

Clase 5

Scrum

(Parte 2)



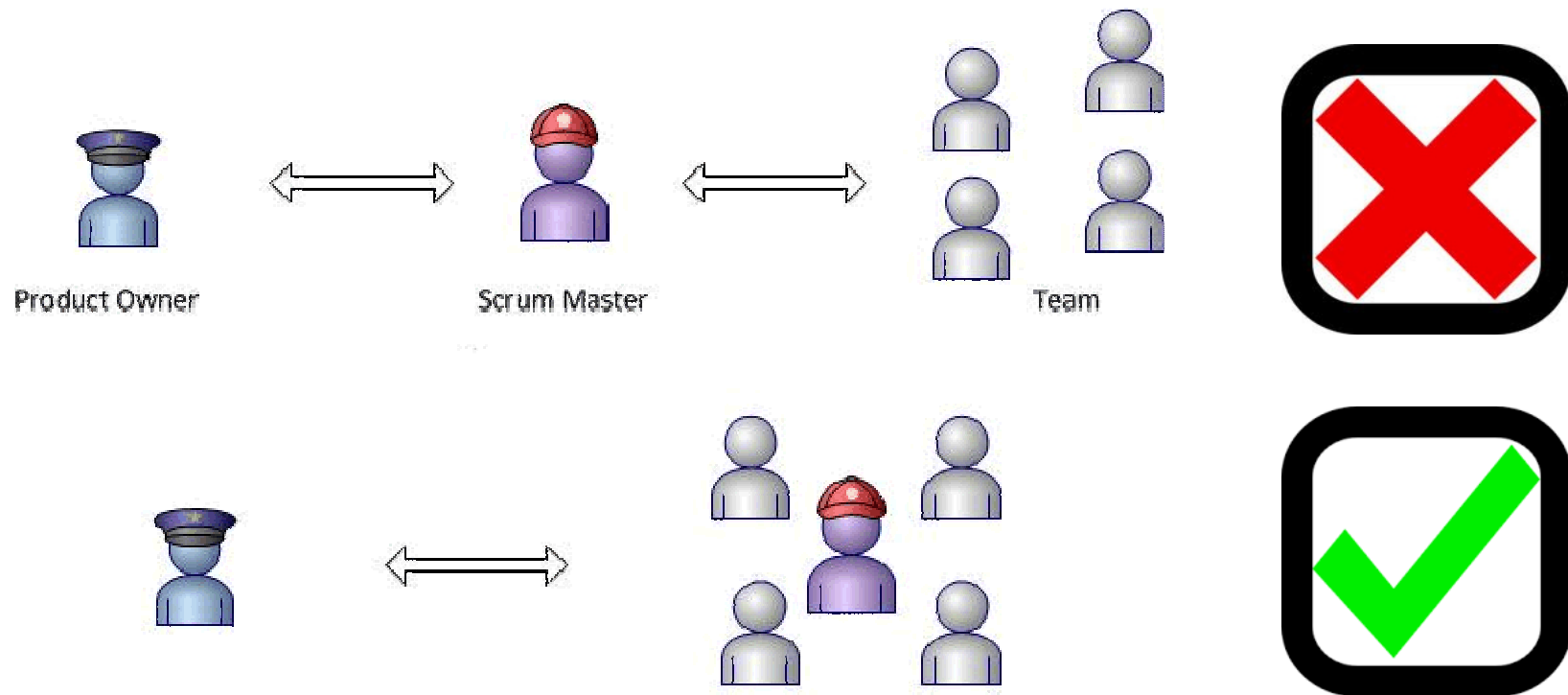
1. Roles

La metodología Scrum no tiene **jefe de proyecto**.

Las responsabilidades del tradicional jefe de proyecto se distribuyen entre los siguientes roles de Scrum:

- **Cliente** (“**Product Owner**”)
- **Facilitador** (“**Scrum Master**”)
- **Equipo** (“**Team**”)

1. Roles



1. Roles

1.1. Cliente o Product Owner

a) Representa a todas las personas interesadas en los resultados del proyecto, internas o externas a la organización, promotores del proyecto y usuarios finales.



1. Roles

1.1. Cliente o Product Owner

- b) Actúa como **interlocutor único** ante el equipo, con autoridad para tomar decisiones.
- c) Define los **objetivos** del producto o proyecto.
- d) Es el propietario de la **planificación del proyecto**: crea y mantiene la lista priorizada con los requisitos necesarios para cubrir los objetivos del proyecto, y conoce el valor que aportará cada uno.

1. Roles

1.1. Cliente o Product Owner

e) Divide la lista de requisitos estableciendo un **calendario de entregas**.

f) Antes de iniciar cada iteración **replanifica el proyecto** en función de los requisitos que aportan más valor en ese momento, de los requisitos completados en la iteración anterior y del contexto del proyecto.

1. Roles

1.1. Cliente o Product Owner

g) Participa de la reunión de planificación de la iteración, proponiendo los requisitos más prioritarios a desarrollar, respondiendo las dudas del equipo y detallando los requisitos que el equipo se compromete a hacer.



1. Roles

1.1. Cliente o Product Owner

- h) Está **disponible** durante el curso de la iteración para responder a las preguntas que puedan aparecer.
- i) **No puede cambiar los requisitos** que se están desarrollando en una iteración, una vez que esté iniciada.
- j) Participa de la reunión de demostración de la iteración, revisando los requisitos completados.

1. Roles

1.2. Facilitador o Scrum Master



- a) **Lidera** al equipo.
- b) Asegura que todos los participantes del proyecto sigan las reglas y el proceso de Scrum, ajustándolos a la cultura particular de la organización.

1. Roles

1.2. Facilitador o Scrum Master

- c) Guía al equipo en su colaboración con el cliente para maximizar la **sinergia**.
- d) Asegura que la lista de requisitos priorizada esté preparada antes de la siguiente iteración.
- e) Facilita las reuniones de planificación de la iteración, de sincronización del equipo, de demostración y de retrospectiva, de manera que sean productivas y consigan sus objetivos.

1. Roles

1.2. Facilitador o Scrum Master

f) Quita los impedimentos que el equipo encuentra en su camino para conseguir el objetivo de cada iteración.

Estos obstáculos se identifican en las reuniones de sincronización y de retrospectiva.



1. Roles

1.2. Facilitador o Scrum Master

g) Protege y aísla al equipo de interrupciones externas durante la ejecución de la iteración, como introducción de nuevos requisitos, "secuestro" no previsto de un miembro del equipo, etc.

De esta manera el equipo puede mantener su productividad y el compromiso que adquirió para terminar los requisitos de la iteración.

1. Roles

1.2. Facilitador o Scrum Master

- h) Asegura la **calidad** de los desarrollos.
- i) Enseña al equipo a **autogestionarse**. No da respuestas, sino que guía al equipo con preguntas para que descubra por sí mismo una solución.

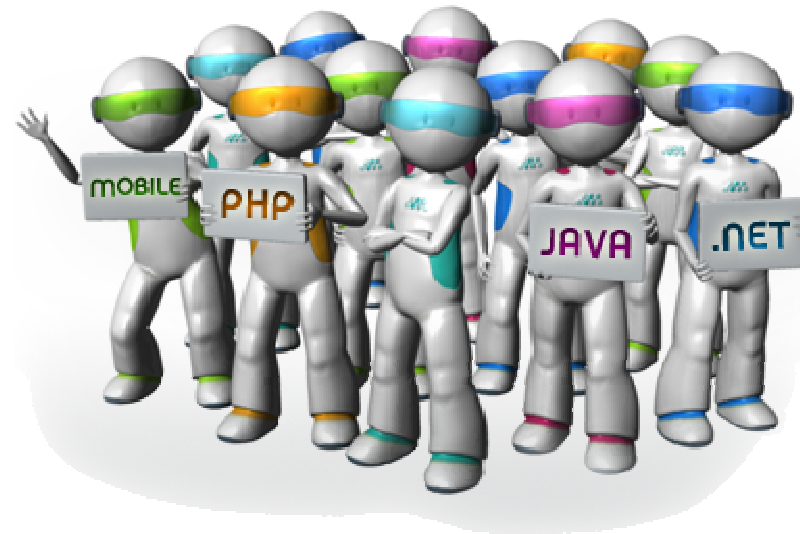


1. Roles

1.3. Equipo o Team

Grupo de personas que **de manera conjunta** desarrollan el producto del proyecto.

Comparten la responsabilidad del trabajo que realizan y su calidad, en cada iteración.



1. Roles

1.3. Equipo o Team

Está estudiado que el tamaño ideal de un equipo Scrum debe estar entre 5 y 9 personas.



1. Roles

1.3. Equipo o Team

Por encima de 9,
la comunicación y
colaboración entre
los integrantes se
hace difícil y se
forman subgrupos



1. Roles

1.3. Equipo o Team

Por debajo de 5, cualquier imprevisto o interrupción sobre un miembro del equipo compromete seriamente el compromiso adquirido.



1. Roles

1.3. Equipo o Team

- a) Es un equipo **auto gestionado**.
- b) Selecciona los requisitos que se compromete a completar en una iteración, de manera que estén preparados para ser entregados al cliente.
- c) Estima la complejidad de cada uno de los requisitos priorizados en la lista.



1. Roles

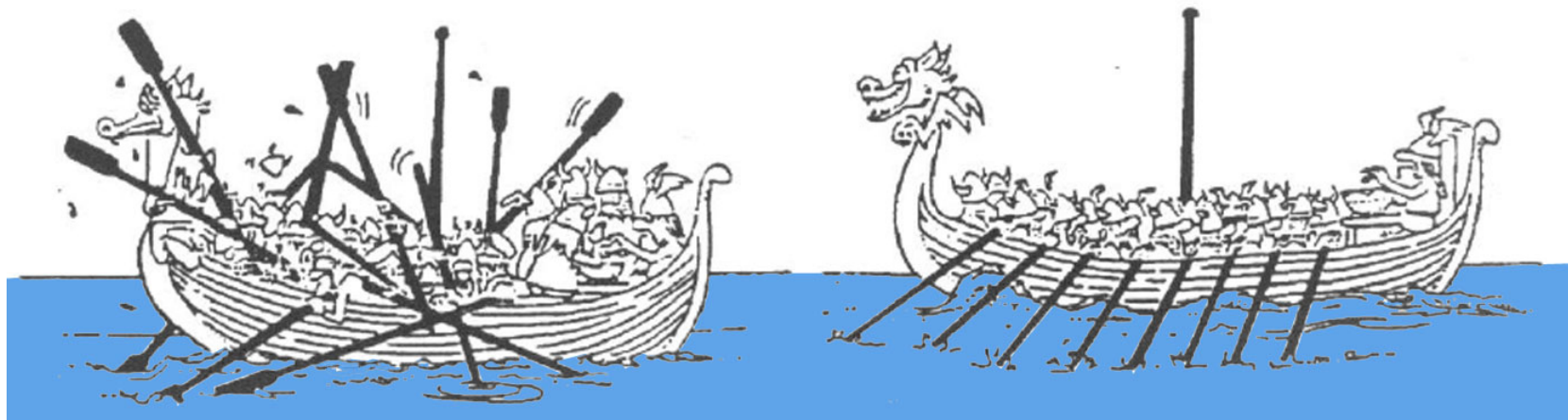
1.3. Equipo o Team

- d) En la reunión de planificación de la iteración, decide cómo va a realizar su trabajo.
- e) Selecciona los requisitos que pueden completar en cada iteración, realizando al cliente las preguntas necesarias.
- f) Identifica todas las tareas necesarias para completar cada requisito.

1. Roles

1.3. Equipo o Team

g) Durante la iteración, trabaja de manera conjunta para conseguir los objetivos fijados.



1. Roles

1.3. Equipo o Team



h) Al finalizar la iteración debe mostrarle al cliente los requisitos completados, y hacer una retrospectiva final para mejorar de forma continua su manera de trabajar.

1. Roles

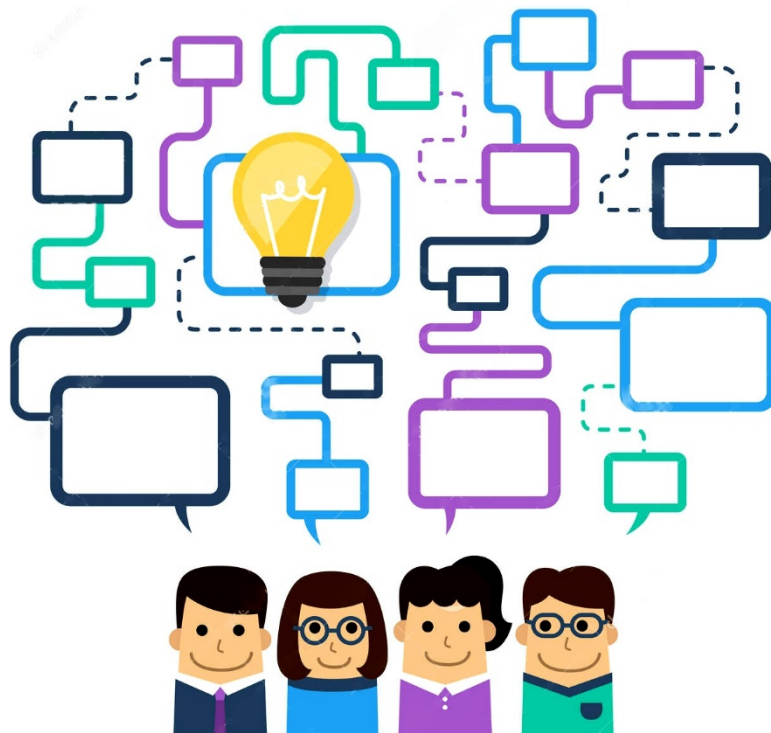
1.3. Equipo o Team

i) El equipo es multidisciplinario: los miembros tienen las habilidades necesarias para poder identificar y ejecutar todas las tareas que permiten proporcionar al cliente los requisitos comprometidos en la iteración.



1. Roles

1.3. Equipo o Team



j) Colaboración creativa:
se crea una sinergia que
permite que el resultado
sea más rico al nutrirse
de las diferentes
experiencias,
conocimientos y
habilidades de todos.

1. Roles

1.3. Equipo o Team

k) Los miembros del equipo deben dedicarse al proyecto a tiempo completo para evitar:

- dañar su productividad por cambios de tareas en diferentes proyectos;
- interrupciones externas;

y así poder mantener el compromiso que adquieren en cada iteración.

1. Roles

1.3. Equipo o Team

I) Todos los miembros del equipo trabajan en la misma localización física, para poder maximizar la comunicación.

Así se minimizan otros canales de comunicación menos eficientes que hacen que las tareas se transformen en “pasar la pelota” o que hacen perder el tiempo en el establecimiento de la comunicación.

1. Roles

1.3. Equipo o Team

m) El equipo debe ser estable durante el proyecto: sus miembros deben cambiar lo mínimo posible para poder aprovechar lo que les costó construir su organización para el trabajo y sus relaciones interpersonales.

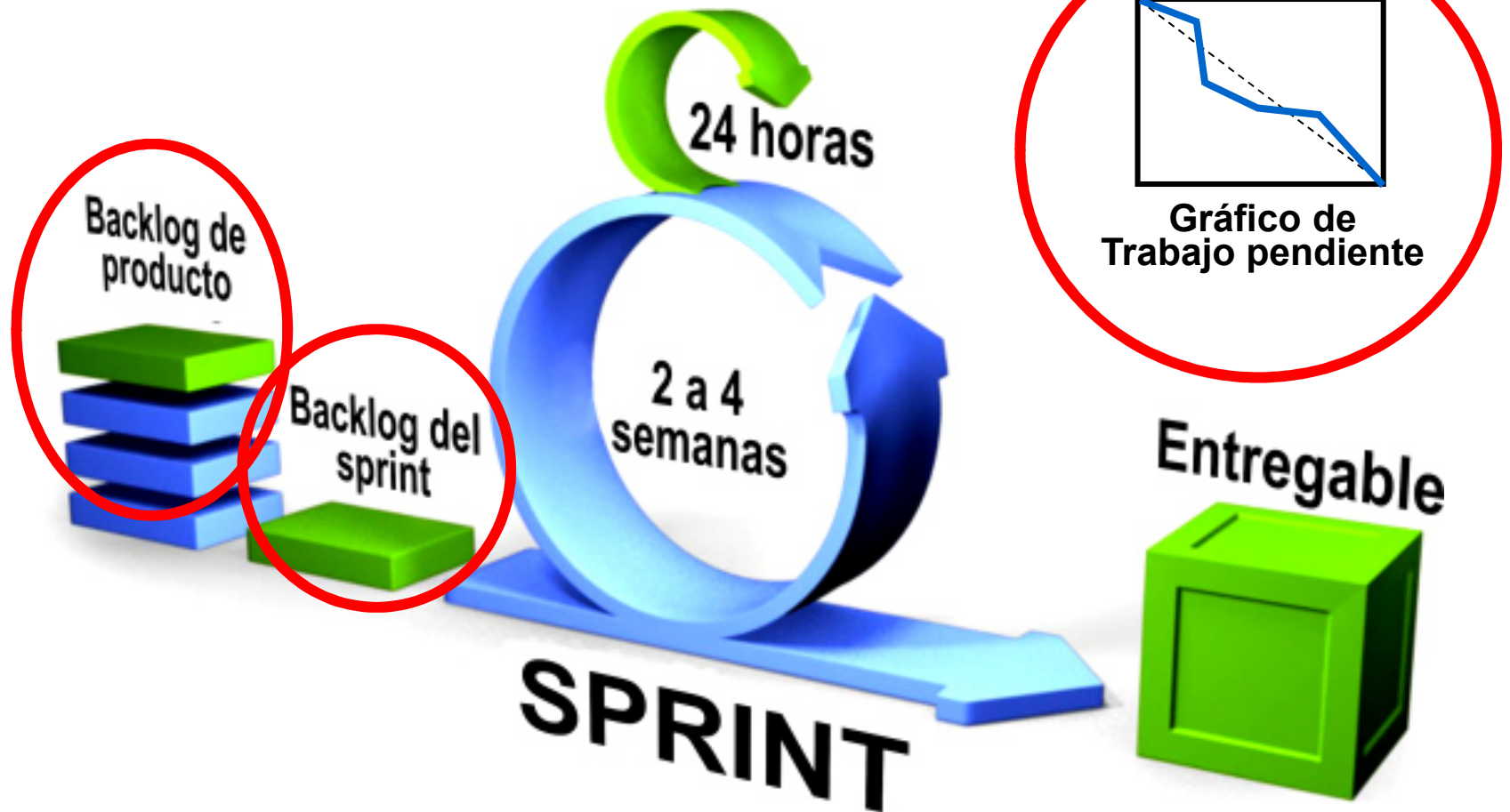


2. Herramientas

Principales herramientas que propone Scrum:

- Lista de requisitos priorizada (*Product Backlog*)
- Lista de tareas de la iteración (*Sprint Backlog*)
- Gráfico de trabajo pendiente (*Burndown*)

2. Herramientas



2. Herramientas

2.1. Product Backlog

Representa las expectativas del cliente respecto a los objetivos y entregas del proyecto.

El **cliente** es el responsable de crear y gestionar esta lista con la ayuda del **facilitador** y del **equipo**, quien proporciona el costo estimado de completar cada requisito.



2. Herramientas

2.1. Product Backlog

Para cada requisito se indica el valor que aporta al cliente y el costo estimado de completarlo.

La lista está priorizada balanceando ambos valores.



2. Herramientas

2.1. Product Backlog

Se indican los puntos en los cuales el cliente desea que se le entreguen los requisitos completados hasta ese momento.

Se planifica en función de la velocidad de desarrollo de los equipos.

Se consideran los riesgos del proyecto y se incluyen las tareas necesarias para mitigarlos.

2. Herramientas

2.1. Product Backlog

Antes de iniciar la primera iteración, el cliente debe tener definida la meta del proyecto y creada la lista de requisitos.



2. Herramientas

2.1. Product Backlog

No es necesario que la lista esté completa ni que todos los requisitos tengan gran detalle. Basta con que los más prioritarios estén identificados y con suficiente detalle como para que el equipo comience a trabajar. Los requisitos de iteraciones futuras pueden ser mucho más amplios y generales.

2. Herramientas

2.1. Product Backlog

La incertidumbre y complejidad propias de un proyecto hacen conveniente no detallar todos los requisitos hasta que esté por comenzar su desarrollo.

Así el esfuerzo de recoger, detallar y desarrollar los requisitos menos prioritarios se reparte en el período de ejecución del proyecto.

2. Herramientas

2.1. Product Backlog

Esto produce varias ventajas:

a) Se evita caer en una parálisis al inicio del proyecto, de manera que se pueda iniciar antes el desarrollo y el cliente pueda empezar a obtener resultados útiles más rápido.



2. Herramientas

2.1. Product Backlog

b) Se evita analizar en detalle requisitos no prioritarios que podrían cambiar durante el transcurso del proyecto, dado que se conocerá mejor cuál ha de ser el resultado a conseguir, o bien porque podrían ser reemplazados por otros.

2. Herramientas

2.1. Product Backlog

c) Se puede llegar a un punto del proyecto en el que no valga la pena analizar ni desarrollar los requisitos restantes.



2. Herramientas

2.1. Product Backlog

El cliente y el equipo tienen que acordar bajo qué condiciones un requisito se considerará “**completado**”

Es decir, qué tiene que hacer el equipo para que el cliente considere que el producto está preparado para ser entregado al finalizar cada iteración.

2. Herramientas

2.2. Sprint Backlog

Lista de tareas que el equipo elabora como plan para completar los requisitos seleccionados para la iteración y que se compromete a demostrar al cliente al finalizar la iteración, en forma de incremento de producto preparado para ser entregado.



2. Herramientas

2.2. Sprint Backlog



Permite identificar las tareas donde el equipo está teniendo problemas y no avanza, para poder tomar decisiones al respecto.

2. Herramientas

2.2. Sprint Backlog

Contiene las tareas, el esfuerzo pendiente para finalizarlas y la auto-asignación que han hecho los miembros del equipo.

El progreso de la iteración y su velocidad con respecto a tareas y horas pendientes se muestra mediante un gráfico de trabajo pendiente llamado “**burndown**”.

2. Herramientas

2.3. Gráfico Burndown

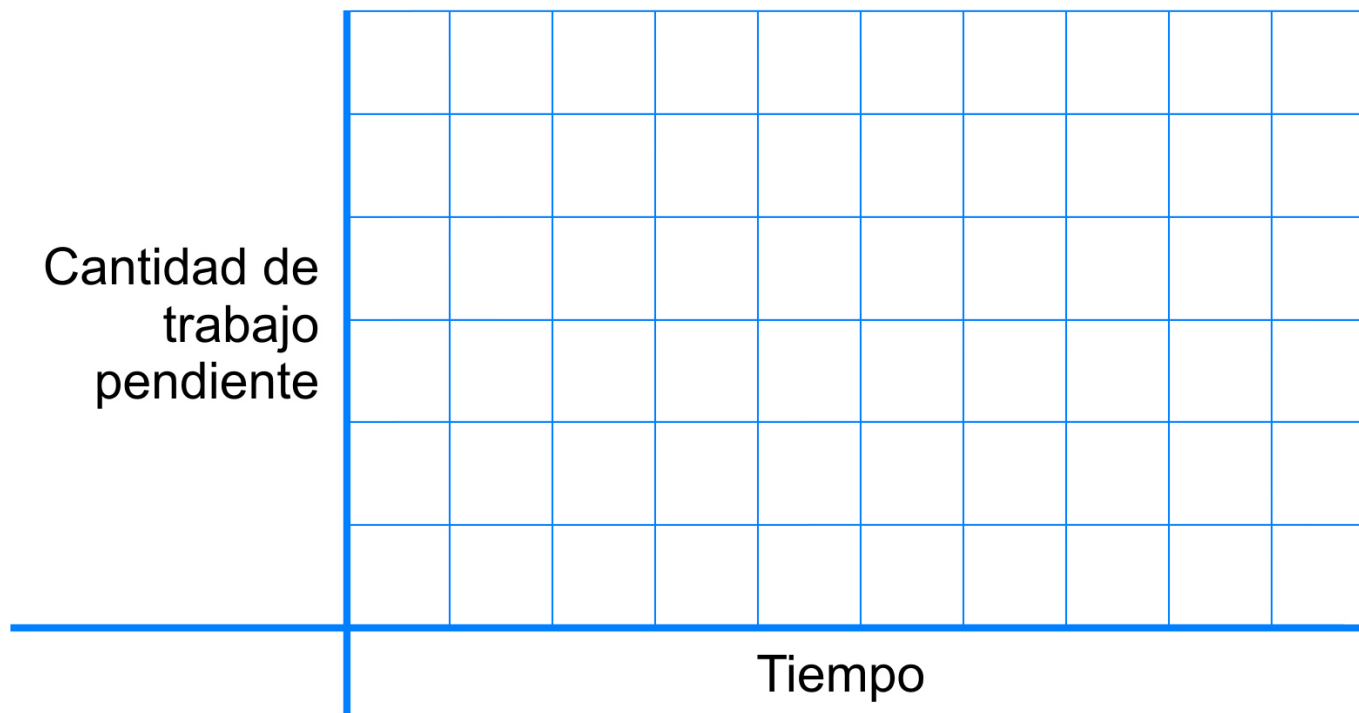
Muestra la velocidad a la que se están completando los requisitos.

Permite predecir si el equipo podrá completar el trabajo en el tiempo estimado o no.



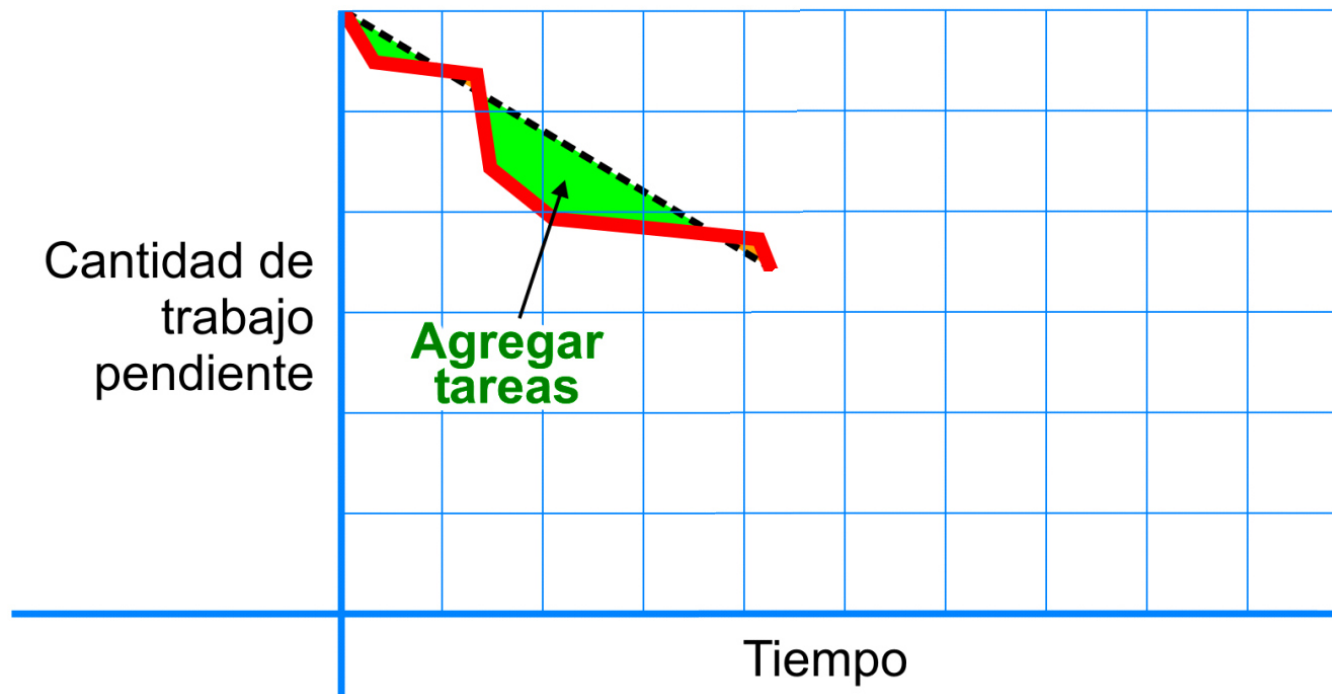
2. Herramientas

2.3. Gráfico Burndown



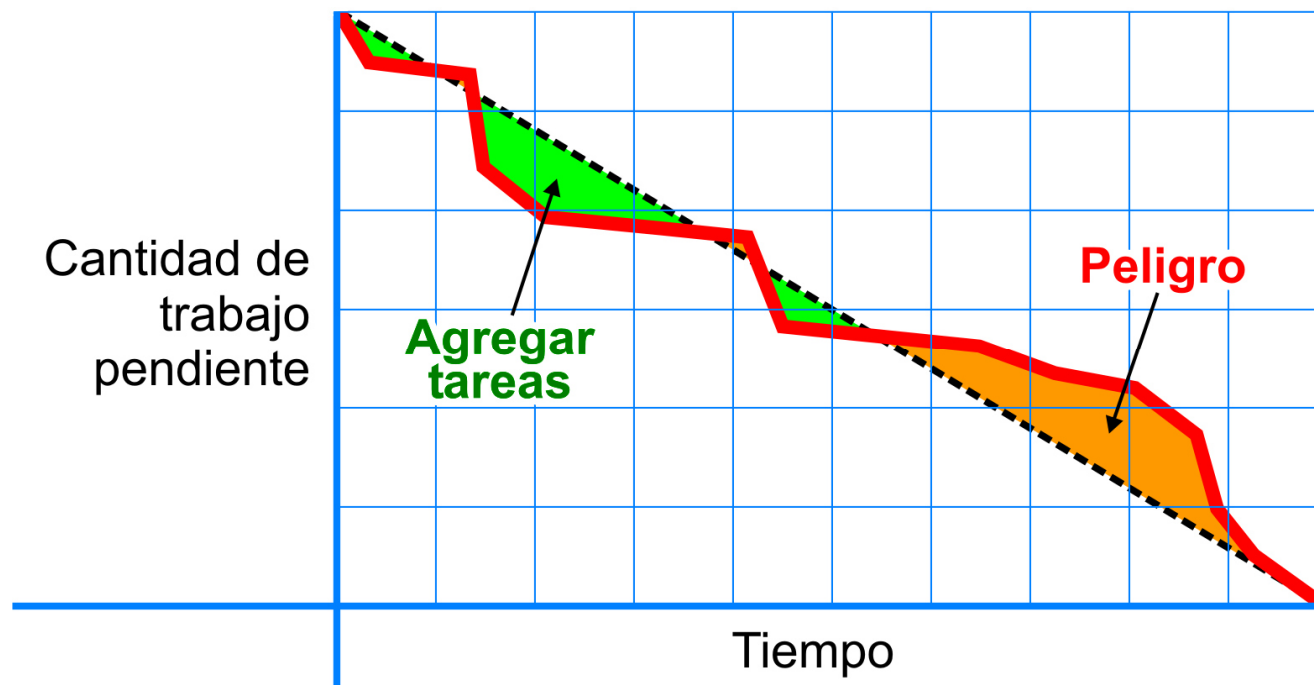
2. Herramientas

2.3. Gráfico Burndown



2. Herramientas

2.3. Gráfico Burndown



2. Herramientas

2.3. Gráfico Burndown

En un simple vistazo muestra la evolución del equipo con respecto a los requisitos del usuario:

- Cuanto trabajo ya se hizo;
- Cuanto trabajo queda por hacer;
- Velocidad del equipo;
- Fecha de finalización esperada.

2. Herramientas

2.3. Gráfico Burndown

Permite realizar diversas simulaciones, por ejemplo:

- Ver qué ocurriría con las fechas de entrega si se agregan requisitos;
- Ver cómo se avanzaría si se quitaran requisitos o si se agregaran equipos.