objeto (pelota, juguete, etc.) para determinar quién habla a continuación. Sólo la persona que tiene el objeto puede hablar y lo entrega aleatoriamente a otra persona para asegurar que todos prestan atención. Con esta estrategia se evitan interrupciones, se asegura que todos sean escuchados y puede agregar un elemento de diversión a la reunión que puede aliviar la tensión y hacer que la reunión sea más amena.

### Toma una carta

Esta es una técnica útil para evitar la monotonía y dar a todos una oportunidad de hablar en diferentes órdenes cada día. Al comienzo del Daily, se reparten cartas numeradas a los miembros del equipo. El número en la carta determinará el orden de habla.

### Caminar por el tablero

En lugar de que cada miembro del equipo simplemente responda las tres preguntas tradicionales, el equipo puede "caminar" por el tablero de Scrum, abordando cada historia de usuario o tarea, desde la más alta prioridad hacia abajo (será imprescindible que el tablero esté actualizado). Con esta estrategia se mantiene el enfoque en el trabajo real, se pueden identificar bloqueos de manera efectiva y se garantiza que las tareas más críticas sean discutidas.

Estas técnicas no son excluyentes y se pueden combinar según las necesidades del equipo. Lo más importante es mantener el propósito del Daily Scrum: asegurarse de que el equipo esté alineado, identificar bloqueos y tomar medidas para avanzar eficazmente hacia el objetivo del sprint.

# Recomendaciones adicionales para el Daily Scrum

A continuación se presentan recomendaciones para mejorar la ejecución del daily, lograr que se cumpla con los objetivos de este evento, y mantener la atención del equipo:

- **Ponerse de pie:** Si el equipo está físicamente presente, considera formar un círculo o semicírculo para que todos puedan verse y escucharse claramente. Ponerse de pie conlleva a la incomodidad física lo que le recuerda a los asistentes que el daily está demorando demasiado.
- **Cronometrar el evento:** Para limitar la duración del evento y acondicionar al equipo para mantener el enfoque en el objetivo. Esto mantiene el ritmo de la sesión, garantiza la eficiencia y evita divagaciones.
- **Rotar al facilitador:** Cada día, un miembro diferente del equipo puede facilitar el Daily Scrum. Esto puede rotar en un orden definido o ser aleatorio, lo que permite eliminar la dependencia con un único facilitador o con el SM para poder iniciar el daily. Rotar al facilitador fomenta la responsabilidad compartida, introduce variedad y permite que diferentes voces guíen la reunión.
- **Evitar las largas historias:** La información concisa hace que el equipo mantenga la atención en lo que se habla en el daily.

## Para guiar la Retrospectiva del Sprint

Es muy común que el equipo pueda perder el interés en participar en la retrospectiva por considerarla aburrida, por lo que existen diversas técnicas que facilitan la transparencia y la adaptación.

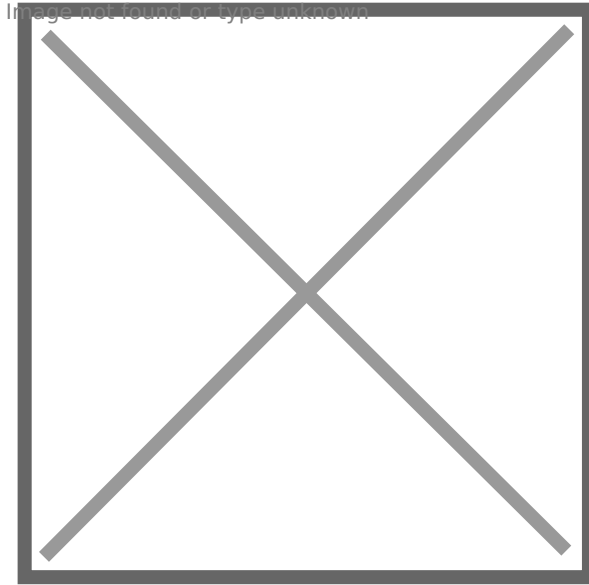
- No existe una técnica mejor que otra. Hay técnicas que funcionan muy bien en un equipo y fracasan en otro, por lo que la recomendación es que se prueben varias técnicas.
- No todas las dinámicas potencian lo mismo. El momento, la situación, e incluso la personalidad del equipo, son cosas que se tienen en cuenta a la hora de elegir la dinámica a usar.
- Es necesario que, al terminar la retrospectiva se concluya con una o varias acciones concretas y realizables

### Retrospectiva de las 4L's

Algunos miembros del equipo suelen decir cosas como "si tan solo tuviéramos...", "Ojalá tuviéramos...", o tal vez digan cosas como "adivina lo que descubrí ...", "realmente me encantó ..."

La retrospectiva de 4Ls está diseñada para que las personas compartan esos pensamientos con el objetivo de mejorar continuamente.

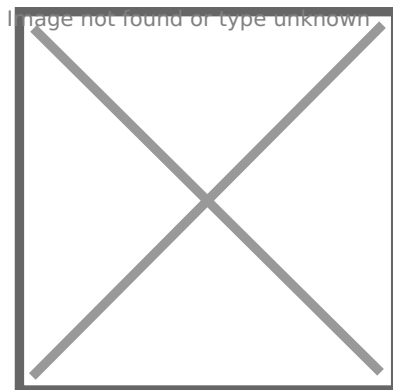
- **Nos gustó:** ¿Qué le gustó al equipo del último Sprint? Podría ser cualquier cosa, desde un proceso, un logro, una acción de equipo en particular o incluso una tecnología.
- **Aprendimos:** ¿Qué cosas aprendió el equipo de experimentos, pruebas, conversaciones y del trabajo entre ellos?
- **Nos faltó:** ¿Qué parecía faltar en el último Sprint?
- **Anhelamos:** Es algo que desearían que existiera o que fuera posible para asegurar el éxito del trabajo.



## Retrospectiva del barco rápido (speed boat)

Speed Boat es una técnica en la que los miembros del equipo simulan que son tripulantes en un barco, el cual debe llegar a una isla (que representa el Objetivo del Producto). Es una forma visual y metafórica de identificar lo que ayuda y obstaculiza al equipo.

- Los remos nos ayudan a avanzar (cosas que hacemos bien)
- Los motores nos ayudan a llegar eficientemente a la isla (cosas por adoptar o mejorar)
- Los anclajes nos impiden llegar a la isla (cosas que hacemos mal)



**Nota:** Se pueden usar notas adhesivas para identificar cada aspecto.

## Comenzar, Detener, Continuar (Start, Stop, Continue)

Los miembros del equipo escriben acciones que les gustaría que el equipo comience a hacer, detenga y continúe haciendo. Esta técnica es sencilla, pero efectiva, para identificar rápidamente áreas de mejora y aquellas en las que el equipo está yendo bien.

- **Start/Comenzar:** Aquellas tareas o prácticas que parecen ser positivas para conseguir los objetivos trazados en el equipo y que aún no se están haciendo
- **Stop/Parar:** Aquellas prácticas que no son tan eficientes o que no están dando los resultados pensados y que se deberían dejar de hacer
- **Continue/Continuar:** Todo aquello que se está haciendo, que funciona y que se debe seguir realizando.

## Mad, Sad, Glad

Esta técnica busca revelar las emociones del equipo sobre eventos, decisiones y circunstancias que ocurrieron durante el Sprint. Se prepara un tablero dividido en tres secciones: **Mad (Enojado), Sad (Triste) y Glad (Feliz)**. Se puede usar un tablero físico con post-its o mediante herramientas digitales.

Cada miembro del equipo reflexiona sobre el sprint pasado y escribe en tarjetas o post-its individuales las cosas que les hicieron sentir enojados, tristes o felices (una tarjeta por cada sentimiento); y colocan sus tarjetas en la sección correspondiente del tablero. Cuando todos hayan terminado, uno por uno explica brevemente por qué escribió cada punto.

Una vez que todos han compartido sus sentimientos se discuten los patrones o temas recurrentes que surgen. ¿Hay muchos post-its sobre un tema particular en la sección "Mad"? Eso podría ser una señal de un problema que necesita ser abordado. Al profundizar en las causas subyacentes de los sentimientos, especialmente en los que generan emociones fuertes, puede revelar problemas más profundos o áreas de mejora.

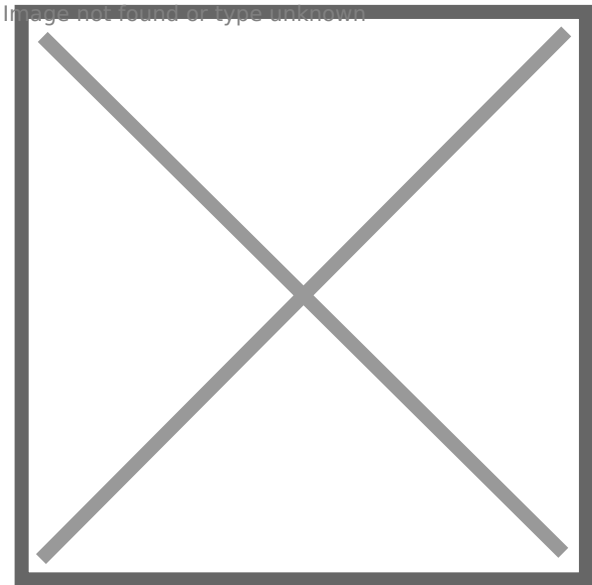
Basándose en la discusión, se identifican las acciones específicas que el equipo puede tomar para abordar los problemas o circunstancias que causaron emociones negativas y para reforzar lo que causó emociones positivas.

## Estrella de Retro (Starfish Retrospective)

Esta técnica ayuda al equipo a reflexionar sobre diferentes áreas y decidir qué deben **comenzar a hacer, dejar de hacer, continuar haciendo, hacer más y hacer menos**.

Dibuja una estrella o estrella de mar en un tablero con cinco secciones: "Comenzar", "Detener", "Continuar", "Hacer más" y "Hacer menos". Cada miembro del equipo escribe en tarjetas o post-its sus opiniones o comentarios sobre lo que el equipo debe comenzar a hacer, dejar de hacer, etc. y los colocan en la sección correspondiente de la estrella.

Se conduce una discusión sobre cada sección. ¿Hay consenso sobre ciertos puntos? ¿Hay ideas contradictorias? Se identifican y priorizan las acciones específicas que surgieron de la discusión.



# 3.6 - Emisores de información útiles para el Scrum Master

**El término "emisor de información" no hace parte de los conceptos core de Scrum.** Sin embargo, en el contexto de la comunicación y la gestión ágil de producto, los emisores de información son herramientas diseñadas para mostrar el estado actual de los Sprints o del producto en su totalidad.

Estos instrumentos ofrecen una visualización clara y concisa de tareas, progresos y obstáculos, facilitando la transparencia y permitiendo un seguimiento y control efectivos. Sirven como medios esenciales para que todos los involucrados, desde el equipo Scrum hasta los stakeholders, tengan una comprensión clara del avance y estado del trabajo.

## El Scrum Board

El Scrum Board o también conocido como tablero Scrum es una adaptación del tablero Kanban que nos permite dar seguimiento a cada Sprint Backlog dentro del equipo Scrum.

El Scrum Board es un emisor de información que le permite al Equipo garantizar la transparencia en los Sprints, mantiene la coordinación y permite realizar al Scrum Master sus labores de Inspección.

Comúnmente está compuesto por 4 columnas: Pendiente (to do), En Progreso (in progress), En Prueba (testing), Terminado (done).

 or type unknown

Cada elemento por desarrollar debe ponerse en una tarjeta individual, y dado que el Scrum Board se actualiza constantemente, todas las tarjetas deberán pasar por las 4 columnas. (No se pueden hacer saltos de columnas).

### Aclaraciones sobre el Scrum Board

- Cuando el tablero está físico en el lugar de trabajo del equipo, generalmente las tarjetas que se manejan son “sticky notes” de distintos colores.

- Cuando se van a desarrollar múltiples historias de usuario en un solo Sprint conviene dividir el tablero en filas, donde cada fila representa un componente o épica del producto.
- Al final de un Sprint se “borra” o reinicia el Scrum Board (para así poderlo usar para el próximo Sprint)
- Para considerar un elemento como “terminado” y moverlo a esta zona en el Scrum Board, se debe considerar la “Definición de terminado (DoD)”.
- Para facilitar la lectura del Scrum Board, a cada miembro del equipo puede asignársele un color de tarjetas (a manera de convenciones).
- Aunque por lo general el Scrum Board se actualiza durante la reunión diaria, cada Developer tiene autonomía para actualizar su conjunto de elementos asignados en el Sprint Backlog.

# Burndown chart del Sprint

El “Burndown Chart” es un emisor de información que muestra la cantidad de trabajo pendiente en el Sprint actual en función del tiempo.

Es un emisor gráfico de 2 dimensiones:

- El eje vertical muestra el trabajo restante, que se determina con la sumatoria de puntos de historia a desarrollar en el Sprint
- El eje horizontal muestra el tiempo, es decir la duración del Sprint en días hábiles.

**burndown chart**  
Image description: type unknown

Al inicio del Sprint, la cantidad de trabajo restante es la más alta y, a medida que se avanza el Sprint, esta cantidad debería disminuir, "quemándose" (burn) el trabajo hasta llegar a cero al final del Sprint.

El Burndown Chart es una herramienta esencial para los equipos Scrum ya que les permite visualizar diariamente su progreso, identificar posibles problemas y ajustar su ritmo de trabajo si es necesario para cumplir con el Objetivo del Sprint (Sprint Goal).

## Aclaraciones sobre el Burndown Chart

- Una posible variación de esta herramienta es el "**Burnup chart**" que muestra **gráficamente la cantidad de trabajo completado** a lo largo del tiempo, en lugar de lo que queda por hacer (a medida que avanza el tiempo, la línea del gráfico asciende, mostrando cuánto trabajo se ha "acumulado" o completado).
- Los **Developers son los responsables de la actualización del Burndown Chart**. El Scrum Master puede facilitar y asegurarse de que el Burndown Chart se mantenga actualizado, pero no es su responsabilidad directa hacerlo.
- Por lo general la actualización se realiza durante los Scrum Diarios.

# Diagrama de flujo acumulado

El Diagrama de Flujo Acumulado (CFD - Cumulative Flow Diagram) es un emisor de información bastante útil para la elaboración de informes y el seguimiento de los resultados del proyecto; y se originó a partir de la metodología Kanban.

Este emisor de información muestra el progreso del producto respecto a los ítems del Product Backlog, y se construye un único diagrama de flujo acumulado por producto.

Este gráfico se compone de dos ejes; el eje vertical "Y" muestra la cantidad de ítems (como tareas, características, user Historias de Usuario). El eje horizontal "X" muestra el tiempo, usualmente días, semanas o meses.

flujo acumulado



Cada área coloreada en el gráfico representa un estado o categoría diferente del flujo de trabajo:

- **La zona "gris"** muestra el comportamiento del Product Backlog; y el aumento de esta zona significa la adición de nuevos requerimientos/tareas/historias de usuario. La disminución de esta zona significa que se retiran requerimientos/tareas/historias de usuario que ya no generan valor (como resultado de los cambios al producto).
- **La zona "azul"** muestra el trabajo que fue seleccionado por el equipo para desarrollar en los distintos Sprints, es decir el trabajo que se seleccionó como Sprint Backlogs.
- **La zona "verde"** muestra cuántos ítems han sido completados en el tiempo. Idealmente, deberías ver un crecimiento estable en esta zona si el trabajo se está completando a un ritmo constante. Cuando la zona azul está por encima de la zona verde, normalmente significa que el equipo seleccionó más trabajo del que podía terminar, esto se conoce como "deuda técnica".
- La diferencia entre la zona "verde" y la zona "gris", está relacionada con el progreso del producto.
- Los puntos marcados en el gráfico muestran los distintos Sprints.

## Aclaraciones sobre el Diagrama de Flujo Acumulado:

- Si una zona del gráfico se ensancha mientras las demás no, es una señal de que los ítems se están acumulando en ese estado y puede indicar que hay un cuello de botella.
- Si las zonas fluyen de manera paralela y sin muchas variaciones en su ancho, indica que el proceso es estable. Si se expanden y contraen drásticamente, puede indicar que hay inconsistencias en el flujo de trabajo.
- El responsable de este emisor de información es el Scrum Master, aunque normalmente es el Product Owner quien lo utiliza para mostrar el progreso a las partes interesadas.



# Registro de obstáculos/impedimentos

El registro de obstáculos, o también llamado registro de impedimentos es una herramienta que se utiliza para llevar un registro y seguimiento de los obstáculos o bloqueos que impiden o ralentizan el progreso del equipo Scrum. Estos impedimentos pueden ser cualquier cosa que obstruya el flujo de trabajo del equipo, desde problemas técnicos hasta cuestiones organizativas.

Un impedimento es cualquier factor que impide que el equipo alcance su productividad potencial. Puede ser algo interno, como un desafío técnico o falta de habilidades específicas, o externo, como la espera de una aprobación de otro equipo o departamento.

Mantener un registro de impedimentos ayuda a asegurar que los problemas no se pasen por alto y que se les dé la debida prioridad. También proporciona una visión histórica que puede ser útil para futuras retrospectivas y para identificar patrones o problemas recurrentes.

## Aclaraciones sobre el registro de impedimentos

- El Scrum Master es el rol encargado de abordar y eliminar los impedimentos que estén afectando al equipo. Sin embargo, es responsabilidad de todo el equipo identificarlos y comunicarlos.
- Durante el Scrum Diario (o Daily), los miembros del equipo informan sobre cualquier nuevo impedimento que haya surgido. El Scrum Master toma nota de estos y los añade al registro de impedimentos.
- Cada entrada en el registro de impedimentos debe tener una descripción clara del problema, la fecha en que fue identificado, su impacto potencial en el equipo o en el producto y cualquier otro detalle relevante.
- También es útil incluir el estado del impedimento (por ejemplo: identificado, en progreso, resuelto) y las acciones tomadas o necesarias para resolverlo.
- En algunas organizaciones este registro es global para todos los proyectos o equipos y sirve como base de conocimiento para todos los integrantes de los distintos equipos Scrum.

# Registro de lecciones aprendidas

Registrar las lecciones aprendidas es una parte fundamental para la mejora continua de las prácticas Scrum dentro de una organización. En esta herramienta se registran las prácticas,

experiencias, conocimientos y recomendaciones que favorecieron el desempeño del equipo en términos de costo, tiempo y calidad, también se registran aquellos errores que se cometieron y a los que se les dio solución. Estas soluciones hacen parte de las lecciones aprendidas, útiles para resolver problemas similares en el futuro.

Las lecciones aprendidas, además, favorecen el aprendizaje organizacional y la gestión del conocimiento. Así que las lecciones aprendidas deben almacenarse, alimentarse y transferirse de manera apropiada. Para esto, es importante que cada lección tenga una estructura para facilitar su búsqueda, como por ejemplo:

- Una identificación (ID)
- Descripción: Breve resumen de la situación o problema.
- Situación/Contexto: En qué fase o circunstancia del proyecto se encontró la lección.
- Lección Aprendida: Detalle de lo que se aprendió, ya sea un error que se debe evitar o una buena práctica que se debe repetir.
- Recomendaciones: Pasos o acciones sugeridos para futuros proyectos a partir de la lección aprendida.
- Responsables: Quiénes estuvieron involucrados o quiénes deberían ser informados sobre esta lección.
- Fecha: Cuándo se identificó la lección.
- Categoría: Por ejemplo, la fase del ciclo de vida del proyecto

#### **Aclaraciones sobre el registro de lecciones aprendidas:**

- Este artefacto normalmente es una de las responsabilidades del Scrum Master.
- Los equipos suelen tener reuniones específicas para discutir y documentar lecciones aprendidas; que pueden darse durante las sesiones de retrospectiva o en revisiones post-proyecto (cuando se manejan enfoques más tradicionales).
- Es crucial que estas sesiones se lleven a cabo en un ambiente abierto y sin culpas, para fomentar la sinceridad y el aprendizaje genuino.
- Al iniciar un nuevo proyecto o la construcción de un producto, el equipo debe revisar registros anteriores para identificar posibles riesgos, desafíos o mejores prácticas relevantes.
- La organización o el equipo debe tener un sistema para almacenar y acceder a estas lecciones, ya sea una base de datos, un repositorio centralizado o incluso herramientas digitales especializadas.

## Portafolio de proyectos

El portafolio de proyectos consiste en un registro de información histórica, esta información podría ser útil (por ejemplo) cuando la organización empieza un proyecto similar a uno que haya ejecutado en el pasado y por tanto algunos elementos sean similares, esto reduce el tiempo de planificación, desarrollo, revisión, implementación y/o cierre, además de hacer uso de las lecciones aprendidas y brindar los insumos para la gestión del conocimiento.

Al igual que las lecciones aprendidas, el portafolio de proyectos debería tener información detallada y organizada que permita su alimentación, almacenamiento y transferencia. Un software específico para la gestión del portafolio de proyectos es útil, por lo que una organización debería considerar esta posibilidad.

Algunos de los elementos que podrían incluirse en el portafolio de proyectos son:

- Una identificación (ID)
- Nombre del Proyecto
- Categoría (Ej: Software – Implementación – Construcción – Investigación, Etc)
- Descripción corta del proyecto (buscando resumir la visión del mismo)
- Tiempo total de proyecto
- Costo total del proyecto
- Enlace a repositorio o herramienta de información (donde se podrá encontrar más información del proyecto)