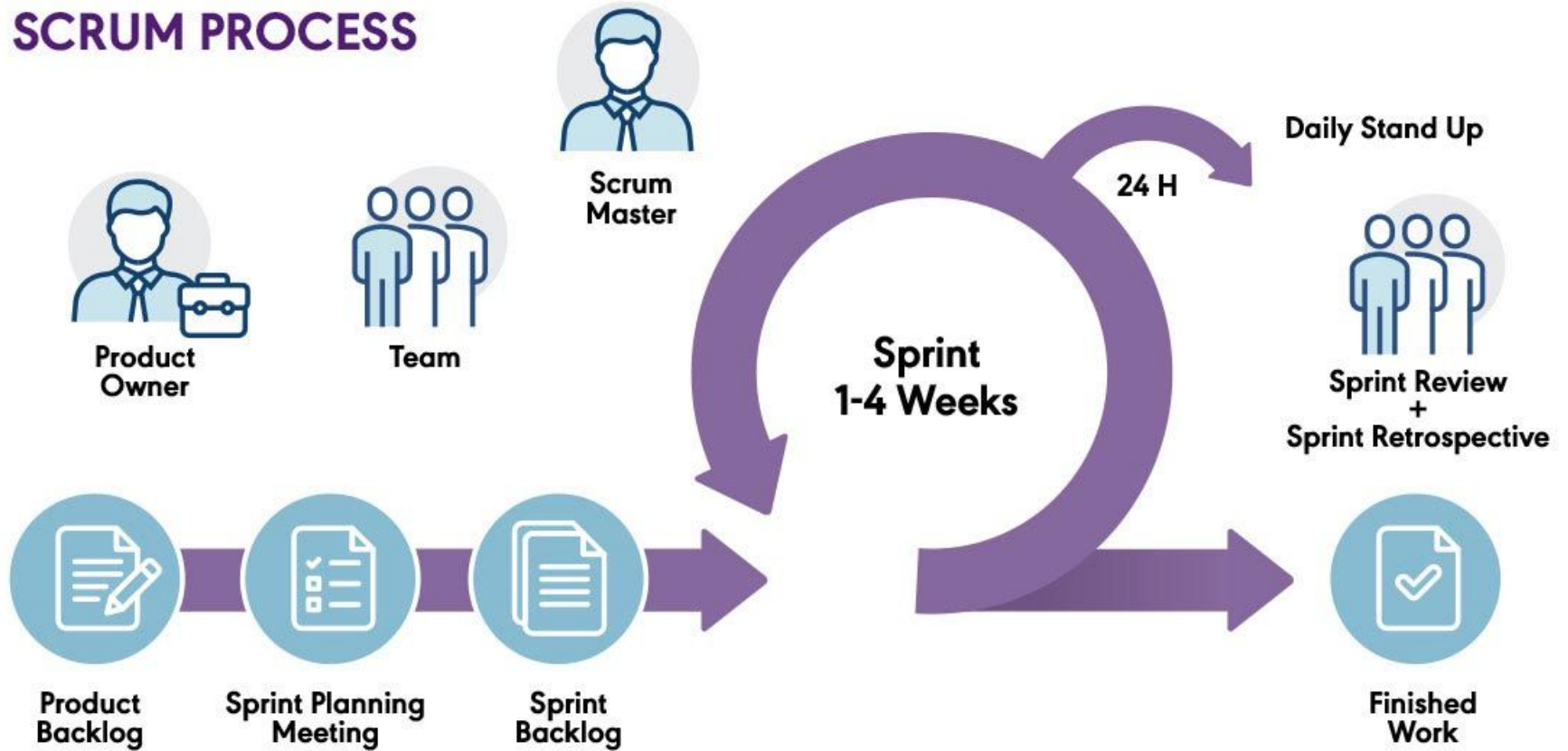


# SCRUM PROCESS




# SCRUM

Scrum es un marco de trabajo para desarrollo ágil de software que se ha expandido a otras industrias.

Es un proceso en el que se aplican de manera regular un conjunto de buenas prácticas para trabajar colaborativamente, en equipo y obtener el mejor resultado posible de proyectos.

Scrum es el marco de trabajo **más utilizado hoy en día** por empresas de desarrollo de software, gracias a su gran capacidad para desarrollar incrementalmente, y su organización inmaculada.



# Origen de Scrum

Este modelo fue identificado y definido por Ikujiro Nonaka y Takeuchi a principios de los 80, al analizar cómo desarrollaban los nuevos productos las principales empresas de manufactura tecnológica (Canon, Honda, Fuji, Epson, Hewlett-Packard). En su estudio, Nonaka y Takeuchi compararon la nueva forma de trabajo en equipo, con el avance en formación de **scrum** de los jugadores de **Rugby**, a raíz de lo cual quedó acuñado el término “scrum” para referirse a ella.



Aunque esta forma de trabajo surgió en empresas de productos tecnológicos, es apropiada para cualquier tipo de proyecto con requisitos inestables y para los que requieren rapidez y flexibilidad, situaciones frecuentes en el desarrollo de determinados sistemas de software.

# Origen de Scrum

Nonaka y Takeuchi describieron una nueva manera de desarrollar productos comerciales, que incrementarían ampliamente la velocidad y flexibilidad, basado en casos de estudio de las compañías mencionadas previamente. Estos casos de estudio fueron llevados a cabo principalmente por Schwaber y Ogunnaike, en la Universidad de Delaware.

En estos estudios, Ogunnaike advirtió que los intentos de desarrollar productos complejos, como software, que no estuvieran basados en el empirismo, estaban condenados a mayores riesgos y altas tasas de fracaso a medida que cambiaban las condiciones iniciales y los supuestos. El empirismo, utilizando inspección y adaptación frecuentes, con flexibilidad y transparencia, es el enfoque más adecuado.

**Canon**



**EPSON**

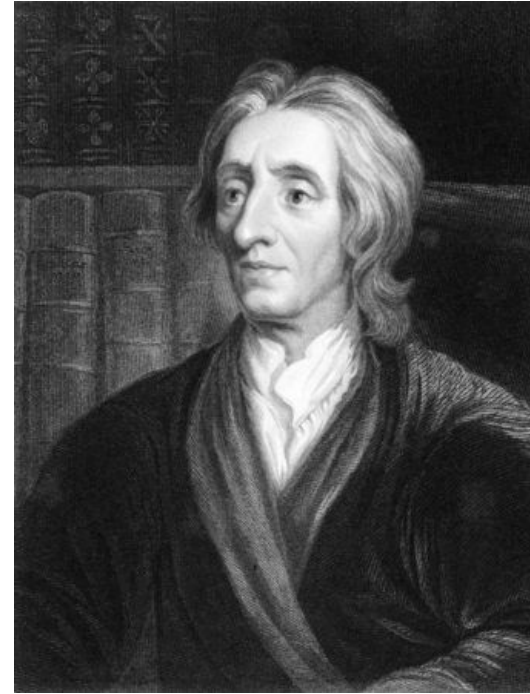


# Empirismo? 🤔

Empirismo (Locke), dice que la experiencia y la evidencia es el punto de partida y fundamento de todo conocimiento posible. Es así que el empirismo sostiene que todo nuestro conocimiento del mundo proviene de los sentidos, a su vez que la experiencia misma posee un papel epistémico insustituible en la justificación de la creencia.

Es así que la idea de Scrum es tomar decisiones basados en la información concreta obtenida de la observación que muestra el progreso del desarrollo de producto, los cambios en el mercado y los comentarios de los clientes.

Lo contrario al empirismo (o scrum), es usar planificación previa, procesos definidos, planes predictivos, hechos no concretos.



# Scrum

Scrum es un proceso en el que se aplican de manera regular un conjunto de buenas prácticas para trabajar colaborativamente, en equipo, y obtener el mejor resultado posible de un proyecto. Estas prácticas se apoyan unas a otras y su selección tiene origen en un estudio de la manera de trabajar de equipos altamente productivos.

En Scrum se realizan entregas parciales y regulares del producto final, priorizadas por el beneficio que aportan al receptor del proyecto. Por ello, Scrum está especialmente indicado para proyectos en entornos complejos, donde se necesita obtener resultados pronto, donde los requisitos son cambiantes o poco definidos, donde la innovación, la competitividad, la flexibilidad, y la productividad, son fundamentales.



# Sprint

Sprint es el nombre que va a recibir cada uno de los ciclos o iteraciones que vamos a tener dentro de un proyecto Scrum.

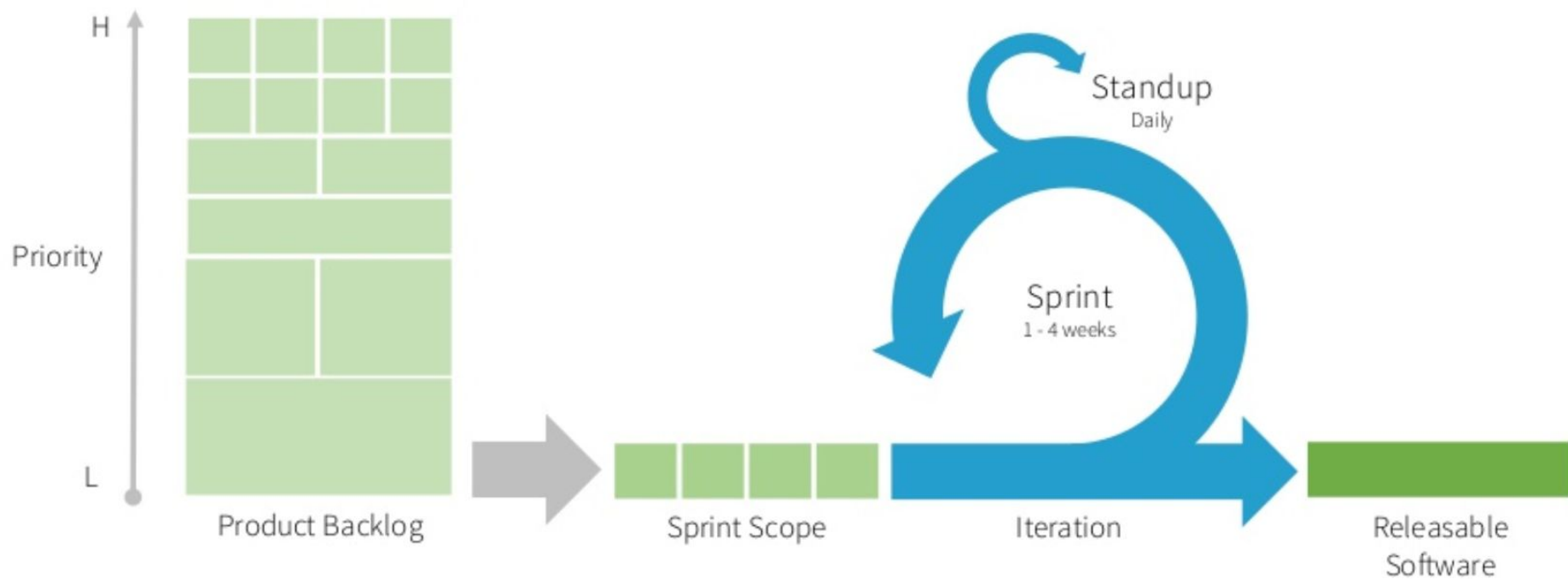
Nos van a permitir tener un ritmo de trabajo con un tiempo prefijado, siendo la duración habitual de un Sprint unas dos semanas, aunque lo que la metodología dice, es que debería estar entre una semana y un máximo de un mes.

En cada Sprint o cada ciclo de trabajo lo que vamos a conseguir es lo que se denomina un entregable o incremento del producto, que aporte valor al cliente.

La idea es que cuando tenemos un proyecto bastante largo, por ejemplo un proyecto de 12 meses, vamos a poder dividir ese proyecto en doce Sprints de un mes cada uno. En cada uno de esos Sprints vamos a ir consiguiendo un producto, que siempre, y esto es muy importante, sea un producto que esté funcionando.



# Traditional Scrum





# Actores o roles principales

El Dueño del Producto (**Product Owner**), representa a los inversionistas o las personas que requieren el software.

El Director Scrum (**Scrum Master**), es el facilitador del equipo, supervisa al equipo y verifica que se lleven a cabo las reuniones y se haga uso de los artefactos. Ayuda a que el proyecto tenga éxito. Elimina los problemas e impedimentos que se pudieran presentar. Ayuda a los miembros del equipo a tomar decisiones responsables y los asesora en todas las maneras posibles para que alcancen sus objetivos.

Los miembros del equipo (**Team Members**), son los que desarrollan el software, poseen las capacidades técnicas para fabricar el producto.



# Product Owner

El **Product Owner** es el responsable de maximizar el valor del producto resultante del trabajo del equipo Scrum. La forma en que se hace esto puede variar ampliamente entre organizaciones, equipos Scrum e individuos.

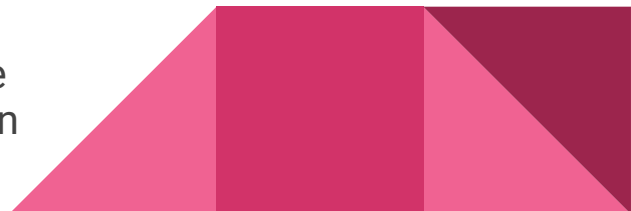
Como miembro del Equipo Scrum, el **Product Owner** proporciona claridad al equipo sobre la visión y el objetivo de un producto. Todo el trabajo se deriva y se prioriza, en función del objetivo del producto, para ofrecer valor a todas las partes interesadas, incluidas las de su organización y todos los usuarios, tanto internos como externos. Los Product Owners identifican, miden y maximizan el valor a lo largo de todo el ciclo de vida del producto.



# Scrum Master

El Scrum Master es responsable de promover y apoyar el modelo Scrum. Los Scrum Masters hacen esto al ayudar a todos a comprender la teoría, las prácticas, las reglas y los valores de Scrum. Tienen dos funciones principales:

1. **Gestionar el proceso Scrum:** el Scrum Master se encarga de gestionar y asegurar que el proceso Scrum se lleva a cabo correctamente, así como de facilitar la ejecución del proceso y sus mecánicas. Siempre atendiendo a los tres pilares del control empírico de procesos y haciendo que la metodología sea una fuente de generación de valor.
2. **Eliminar impedimentos:** esta función del Scrum Master indica la necesidad de ayudar a eliminar progresiva y constantemente impedimentos que van surgiendo en la organización y que afectan a su capacidad para entregar valor, así como a la integridad de esta metodología. El Scrum Master debe ser el responsable de velar porque Scrum se lleve adelante, transmitiendo sus beneficios a la organización facilitando su implementación.



# Team Members o equipo de desarrollo

El equipo de desarrollo suele estar formado por entre 3 a 9 profesionales que se encargan de **desarrollar el producto**, auto-organizándose y auto-gestionándose para conseguir entregar un incremento de software al final del ciclo de desarrollo.

El equipo de desarrollo se encargará de crear un incremento terminado a partir de los elementos del Product Backlog seleccionados (Sprint Backlog) durante el Sprint Planning. (ahí vemos qué significa esto).

Es importante que en la metodología Scrum todos los miembros del equipo de desarrollo conozcan su rol, siendo solo uno común para todos, independientemente del número de miembros que tenga el equipo y cuales sean sus roles internos. Cómo el equipo de desarrollo decida gestionarse internamente es su propia responsabilidad y tendrá que rendir cuentas por ello como uno solo; hay que evitar intervenir en sus dinámicas.

Habitualmente son equipos 'cross-funcional', capaces de generar un incremento terminado de principio a fin, sin otras dependencias externas.

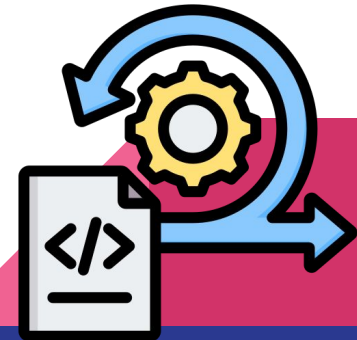


# Tres artefactos

La Pila de Producto (**Product BackLog**), que es una lista de todas las cosas “Por Realizar” del proyecto. Esta lista es confeccionada por el Dueño del Producto de acuerdo a los requerimientos y esta ordenada por la prioridad que tiene cada elemento en la pila. Es decir, de mayor a menor importancia.

La Pila del Sprint (**Sprint BackLog**), son las actividades que se van a realizar dentro de un sprint.

El Grafico de Trabajo Pendiente (**Burndown Chart**). Representa visualmente el trabajo que está por hacer versus el tiempo restante del proyecto. El trabajo pendiente se representa en el eje vertical y el tiempo en el eje horizontal. Es útil para predecir en que tiempo se terminaría todo el trabajo, incluso se puede establecer el ritmo de avance del proyecto del equipo.

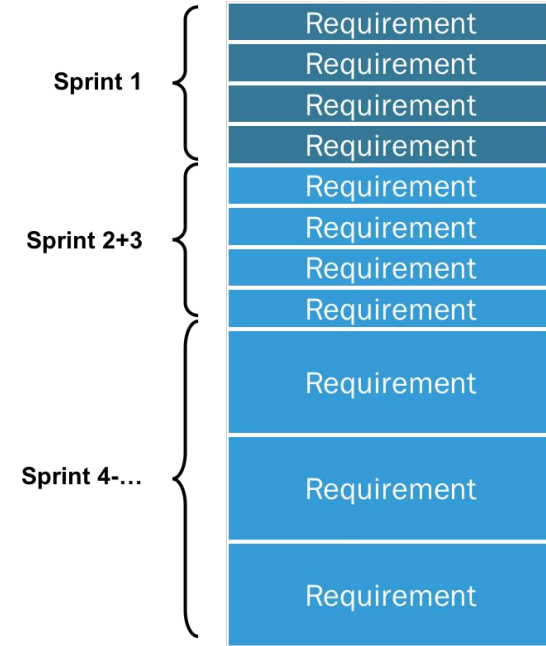


# Product Backlog

El Product Backlog (o pila de producto) es un listado de todas las tareas que se pretenden hacer durante el desarrollo de un proyecto.

Algunos product backlog pueden asociarse con proyectos de varios años, incluso.

Todas las tareas deben listarse en el product backlog, para que estén visibles ante todo el equipo y se pueda tener una visión panorámica de todo lo que se espera realizar. Si el proyecto o producto amerita una lista larga, podemos tener cientos o miles de actividades en el product backlog. Se asignan prioridades sobre el product backlog, en base a las necesidades del cliente y la complejidad de cada actividad. Así mismo, de forma sucesiva, cada sprint produce detalles adicionales, en el diseño, en la funcionalidad, y la verificación de actividades.



Más detalle

Historia de usuario (4-8 días/programador)

RELEASE ACTUAL  
PRÓXIMO SPRINT

Epic (2-4 semanas/equipo)

RELEASE SIGUIENTE

FUTURAS RELEASES

Temas

Menos detalle



# Sprint Backlog

El sprint backlog es básicamente una lista de tareas identificadas por el scrum team; ésta deberá ser completada durante cada sprint.

El sprint backlog es representado a través de un tablero de tareas; hace visible todo el trabajo necesario para alcanzar el compromiso que se hizo con el product owner para el sprint. Permite ver las tareas donde el equipo está teniendo problemas y no avanza, para tomar decisiones al respecto.

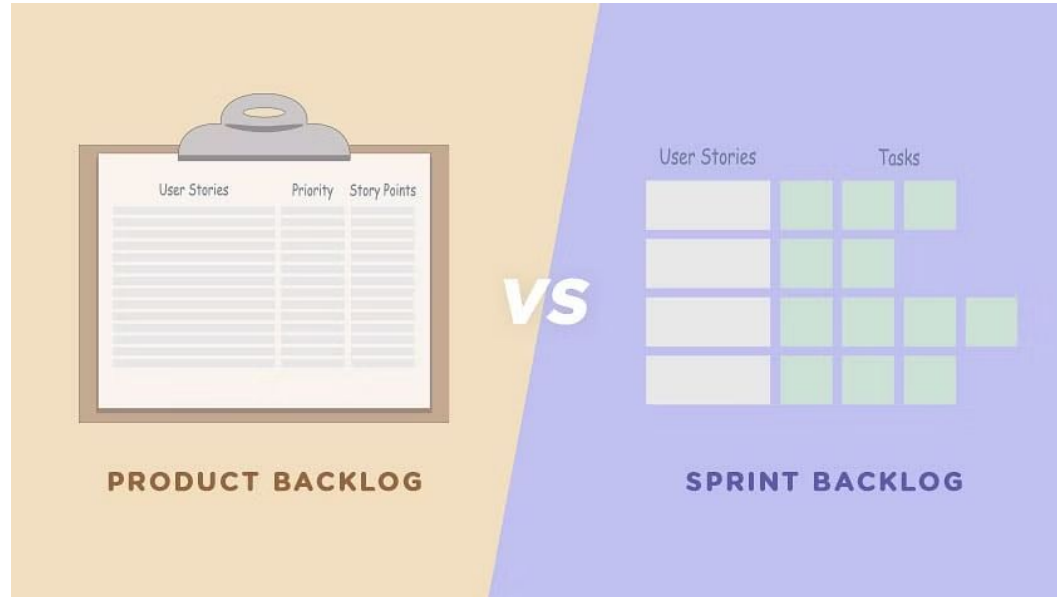
Para cada uno de los objetivos/requisitos del proyecto se muestran las tareas a cubrir; el esfuerzo pendiente para finalizarlas y la auto asignación que han hecho los miembros del equipo, también son visibles.





# Sprint BackLog vs Product BackLog

La diferencia principal entre Sprint y Product BackLog, es que el product BackLog es una parte generalizada, mientras que el sprint BackLog es específico y más orientado a detalle. Ambos backlogs requieren una gestión estratégica, y diferente entre ellos.



Objetivo del Sprint	Pendiente	En progreso	Completo
<i>Habilitar todas las partes esenciales de la tienda online para permitir que los usuarios puedan experimentar un proceso de compra completo. Esto hará que otras características de la página web sean más significativas.</i>			<div>Item #1</div> <div>t.1.6 t.1.1 t.1.3 t.1.5 t.1.2</div>
	<div>Item #2</div> <div>t.2.7</div>	<div>t.2.6 t.2.5</div>	<div>t.2.1 t.2.3 t.2.2 t.2.4</div>
	<div>Item #3</div> <div>t.3.4 t.3.5 t.3.3 t.3.2</div>	<div>t.3.1</div>	
	<div>Item #4</div> <div>t.4.4 t.4.2 t.4.5</div>		
	<div>Item #5</div>		

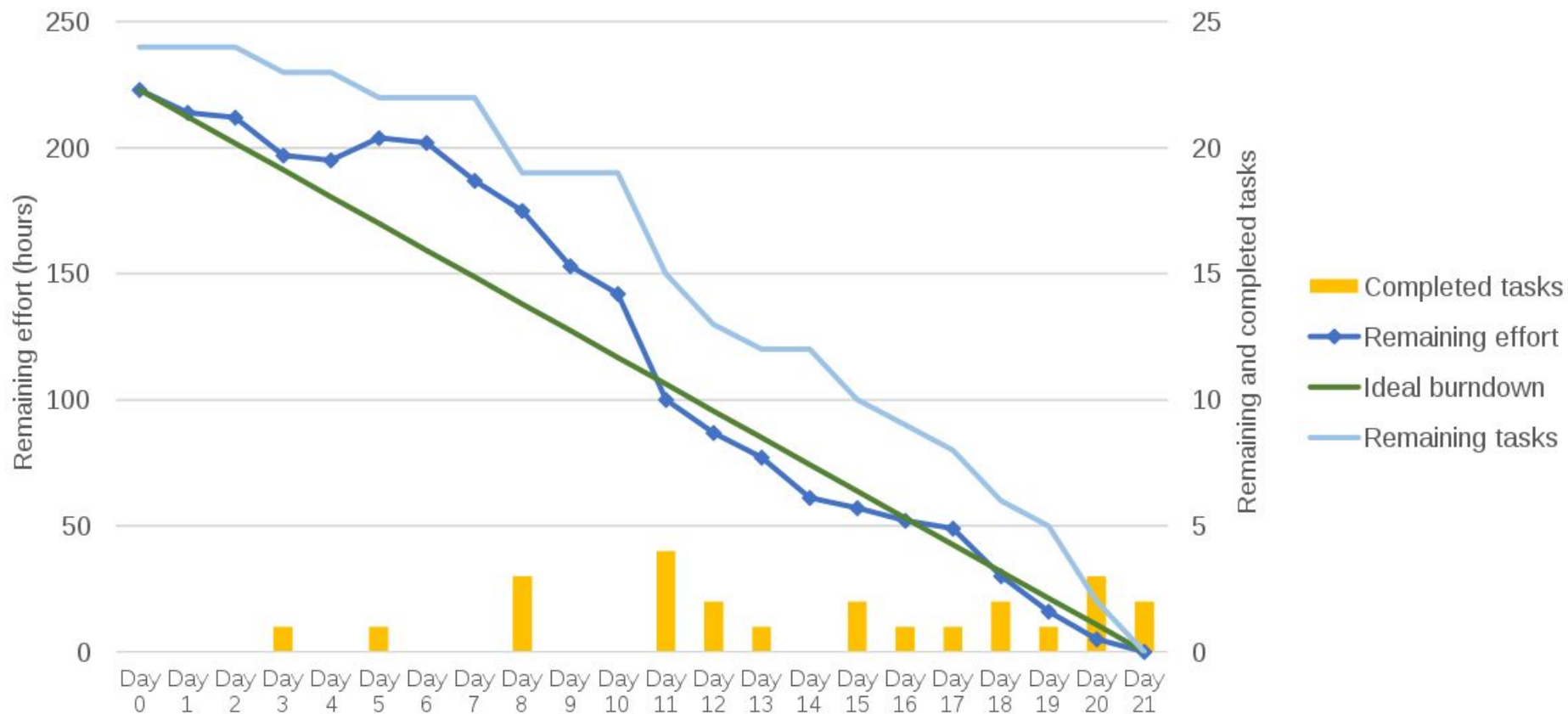
# Burndown Chart

Un Burndown Chart, diagrama de quemado o gráfica de trabajo pendiente es una gráfica en la que podemos ver el estado del progreso de un Sprint.

Un burndown chart puede ofrecernos mucha más información que el avance de un Sprint si sabemos interpretarla correctamente y podremos ayudar mucho más a los equipos gracias a esto.



# Sample Burndown Chart




# Construcción de un Burndown Chart

## 1. Eje X: Tiempo

En el eje X de nuestro burndown representamos el tiempo. Pero un tiempo finito. En el caso de un Sprint, cuando  $x=0$  debe ser el inicio del Sprint y el último valor de nuestro eje X, (en el ejemplo de la diapositiva pasada, 21 días), debe ser el final del Sprint.

## 2. Eje Y: Cantidad de trabajo

En el eje Y de la gráfica reflejamos la cantidad de trabajo. Es el trabajo que hay comprometido para realizar en el tiempo estipulado. En el caso de Scrum, sería el Sprint Backlog. Para poder crearla gráfica es necesario que las tareas estén estimadas (da igual que sean puntos de historia, jornadas, horas, etc). De esta forma podremos pintar nuestra línea guía.



# Construcción de un Burndown Chart

## 3. Línea guía: Referencia

Llamamos línea guía a la diagonal que une el último valor de nuestro eje Y (el máximo de trabajo comprometido) con el último valor del eje X (la que dice “Ideal Burndown”). Esta será nuestra referencia para saber si vamos bien, vamos mejor que bien, o si está todo mal.

## 4. Línea de progreso

Arrancaremos una nueva línea que irá actualizándose con cada tarea (que, recordemos, debe estar estimada) que se termine.

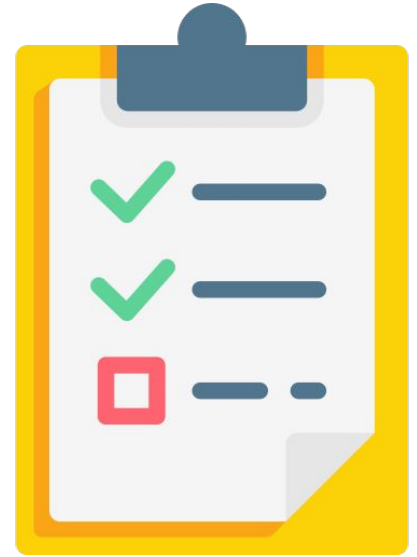
Por cada tarea terminada, bajaremos la cantidad proporcional a la estimación de la tarea terminada, y, en una situación ideal, tendremos un burndown chart en el que las tareas pendientes y el tiempo estimado “bajan” de manera ideal.



# El proceso

En Scrum un proyecto se ejecuta en bloques temporales cortos y fijos (iteraciones de un mes natural y hasta de dos semanas, si así se necesita). Cada iteración tiene que proporcionar un resultado completo, un incremento de producto final que sea susceptible de ser entregado con el mínimo esfuerzo al cliente cuando lo solicite.

El proceso parte de la lista de objetivos/requisitos priorizada del producto, que actúa como plan del proyecto. En esta lista el cliente prioriza los objetivos balanceando el valor que le aportan respecto a su coste y quedan repartidos en iteraciones y entregas.



# Planificación de la iteración (Sprint Planning)

El **primer día** de la iteración se realiza la reunión de planificación de la iteración. Tiene dos partes:

**Selección de requisitos** (4 horas máximo). El cliente presenta al equipo la lista de requisitos priorizada del producto o proyecto. El equipo pregunta al cliente las dudas que surgen y selecciona los requisitos más prioritarios que se compromete a completar en la iteración, de manera que puedan ser entregados si el cliente lo solicita.

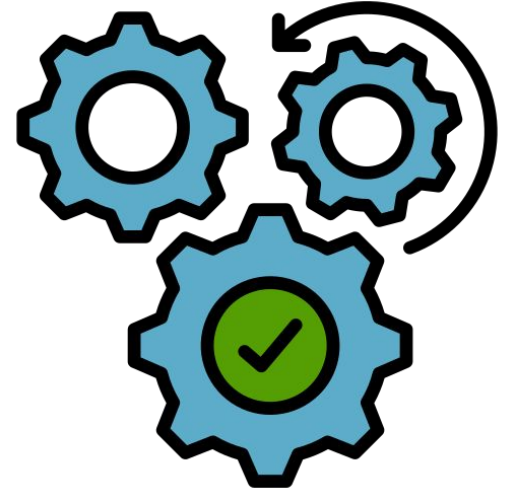
**Planificación de la iteración** (4 horas máximo). El equipo elabora la lista de tareas de la iteración necesarias para desarrollar los requisitos a que se ha comprometido. La estimación de esfuerzo se hace de manera conjunta y los miembros del equipo se autoasignan las tareas.





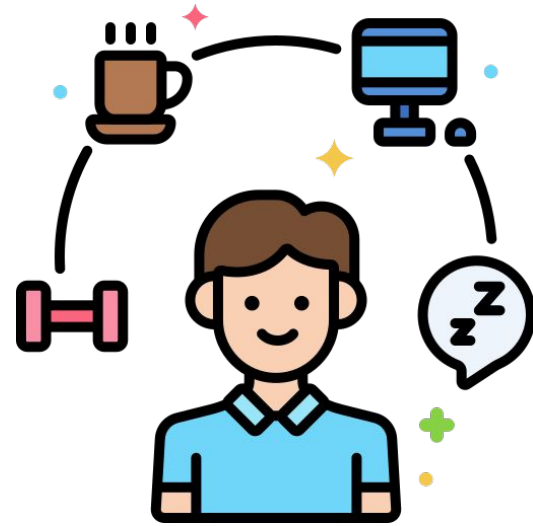
# Ejecución de la iteración (Scrum/Sprint Planning)

Planeación del Sprint (**Scrum Planning**), con la ayuda del Product Owner y el Scrum Master y el equipo, se compromete con las actividades que se deben realizar de la Pila del Producto. Es decir se seleccionan actividades prioritarias y relacionadas y con la restricción del tiempo existente, para que estas formen parte de la siguiente iteración (Sprint).



# Ejecución de la iteración (Daily Scrum)

Reunión Diaria de Scrum (**Daily Scrum**), esta actividad se lleva a cabo todos los días y debe durar quince minutos o menos (si bien suele durar alrededor de 40 minutos en la vida real). Cada miembro del equipo debe comunicar al resto del equipo y al Scrum Master, tres ítems indicados en la Tabla de Tareas: lo que se realizó, lo que se va realizar el día de hoy, y si existe algún impedimento para llevar a cabo el trabajo. Cada tarea debe evolucionar desde Por hacer (**To Do**), En Progreso (**In Progress**), Realizadas (**Done**). Al finalizar el sprint todas las tareas deben estar realizadas. Esta reunión es muy importante porque acá los miembros comparten información, para determinar la mejor manera de cumplir sus tareas y poder terminar con las metas del Sprint. Se manifiestan los riesgos encontrados y es útil porque permite que los miembros del equipo sean más eficaces.



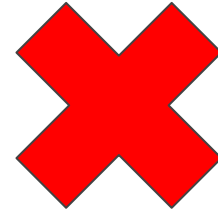
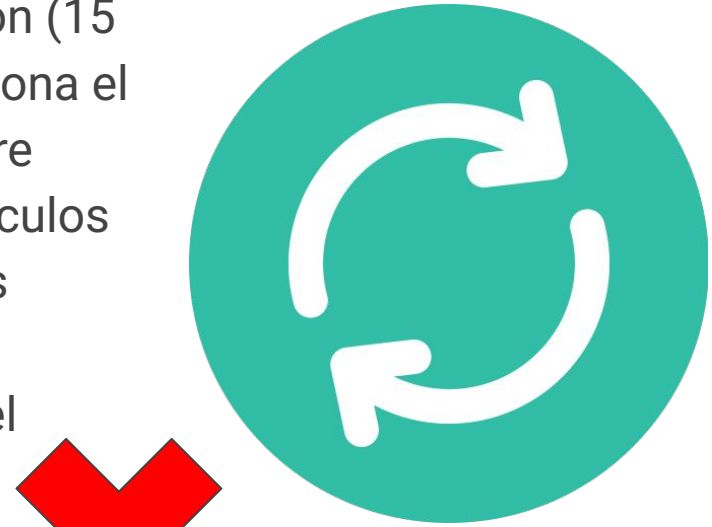
# Ejecución de la iteración (Sync meeting)

Cada día el equipo realiza una reunión de sincronización (15 minutos máximos). Cada miembro del equipo inspecciona el trabajo que el resto está realizando (dependencias entre tareas, progreso hacia el objetivo de la iteración, obstáculos que pueden impedir este objetivo) para poder hacer las adaptaciones necesarias que permitan cumplir con el compromiso adquirido. En la reunión cada miembro del equipo responde a tres preguntas:

¿Qué hice desde la última reunión de sincronización?

¿Qué voy a hacer a partir de este momento?

¿Qué impedimentos tengo o voy a tener?

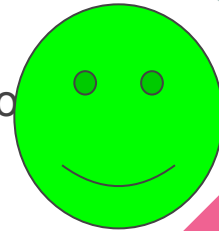


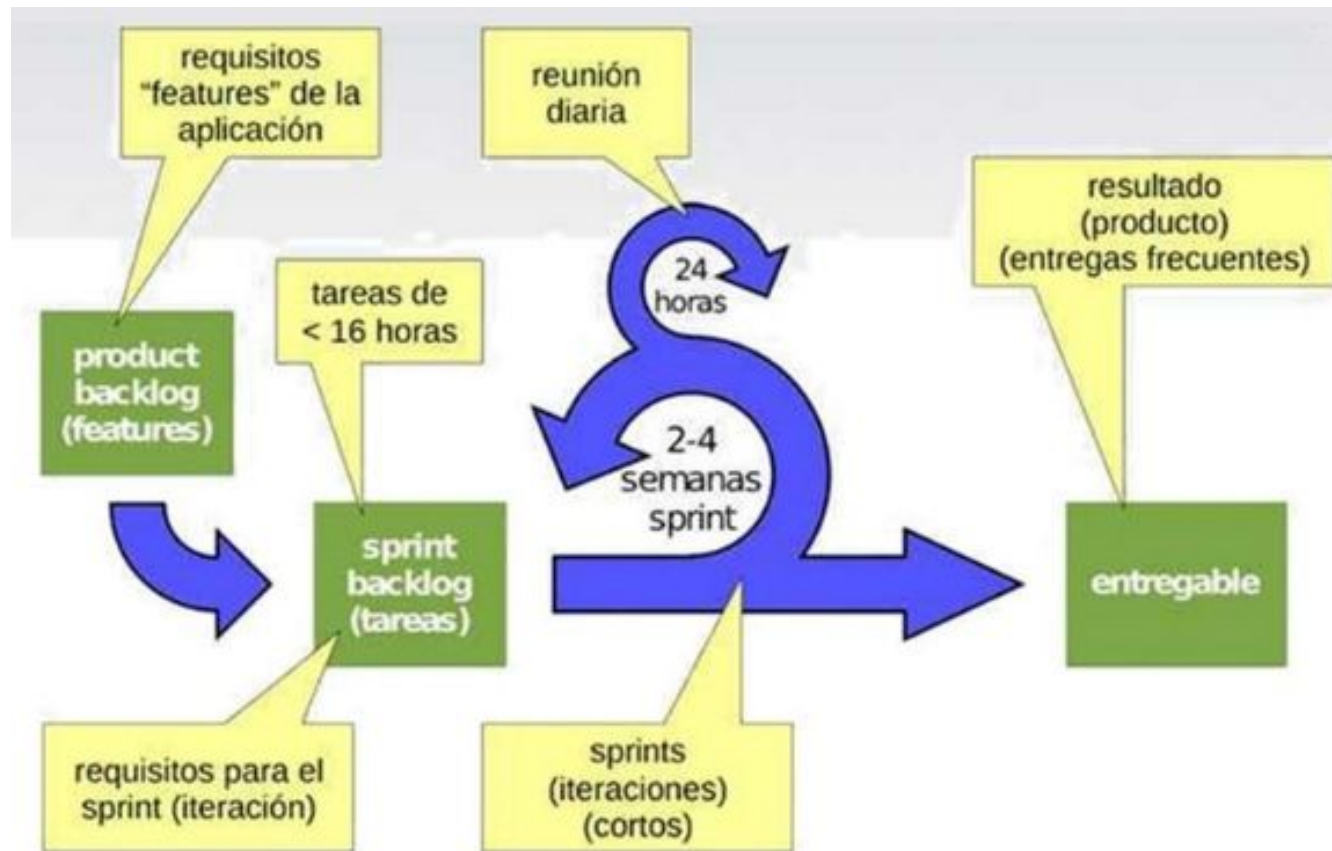
# Ejecución de la iteración (Sync meeting)

La sync meeting siempre ocurre el **último día de la iteración**, y tiene dos partes:

**Demostración** (4 horas máximo). El equipo presenta al cliente los requisitos completados en la iteración, en forma de incremento de producto preparado para ser entregado con el mínimo esfuerzo. En función de los resultados mostrados y de los cambios que haya habido en el contexto del proyecto, el cliente realiza las adaptaciones necesarias de manera objetiva, ya desde la primera iteración, replanificando el proyecto.

**Retrospectiva** (4 horas máximo). El equipo analiza cómo ha sido su manera de trabajar y cuáles son los problemas que podrían impedirle progresar adecuadamente, mejorando de manera continua su productividad. El Facilitador se encargará de ir eliminando los obstáculos identificados.

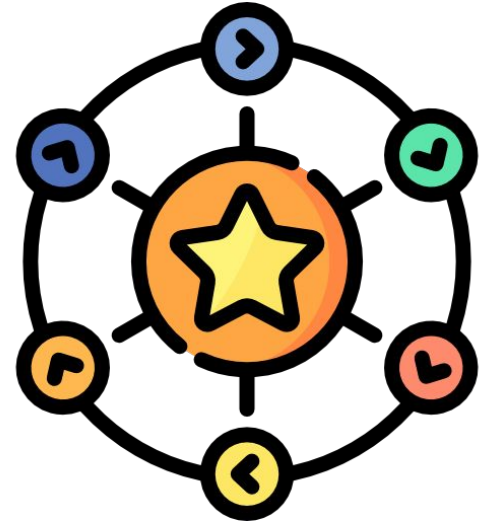




# Ventajas de scrum

## Ventajas:

- El cliente puede comenzar a utilizar el producto rápidamente.
- El cliente puede decidir los nuevos objetivos a realizar.
- Se agiliza el proceso, porque se divide el problema en pequeñas tareas.
- Menos probabilidad de que se den sorpresas o desarrollos inesperados porque el cliente va viendo poco a poco lo que se está desarrollando.



# El cliente puede comenzar a utilizar el producto rápidamente

La ventaja de que el cliente pueda comenzar a utilizar el producto rápidamente en el modelo Scrum se debe a la entrega incremental y frecuente de funcionalidades durante cada sprint. En lugar de esperar hasta que todo el producto esté completo, el equipo de desarrollo trabaja en iteraciones cortas para entregar partes funcionales del producto en intervalos regulares. Aunque el producto aún no esté completamente desarrollado, el cliente puede recibir y comenzar a utilizar las funcionalidades entregadas al final de cada sprint. Este enfoque de entrega incremental permite que el cliente experimente y evalúe el producto en etapas tempranas del desarrollo.



# El cliente puede decidir los nuevos objetivos a realizar

Durante las reuniones de Scrum planning, el cliente tiene la oportunidad de priorizar los elementos del backlog del producto y definir los objetivos específicos que le gustaría que el equipo de desarrollo logre en el próximo sprint. Esta participación activa del cliente en la definición de los objetivos permite una mayor flexibilidad en el proceso de desarrollo. A medida que se avanza en el proyecto y se obtiene una mejor comprensión de los requisitos y necesidades del cliente, es posible que surjan nuevos objetivos o cambios en los existentes. En lugar de seguir un plan rígido y predefinido, el modelo Scrum permite que los objetivos sean reevaluados y ajustados en cada sprint.

A su vez, al permitir que el cliente defina los nuevos objetivos, se fomenta una mayor satisfacción del cliente y se maximiza el valor entregado por el producto final.

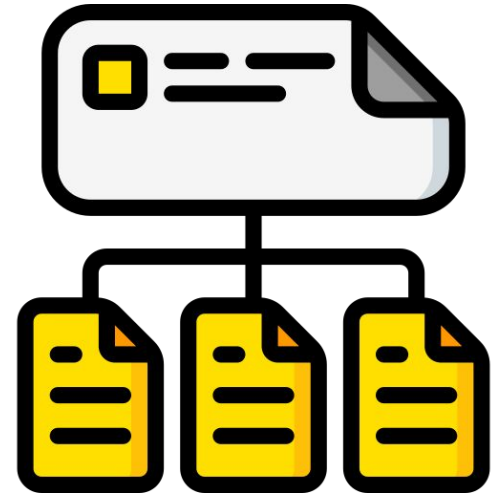




# Se agiliza el proceso, porque se divide el problema en pequeñas tareas.

Recordemos que el equipo de desarrollo divide el problema o el proyecto en pequeñas tareas o funcionalidades que pueden ser abordadas de manera individual. Estas tareas se registran en el backlog del producto y se priorizan según su importancia. Durante cada sprint, el equipo selecciona un conjunto de tareas del backlog para trabajar en ellas.

Al dividir el problema en tareas más pequeñas, se logran varios beneficios. En primer lugar, las tareas individuales son más fáciles de comprender y abordar por parte del equipo de desarrollo. Esto permite una mayor eficiencia y reduce el riesgo de errores o malentendidos. Cada tarea se puede asignar a un miembro del equipo, lo que permite una distribución equitativa de la carga de trabajo y una mayor especialización en la resolución de problemas específicos.



# Menos probabilidad de que se den sorpresas o desarrollos inesperados porque el cliente va viendo poco a poco lo que se está desarrollando.

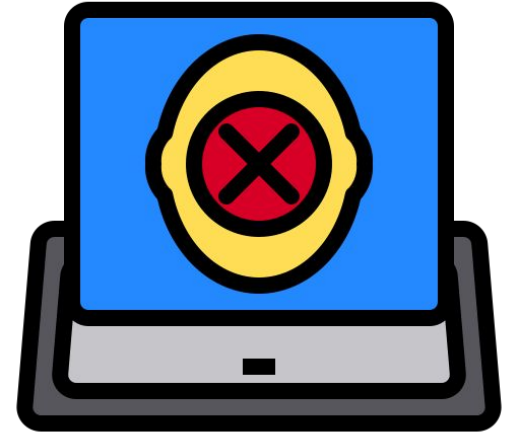
Al mostrar al cliente el producto en desarrollo de manera incremental, se minimiza la posibilidad de sorpresas o desarrollos inesperados en etapas posteriores del proyecto. El cliente tiene la oportunidad de revisar y proporcionar retroalimentación sobre el producto en cada entrega y asegurarse de que se esté alineando con sus expectativas y requisitos. También existe una visibilidad y transparencia por parte del cliente, a la hora de llevar a cabo un modelo scrum. Esto permite que el cliente tenga una comprensión clara y actualizada del estado del proyecto en todo momento. Esto reduce la incertidumbre y la probabilidad de que se den sorpresas desagradables más adelante, ya que el cliente tiene la oportunidad de corregir el rumbo o realizar ajustes a medida que avanza el desarrollo.



# Desventajas de scrum

## Desventajas:

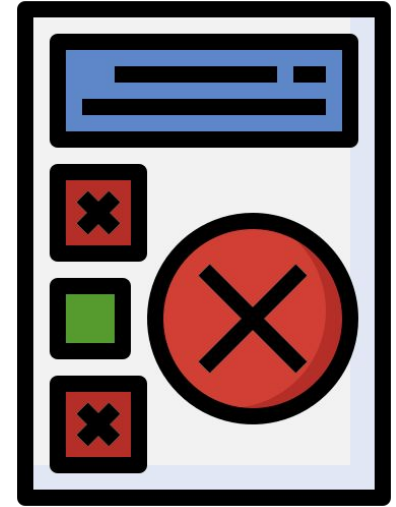
- Existe la tendencia de que se deje una tarea sin terminar y que por las exigencias del Product Owner se deban realizar otras nuevas. Estas tareas no terminadas pueden obstaculizar la planeación de nuevos Sprint y se deba volver al problema original.
- Alto nivel de estrés de los miembros del equipo. El desgaste puede ser excesivo y estresante, lo que puede disminuir el rendimiento.
- La necesidad de contar con equipos multidisciplinarios puede ser un problema, porque cada integrante del equipo debe estar en capacidad de resolver cualquier tarea y no siempre se cuenta con este perfil en la empresa.
- El equipo puede estar tentado de tomar el camino más corto para cumplir con un sprint, que no necesariamente puede ser el de mejor calidad en el desarrollo del producto.



# Tareas sin terminar

Existe la tendencia de que se deje una tarea sin terminar y que por las exigencias del Product Owner se deban realizar otras nuevas. Estas tareas no terminadas pueden obstaculizar la planeación de nuevos Sprint y se deba volver al problema original.

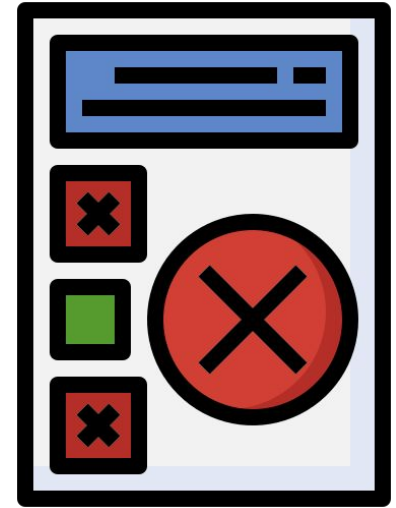
Durante el proceso de desarrollo, es posible que el Product Owner solicite cambios o agregue nuevas tareas que deben abordarse de manera prioritaria. Esta situación puede generar un desafío para el equipo de desarrollo, ya que pueden encontrarse en la posición de tener que dejar una tarea incompleta para abordar las nuevas exigencias. Si esta tendencia se repite en varios sprints, puede haber un acumulamiento de tareas incompletas y un retraso en la finalización del trabajo original.



# Tareas sin terminar

Cuando se dejan tareas sin terminar, esto puede tener un impacto negativo en la planificación de los nuevos sprints. El equipo de desarrollo puede enfrentar dificultades para estimar de manera precisa y planificar las tareas futuras si hay trabajo pendiente y no se ha completado. Esto puede llevar a una falta de visibilidad y una planificación deficiente, ya que es necesario considerar tanto las tareas pendientes como las nuevas tareas solicitadas.

Además, si no se abordan adecuadamente las tareas incompletas, puede haber una acumulación de trabajo no terminado que afecte la calidad del producto y la satisfacción del cliente. Las tareas no completadas pueden generar dependencias y afectar la funcionalidad general del producto, lo que requiere que el equipo deba volver al problema original y dedicar tiempo adicional para completar las tareas pendientes.



# ¿Cómo abordar la problemática de las tareas inconclusas?

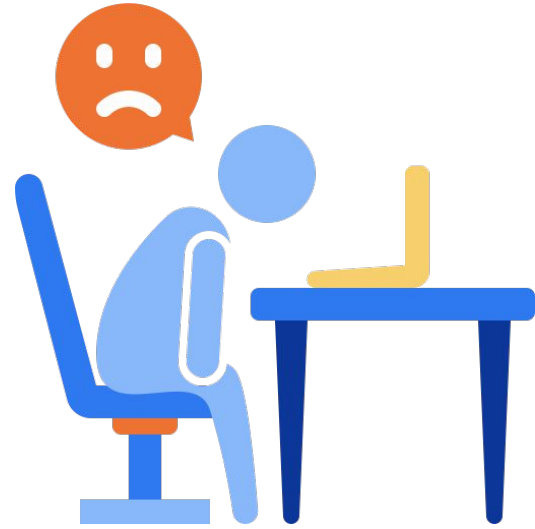
Para abordar esta desventaja, es importante tener una comunicación clara y continua entre el equipo de desarrollo y el Product Owner. Es fundamental establecer expectativas realistas y priorizar adecuadamente las tareas para evitar una sobrecarga de trabajo y una acumulación excesiva de tareas no terminadas. Además, es importante realizar una gestión eficaz del backlog y asegurarse de que las tareas sean adecuadamente refinadas y estimadas antes de comprometerse a trabajar en ellas.



# Alto nivel de estrés

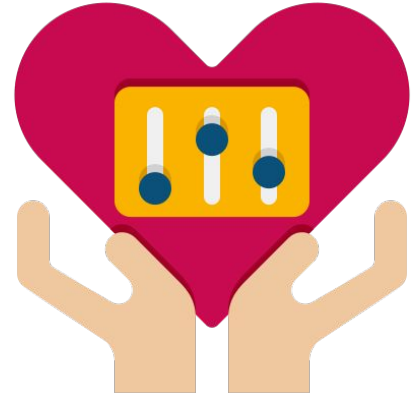
Alto nivel de estrés de los miembros del equipo. El desgaste puede ser excesivo y estresante, lo que puede disminuir el rendimiento. Esto se debe al ritmo de trabajo acelerado y las exigencias constantes del proceso. Esto puede generar un nivel de estrés elevado en los miembros del equipo. Pueden surgir situaciones en las que los plazos sean ajustados, los recursos sean limitados o los obstáculos sean difíciles de superar. Esto puede llevar a una sensación de presión constante y una carga de trabajo abrumadora.

El desgaste excesivo y el estrés prolongado pueden tener un impacto negativo en el rendimiento y la salud de los miembros del equipo. El estrés crónico puede llevar a la disminución de la productividad, la falta de concentración, la toma de decisiones deficientes y la disminución de la calidad del trabajo. Además, el alto nivel de estrés puede afectar negativamente el bienestar emocional y físico de los miembros del equipo, lo que puede llevar a un agotamiento o burnout.



# Cómo evitar el estrés en los miembros del equipo

Es importante reconocer y abordar este desafío para mantener un ambiente de trabajo saludable y productivo. Los equipos Scrum pueden implementar prácticas como la gestión efectiva de la carga de trabajo, la planificación realista, el establecimiento de límites y la promoción de un equilibrio adecuado entre el trabajo y la vida personal. Además, la comunicación abierta y el apoyo mutuo dentro del equipo son fundamentales para identificar y abordar los problemas de estrés y encontrar soluciones adecuadas.

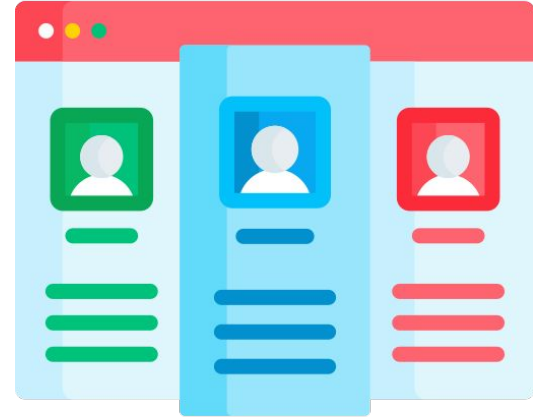




# Equipos multidisciplinarios

En el modelo Scrum, se espera que los equipos de desarrollo sean multidisciplinarios, es decir, que cuenten con miembros que posean habilidades y conocimientos diversos. Esto se debe a que las tareas o funcionalidades en el backlog del producto pueden requerir diferentes áreas de experiencia, como diseño, desarrollo de software, pruebas, implementación, entre otros.

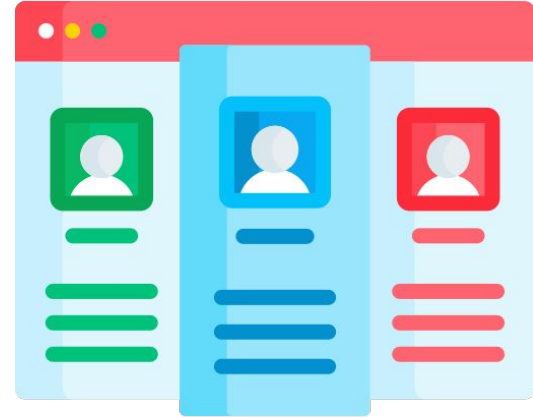
Sin embargo, puede haber situaciones en las que la empresa no cuente con miembros del equipo que posean todas las habilidades necesarias para abordar todas las tareas del proyecto. Esto puede deberse a limitaciones de recursos, falta de personal especializado o falta de diversidad en las habilidades dentro de la organización.



# ¿Cómo hacer que esto no sea una desventaja?

La falta de un equipo multidisciplinario puede generar desafíos y obstáculos en el desarrollo del proyecto. Si un miembro del equipo no tiene las habilidades necesarias para abordar una tarea específica, puede haber retrasos en su implementación o la calidad del trabajo puede verse comprometida. Además, puede generar una dependencia excesiva en ciertos miembros del equipo que poseen las habilidades requeridas, lo que puede generar desequilibrios y sobrecargas de trabajo.

Para abordar esta desventaja, es importante evaluar y planificar adecuadamente los recursos necesarios para el proyecto. Esto puede implicar la contratación de personal adicional con habilidades específicas o la capacitación y desarrollo de los miembros del equipo existentes para adquirir las habilidades requeridas. También es posible buscar apoyo externo, como consultores o especialistas, para cubrir las áreas de experiencia faltantes.



# Tomar el camino más corto

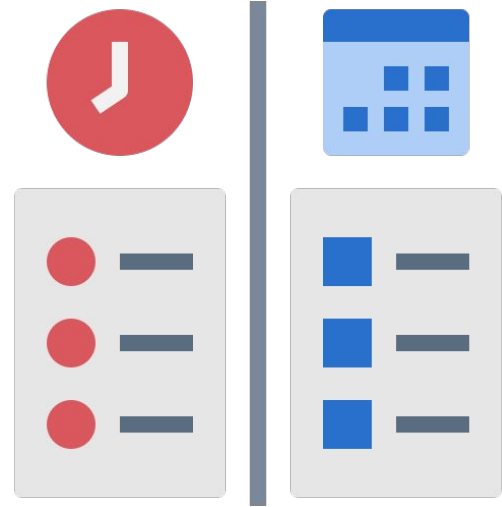
Recordemos el tema del estrés y la presión por llegar al plazo previsto. Teniendo esto en cuenta, esta presión puede llevar al equipo a tomar atajos o tomar el camino más corto para cumplir con las entregas en el tiempo asignado. Esto puede significar omitir ciertos pasos de desarrollo, como pruebas exhaustivas, revisiones de código o refinamiento del diseño. El objetivo principal se convierte en completar la tarea dentro del sprint en lugar de garantizar la calidad óptima del producto.

El problema con tomar el camino más corto es que puede afectar negativamente la calidad del producto y la satisfacción del cliente a largo plazo. Al omitir pasos importantes o realizar un desarrollo apresurado, pueden surgir problemas técnicos, errores o deficiencias en la funcionalidad. Estos problemas pueden requerir más tiempo y esfuerzo para solucionar en futuros sprints, lo que puede ralentizar el progreso general del proyecto.



# ¿Qué se debe hacer en estos casos?

Para abordar esta desventaja, es importante fomentar un enfoque equilibrado entre la entrega dentro del sprint y la calidad del producto. El equipo de desarrollo debe ser consciente de la importancia de cumplir con los plazos establecidos, pero también de la necesidad de garantizar la calidad en cada entrega. Se deben establecer estándares de calidad claros y promover prácticas de desarrollo sólidas, como pruebas exhaustivas, revisiones de código y refinamiento del diseño.



# ¿Para qué proyectos es adecuado usar scrum?

Scrum se utiliza para resolver situaciones en que **no se está entregando al cliente lo que necesita**, cuando **las entregas se alargan demasiado**, **los costes se disparan o la calidad no es aceptable**, cuando se necesita capacidad de reacción ante la competencia, cuando **la moral de los equipos es baja y la rotación alta**, cuando **es necesario identificar y solucionar ineficiencias sistemáticamente** o cuando **se quiere trabajar utilizando un proceso especializado en el desarrollo de producto**.

Vamos a analizar cada una de las frases que están resaltadas:




# “No se está entregando al cliente lo que necesita”

Scrum se utiliza cuando existe una brecha entre lo que se está entregando al cliente y sus necesidades reales. Al adoptar Scrum, se enfatiza la colaboración con el cliente a través de la participación regular y activa en el proceso de desarrollo. Esto permite una retroalimentación constante y una mayor alineación entre las necesidades del cliente y el producto final.

# “Las entregas se alargan demasiado”

Podemos también abordar el tema de las entregas prolongadas al trabajar en sprints de tiempo fijo. Estos sprints tienen una duración predeterminada, generalmente de una a cuatro semanas. Al dividir el trabajo en incrementos manejables y establecer plazos definidos, Scrum fomenta la entrega regular de funcionalidades y productos, evitando así la prolongación innecesaria de los plazos de entrega. Obvio que también causa la desventaja que fue mencionada previamente.




## "Los costes se disparan"

Scrum busca controlar los costos mediante la planificación y priorización efectiva del trabajo. Al tener ciclos de desarrollo más cortos y entregar incrementos de valor con mayor frecuencia, se pueden identificar y abordar rápidamente los riesgos y desviaciones que pueden llevar a un aumento de los costos. Además, el enfoque en la colaboración y la transparencia dentro del equipo de desarrollo y con el cliente ayuda a optimizar la eficiencia y evitar gastos innecesarios.

## "La calidad no es aceptable"

Se promueve la calidad a través de la participación activa del cliente y la retroalimentación continua. Al entregar incrementos de valor en cada sprint, se fomenta la detección temprana de problemas de calidad y se pueden realizar correcciones rápidas. Además, la planificación y ejecución de pruebas periódicas aseguran que los estándares de calidad se cumplan en cada entrega.




# "Se necesita capacidad de reacción ante la competencia"

Podemos obtener una mayor capacidad de respuesta al mercado y a la competencia gracias a scrum. Al trabajar en sprints y realizar entregas regulares, se pueden realizar ajustes y mejoras en función de las necesidades cambiantes del mercado. Esto permite adaptarse rápidamente a la competencia y mantener la relevancia del producto.

# "La moral de los equipos es baja y la rotación alta"

Scrum busca mejorar la moral y la retención del equipo al fomentar la autogestión y la colaboración, de una manera muy similar a cómo lo hacía XP (clase pasada). Al proporcionar un marco de trabajo que valora la participación activa, la toma de decisiones conjunta y el empoderamiento del equipo, se promueve un entorno de trabajo más motivador y satisfactorio.





## "Es necesario identificar y solucionar ineficiencias sistemáticamente"

Obviamente, al ser un modelo incremental, por naturaleza se fomenta la mejora continua a través de la revisión regular del proceso y la identificación de ineficiencias. A su vez, mediante la celebración de reuniones de retrospectiva al final de cada sprint, se pueden analizar y abordar las áreas de mejora, lo que permite un desarrollo más eficiente y efectivo.

## "Se quiere trabajar utilizando un proceso especializado en el desarrollo de producto"

Al final del día, scrum es un marco de trabajo ágil especializado en el desarrollo de productos complejos. Al utilizar Scrum, se aprovechan las mejores prácticas y principios específicos para gestionar eficientemente el desarrollo de productos, lo que garantiza una mayor eficacia y éxito en el resultado final, al igual que ocurre en los modelos ágiles que vimos durante la cursada.

