

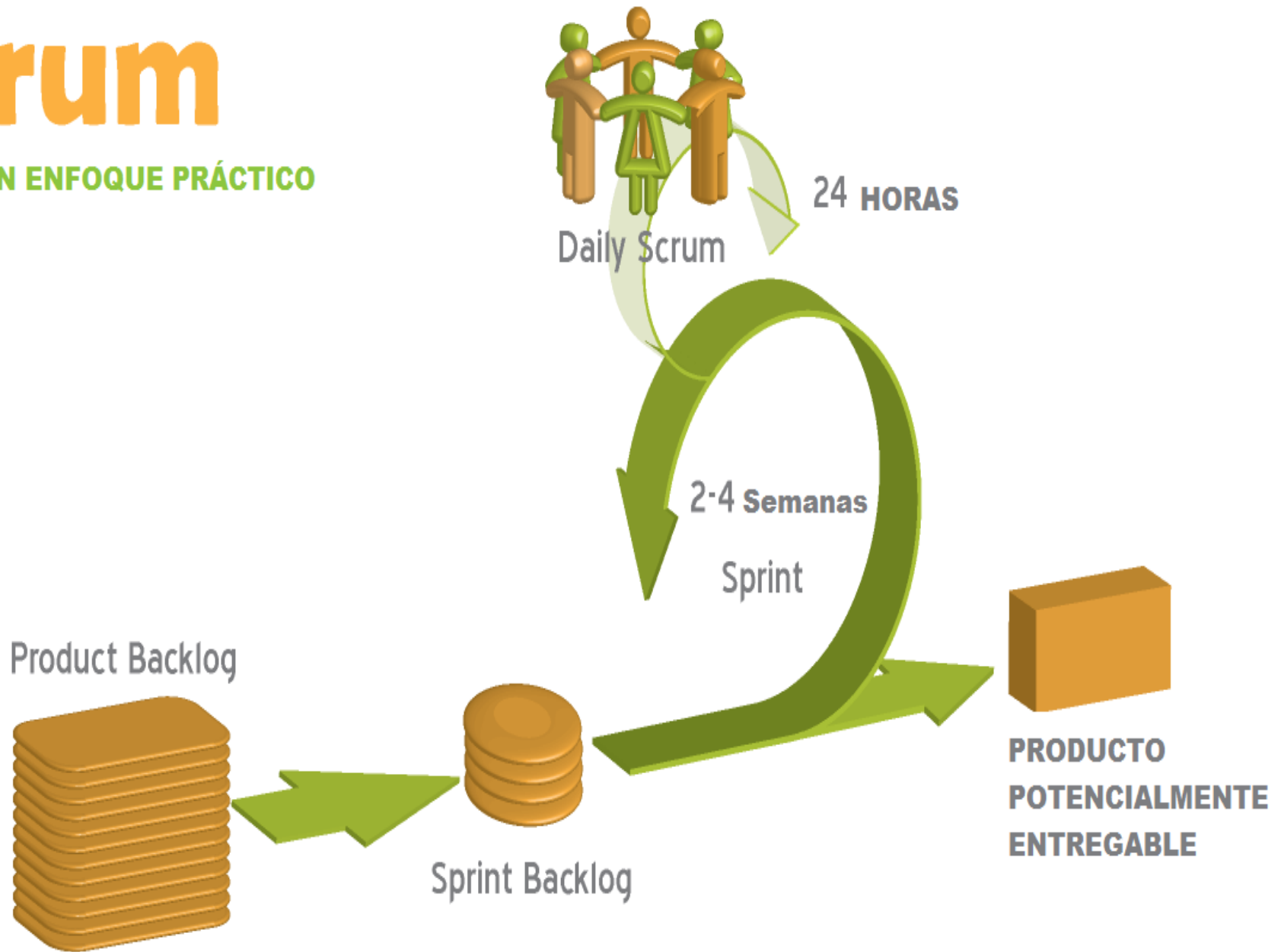
# ***SCRUM***

**Metodología de trabajo ágil**

**UN ENFOQUE PRÁCTICO**

# Scrum

UN ENFOQUE PRÁCTICO



# Índice

- Introducción
- Características
- Criterios de referencia
- Fortalezas de Scrum
- Trazabilidad
  - Definición
  - Tipos
- Los Sprint
- Prácticas para la evolución del proyecto
- Componentes de Scrum:
  - Reuniones
  - Elementos
  - Roles
- Herramientas
- Factores claves en Scrum

**“SCRUM es una estrategia de gestión donde se aplican de manera regular un conjunto de prácticas para mejorar el trabajo colaborativo y obtener el mejor resultado posible en la gestión de un proyecto software”**

# Introducción

- Simple pero duro. No se basa en el seguimiento de un plan sino en la adaptación continua a la evolución del proyecto
- Principios:
  - Adaptabilidad
  - Orientado a las personas y no a los procesos
  - Iterativo e incremental (desarrollo ágil)

# Características

- Metodología de trabajo ágil
- Diseñada para acortar el ciclo de desarrollo
- Conseguir una mejor aproximación entre las funcionalidades del software y los requerimientos del cliente
- Evitar la burocracia innecesaria
- Mayor versatilidad frente a los cambios
- Comenzar el trabajo lo más rápidamente posible
- Manejo más eficiente de los requerimientos cambiantes en un proyecto
- Mejorar la comunicación entre el cliente y el equipo desarrollador

# Criterios de referencia

- Aumento de la productividad y de la comunicación directa entre el cliente y el equipo desarrollador. Recomendado para equipos de trabajo pequeños (máx. 8 personas)
- Desarrollo incremental e iterativo producción frecuente de prototipos para evaluación del cliente
- Manejo más eficiente de los requerimientos cambiantes en un proyecto mejorando la versatilidad frente a los cambios.
- SCRUM no dice Qué hacer sino Cómo hay que hacer las cosas

# Fortalezas de SCRUM

<b>Gestión regular de las expectativas del cliente</b>	<ul style="list-style-type: none"><li>• Priorización de requisitos</li></ul>
<b>Resultados anticipados (“time to market”)</b>	<ul style="list-style-type: none"><li>• Demostración del proyecto en cada Sprint</li><li>• Priorización de requisitos por valor/coste</li></ul>
<b>Flexibilidad y adaptación</b>	<ul style="list-style-type: none"><li>• Replanificación en el inicio de cada iteración</li></ul>
<b>Retorno de inversión (ROI)</b>	<ul style="list-style-type: none"><li>• Priorización de requisitos</li></ul>
<b>Mitigación de riesgos</b>	<ul style="list-style-type: none"><li>• Desarrollo iterativo e incremental</li></ul>
<b>Productividad de calidad</b>	<ul style="list-style-type: none"><li>• Mejora continua</li><li>• Comunicación diaria del equipo</li><li>• TimeBoxing</li><li>• Equipo multidisciplinar</li><li>• Estimación de esfuerzo conjunta</li><li>• Compromiso del equipo</li><li>• Demostración de resultados</li></ul>
<b>Alineamiento entre cliente y equipo</b>	<ul style="list-style-type: none"><li>• Reuniones en cada itinerario (Sprint)</li></ul>
<b>Equipo motivado</b>	<ul style="list-style-type: none"><li>• Equipo autosugestionado</li><li>• Reuniones diarias y en cada Sprint</li></ul>



# Trazabilidad (I): Definición

- Capacidad de establecer de forma precisa e inequívoca el seguimiento de un producto y/o servicio durante todo su ciclo de vida.
- Está formado por un conjunto de acciones, medidas y procedimientos técnicos que permite identificar y registrar cada requerimiento de manera que se pueda seguir su ciclo de vidas tanto para atrás, desde su origen, como hacia delante, en la entrega o comercialización del producto.
- Toda la documentación, códigos y guiones de prueba deberán apuntar a su fuente de origen para permitir saber en todo momento el origen, la implementación y las pruebas que se hagan a cualquier requerimiento

# Trazabilidad (II): Tipos

- Bidireccional: A partir de un requisito se llega al código que lo implementa y a partir de un determinado código saber el o los requisitos a los que corresponde.
- Vertical: Garantiza que todos los requerimientos serán diseñados y que todos los diseños serán codificados y probados.
- Horizontal: Permite detectar si hay conflictos entre requerimientos, diseño, lógica de codificación y/o casos de prueba

# Los Sprint

- Cada iteración se llama sprint y se realiza una revisión de los requisitos con todas las personas involucradas en el proyecto
- Dentro de cada sprint, SCRUM gestiona la evolución del proyecto mediante reuniones breves de seguimiento en las que se revisa el trabajo realizado desde el hito anterior y los planes para el hito siguiente
- Las reuniones de seguimiento de cada sprint deben ser diarias

# Prácticas para la evolución del proyecto

- **Revisión de las iteraciones:** al final de cada sprint
- **Desarrollo incremental:** Al final de cada sprint debe haber una parte del producto operativa que se pueda inspeccionar y evaluar
- **Desarrollo evolutivo:** No se define la estructura final, la arquitectura o el diseño final del producto ya que los requisitos son cambiantes. Se utilizan técnicas de refactorización en las fases de diseño y codificación
- **Auto-organización:** Los equipos son auto-organizados con márgenes de decisión suficientes para tomar las decisiones que se consideren oportunas en los sucesitos sprint
- **Colaboración:** Se apuesta por una colaboración abierta entre todos los integrantes según sus conocimientos y capacidades, no según su rol o puesto.

# Componentes de SCRUM

- **Las Reuniones**
  - Planificación del Sprint
  - Seguimiento del Sprint
  - Revisión del Sprint
- **Los elementos**
  - Product Backlog
  - Sprint Backlog
  - Incremento
- **Los roles o responsabilidades:**
  - Responsables del producto: “Product Owner”
  - Responsables del desarrollo: “Scrum Team”
  - Responsables del funcionamiento de Scrum: “ScrumMaster”

# Las Reuniones (I): Planificación

- Reunión previa al comienzo de cada sprint:

- Cuál es el trabajo
- Objetivos a cumplir

- Intervienen todos los roles

- Se genera el “**Sprint Backlog**” o lista de tareas que se van a realizar

- Se determina el “**objetivo del Sprint**” (funcionalidad del negocio que se va a generar)



# Las Reuniones (II): Seguimiento

- Breve reunión diaria para repasar cada una de las tareas y el trabajo previsto de la jornada
- Sólo interviene el equipo de desarrollo
- Cada miembro responde a tres cuestiones:
  - **Trabajo realizado** desde la reunión anterior
  - **Trabajo que se va a realizar** hasta la próxima reunión de seguimiento
  - **Problemas** que se deben solucionar para realizar el trabajo propuesto



# Las Reuniones (II): Revisión

- Análisis y revisión del incremento generado
- Constituye la presentación de resultados





# Los Elementos (I): Product Backlog

- Se parte del resultado que se desea obtener **evolucionando** durante el desarrollo.
- Es un **documento vivo**
- Todos los integrantes del equipo de desarrollo podrán acceder a él **aportando ideas**.
- El responsable es una única persona (**Propietario del producto**)

# Los Elementos (II): Sprint Backlog

- Lista de trabajos que realizará el equipo durante el sprint
- **Incremento** previsto para el sprint
- **Compromiso** de ejecución
- Asignación de tareas de forma **personal** con estimación de tiempos y recursos necesarios

# Los Elementos (III): Incremento

- Demostración de los objetivos alcanzados en cada sprint
- Asistencia de todos los roles, “Product Owner” e incluso usuarios
- Sólo el Scrum Master puede abortar un Sprint debido a una de las siguientes razones:
  - La tecnología seleccionada no funciona o es incompatible con los objetivos definidos
  - Han cambiado las circunstancias de negocio
  - El Scrum Team ha tenido inferencias

# Los roles: Product Owner

- Persona conocedora del entorno de negocio del cliente y de la visión del producto.
- Representa a todos los interesados en el producto final
- Es el responsable del Product Backlog

## Procesos internos:

- Responsable de marketing
- El Product Manager

## Procesos externos:

- Responsable del proceso de adquisición del cliente

# Los Roles: Scrum Team

- Equipo multidisciplinar que cubre todas las habilidades necesarias para generar el resultado
- Se auto-gestiona y auto-organiza
- Dispone de atribuciones suficientes para toma de decisiones sobre cómo realizar su trabajo

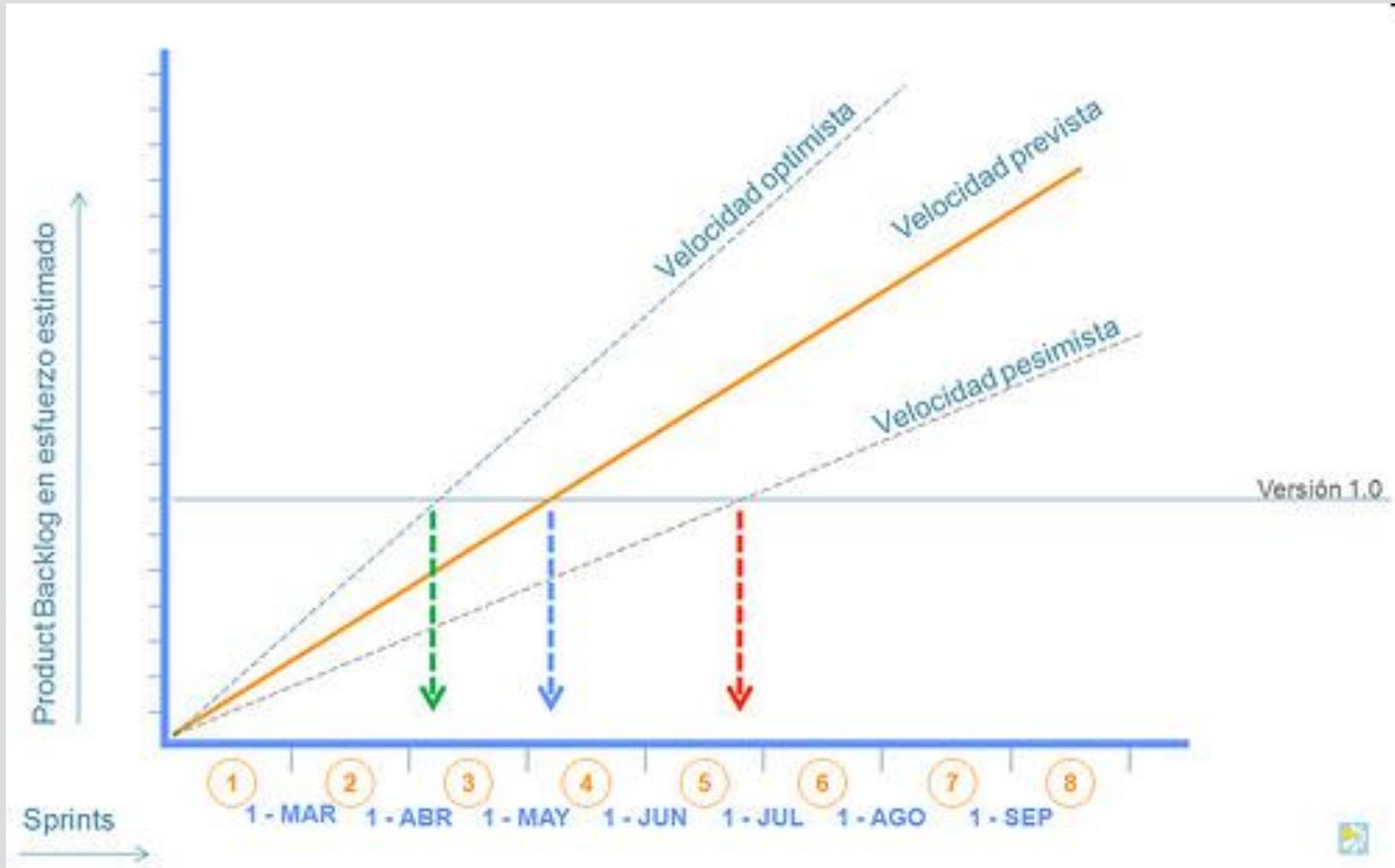
# Los roles: Scrum Master

- Garantiza el funcionamiento de los procesos y metodologías que se emplean
- No designa a una persona sino más bien a la responsabilidad de funcionamiento del modelo
- Es un role flexible:
  - Dirección de la empresa, con el conocimiento de gestión y desarrollo ágil y facilitando los recursos necesarios
  - Responsables del Departamento
  - Responsables del área de gestión de proyectos
  - ...

# Herramientas: Gráfico Burn-Up (I)

- Utilizado por el Product Owner
- Datos que muestra:
  - Las versiones previstas de un producto
  - Funcionalidades de cada una de ellas
  - Velocidad estimada
  - Fechas probables para cada versión
  - Margen de error previsto en las estimaciones
  - Avance real

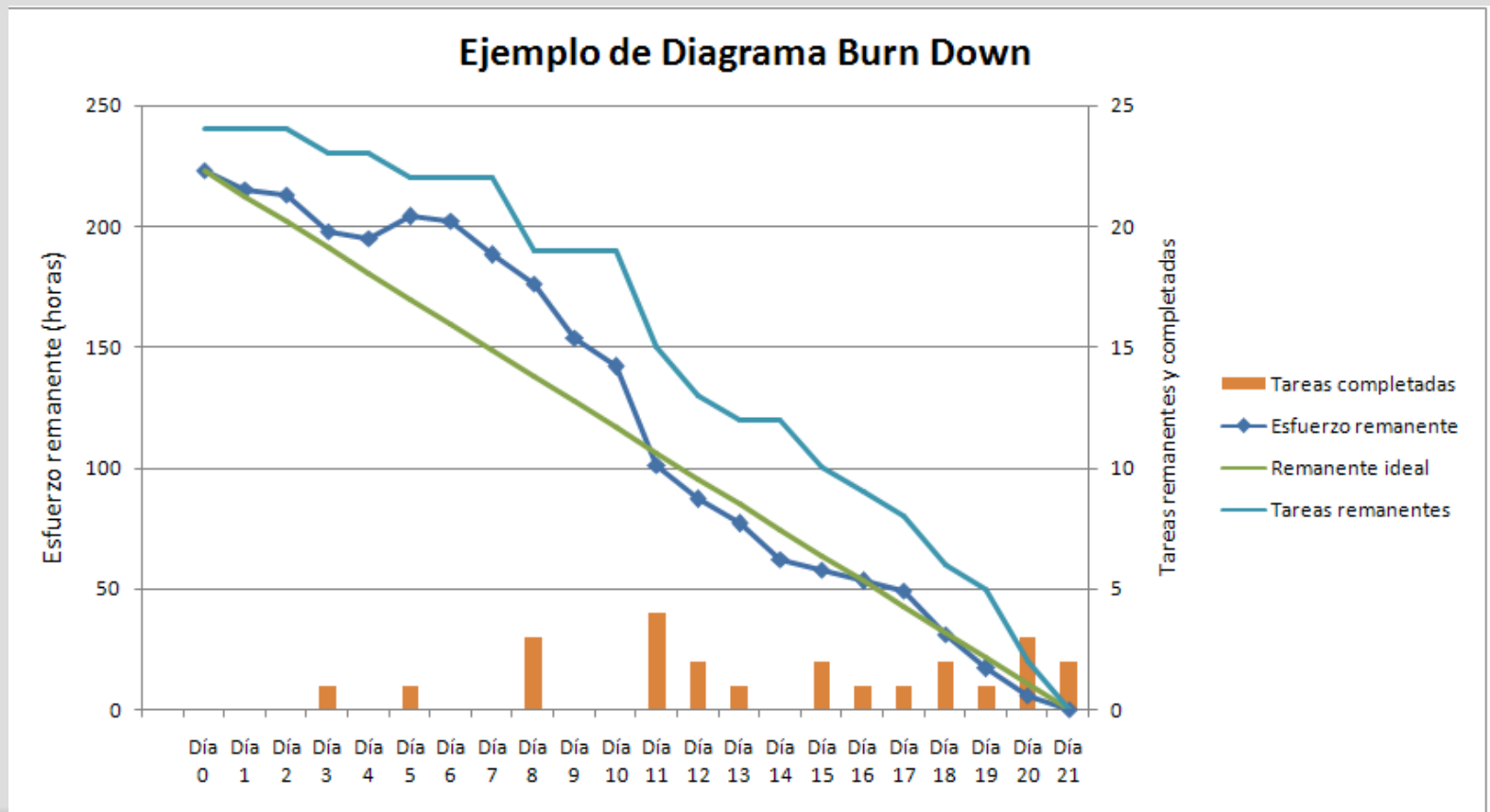
# Herramientas: Gráfico Burn-Up (II)





# Herramientas: Gráfico Burn-Down

- Utilizado por el Scrum Team para seguimiento del trabajo de cada Sprint



# Herramientas: Protocolos de decisión

- Solución:
  - Utilización de metodologías de estimación de tareas en un principio: Gráfico Pert
  - En base a los datos obtenidos en varios BurnDown se obtiene la velocidad media de obtención de resultados para los Sprint
  - Menos riesgo que la estimación de tareas ya que la estimación que se obtiene está basada en el esfuerzo del ScrumTeam

# Factores claves en Scrum

- Delegación de atribuciones al ScrumTeam: auto-organización y toma de decisiones
- Respeto entre las personas: confianza en los conocimientos y capacidades
- Responsabilidad y autodisciplina
- Trabajo centrado en el compromiso de desarrollo
- Información, transparencia y visibilidad en el desarrollo del proyecto

Gracias

---