

6. Scrum Framework

- Es un marco de trabajo ágil.
- se basa en ciclos cortos de trabajo: sprints
- en cada sprint, un equipo se autoorganiza para colaborar en entregar incrementos funcionales del producto.
- Se emplean ceremonias para estructurar el flujo de trabajo y la entrega de mejora continua.

CONCEPTOS PREVIOS

- *Historia de usuario*: forma de establecer una funcionalidad o característica deseada en el producto o servicio que se centra en la perspectiva del usuario. Sigue la forma:
 - como *estereotipo de usuario* quiero *tal funcionalidad o necesidad* para *tal motivación o tal beneficio*
- *Stake Holder*: cualquier persona con interés en el producto o el resultado del proyecto. Su opinion influye en el Product backlog. pueden ser:
 - usuarios finales
 - representantes del negocio
 - interesados externos al equipo scrum
- *Historia de usuario*: forma de expresar una necesidad o funcionalidad del sistema, desde la perspectiva de quien lo usará. Describe *qué* se necesita, *para quién*, y *porqué*. (el clásico *como, quiero, para*). No se centran en detalles técnicos, tienen criterios de aceptación indicando lo que se debe cumplir para considerarse terminada. Por ejemplo:
 - user story: Como usuario, quiero poder filtrar productos por categoría para encontrar lo que busco más rápido.
 - criterios de aceptación:
 - debe haber un menú desplegable con las categorías disponibles

- al seleccionar una categoría solo se muestran productos de esta categoría
- si no hay producto en esa categoría se muestra un mensaje.

Épicas: es una unidad grande de trabajo, representa una necesidad de alto nivel. Puede ser muy grande o compleja para implementarse en un solo Sprint. *se divide en historias de usuario más pequeñas*, estas sí se pueden completar dentro de un sprint.

- Una épica es una visión general o una funcionalidad amplia, las historias de usuario son partes más accionables de esa funcionalidad. Por ejemplo:
 - Épica: gestión de cuenta de usuario
 - historias de usuario:
 - como usuario quiero poder registrarme para crear una cuenta
 - como usuario quiero iniciar sesión con mi email y mi contraseña.
 - ...

Roles

Product Owner: responsable de definir los detalles de los criterios de aceptación de las historias de usuario. Gestiona las tareas del Backlog

Scrum Master: quien facilita las ceremonias Scrum, busca que el equipo se pueda autorganizar. Verifica que se sigan las reglas del scrum: elimina obstáculos y busca ayudar a mejorar la forma de trabajar (como una guía que aconseja en caso de encontrar obstáculos técnicos o de organización del equipo)

Development Team: desarrolladores y profesionales que desarrollan el sprint. Llevan a código el proyecto. Definen cuánto esfuerzo se necesita para cumplir con el objetivo del sprint.

Conceptos relacionados

- **Ceremonias ágiles**: buscan facilitar la comunicación entre los miembros del equipo, además de promover una única comprensión del objetivo a construir en ese sprint (todos andamos detrás del mismo objetivo...)
- **Product Backlog**: lista mantenida y actualizada por el product owner. incluye funcionalidad, requisitos técnicos o no funcionales. La lista contiene **TODO** Lo que se necesita en el producto y está priorizada (según el valor para el negocio).
 - Es dinámico.
 - Cada ítem es una unidad de valor del producto: puede ser una historia de usuario
 - por ejemplo: recuperación de contraseña
- **Sprint Backlog**: lista creada por el product owner y el equipo de desarrollo en conjunto. Es un **SUBCONJUNTO** del product backlog con los ítems para trabajar en un sprint específico.
 - cada tarea es atómica, está estimada y todo el equipo conoce cómo realizarlas.
 - suele ser fijo durante el sprint
 - Cada ítem es una unidad de trabajo, compuesto por un ítem del product backlog para este sprint + desglose técnico en tareas (pequeñas, detalladas)
 - por ejemplo: enviar el email con token.

Ceremonias de Scrum

- ceremonias ágiles para organizar el trabajo en equipo. Fomenta la comunicación para mejorar continuamente el proceso de desarrollo,
- garantiza que todo el equipo esté alineado en cuanto objetivos, estrategia para lograrlo (planificación) y en cuanto al feedback obtenido.
- Hay 5 ceremonias

Sprint planning: reunión previa al sprint.

- busca: identificar objetivos del sprint (se escogen historias del product backlog), alinear a todo el equipo Scrum para compartir una visión única del sprint.

- se recomienda que los miembros del equipo de desarrollo negocien los elementos del objetivo del sprint. (así se cumple con lo previsto)
- el procedimiento básico sería:
 - elegir historias del product backlog (participan todos los actores scrum)
 - cargarlas en el sprint backlog (se hacen atómicas, priorizadas, estimadas y si se agregan al sprint backlog es porque el equipo entiende cómo realizarlas y su dificultad.)

Al planificar el sprint todo el equipo colabora para decidir cuánto tiempo asignar al sprint, el product owner resume sus expectativas del sprint y establece el objetivo compartido. Las historias de usuario se agregan al sprint backlog (en base a un acuerdo del equipo de desarrollo). Durante esta reunión el equipo pregunta los detalles de las historias (esto se puede tratar en la refinación del backlog).

Se puede refinar los conceptos abarcados en el sprint planning: [Estimaciones, historias de usuario, dailys, y retros](#)

Daily Scrum:

- reunión breve y diaria (dura aprox 15 min). se busca coordinar la colaboración. no se hacen reportes individuales.
- participa el equipo de desarrollo y el scrum master (puede participar el owner)
- busca:
 - sincronizar el trabajo en equipo
 - identificar obstáculos o bloqueos
 - alinear la planificación del día con el objetivo del sprint

Sprint Review (DEMO):

- el último día del sprint el equipo scrum muestra lo conseguido (generalmente demo de features del producto) ante los stakeholders. Los stakeholders proporcionan feedback del trabajo expuesto, y los cambios sugeridos se documentan para considerarse.

Retro:

- reflexión al final del sprint (en un ambiente seguro) sobre el rendimiento, si se alcanzan las metas: ¿qué salió bien? ¿qué no salió bien? ¿qué podemos mejorar? aunque es mejor si se personalizan al producto para tener feedback más detallado

- se **analizan** dificultades y logros durante el sprint.
- Se **explora** oportunidades para mejorar el valor el valor proporcionado.
- En la reunión, asiste el scrum master y el equipo de desarrollo. Se debate, se acuerdan acciones para mejorar el proximo sprint, se trata de tener en cuenta las sugerencias del sprint review.
- Se usa para implementar mejoras para futuros sprints
- necesarias para mejorar los procesos con regularidad (reduce obstaculos)

Algunas otra colaboraciones:

- **refinamiento del backlog**: el product owner quia al equipo para refinar las historias de usuario más prioritarias. Se llevan a cabo **constantemente**, pueden ser espontáneas.
- **Parking Lot**: se abordan, resuelven y documentan impedimentos al progreso del sprin (se realizan en cualquier momento)

⚡ Estimaciones

- empleadas para planificar, no necesitan ser exactas. suelen estimar historias de usuario.
- busca determinar el **esfuerzo relativo** de las tareas basándose en la complejidad y el tamaño (NO ESTIMA TIEMPO, NI COMPLEJIDAD).
- el objetivo es que el equipo se pueda comprometer a un volumen de trabajo realista.
- Se puede medir en **story points**. El equipo establece cuántos story point se puede completar por sprint (esto es la **velocidad del equipo**)
- Hay diferentes formas de estimar os story points:

Planning Poker: estimación basada en consenso. se busca asignar un puntaje de esfuerzo a cada historia de usuario.

- el equipo se reúne, se discute el entendimiento de la historia de usuario
- para una historia de usuario, cada miembro elige una carta con un número (fibonacci digamos).
- todos revelan su carta al mismo tiempo
- se debaten las diferencias grandes y hasta llegar a consenso se vuelve a votar

T-shirt Sizes: Estimación más rápida, menos precisa que asigna tamaños de ropa

- el equipo le asigna a cada historia de usuario una talla según: **complejidad, esfuerzo o incertidumbre**.
- sirve como estimación rápida inicial.

Para estimar bien, el equipo necesita:

- aclarar dudas antes de estimar
- definir qué incluye y qué no la historia de usuario (definir cuánto se abarca)
- identificar las dependencias con otras tareas (torneos requiere de matches)
- hace preguntas al product owner:
 - ¿qué nivel de seguridad se espera?
 - la validación de contraseñas incluye reglas específicas?
 -

🔗 Métricas de la velocidad del equipo

Burndown Chart:

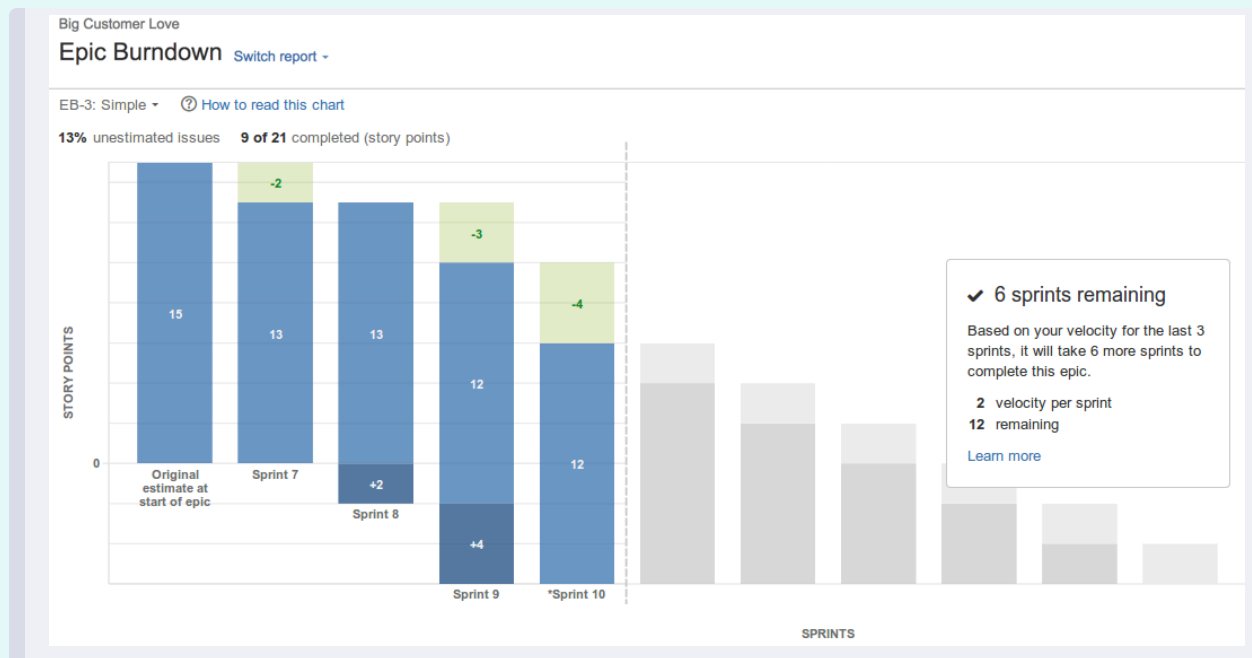
- sigue el **progreso del sprint**.
- el gráfico debe de decrecer, esto indica que el **valor a entregar faltante** disminuye.
- x v/s y: story point v/s tiempo (días del sprint)
- suele tener dos líneas:
 - línea proyectada: función lineal decreciente, velocidad constante: $\text{userPoint} / \text{semanaSprint}$.
 - ritmo ideal de trabajo diario: ejemplifica lo que se lograría si se completara la misma cantidad de trabajo todos los días
 - línea real: línea escalonada, muestra el trabajo realmente realizado día a día.
 - se actualiza a medida que el equipo completa tareas
- interpretación: si la **línea real** está por debajo de la idea, el equipo **va adelantado**, si está por encima entonces va atrasado.
- podemos detectar:
 - bloqueos si la línea real no baja por varios días. (Quizás días libres),

- subestimaciones: si es que se consideró que los storypoints no se "consumen" a la velocidad planteada
- si el ritmo es realista (la linea real se ajusta bien a la linea idea)



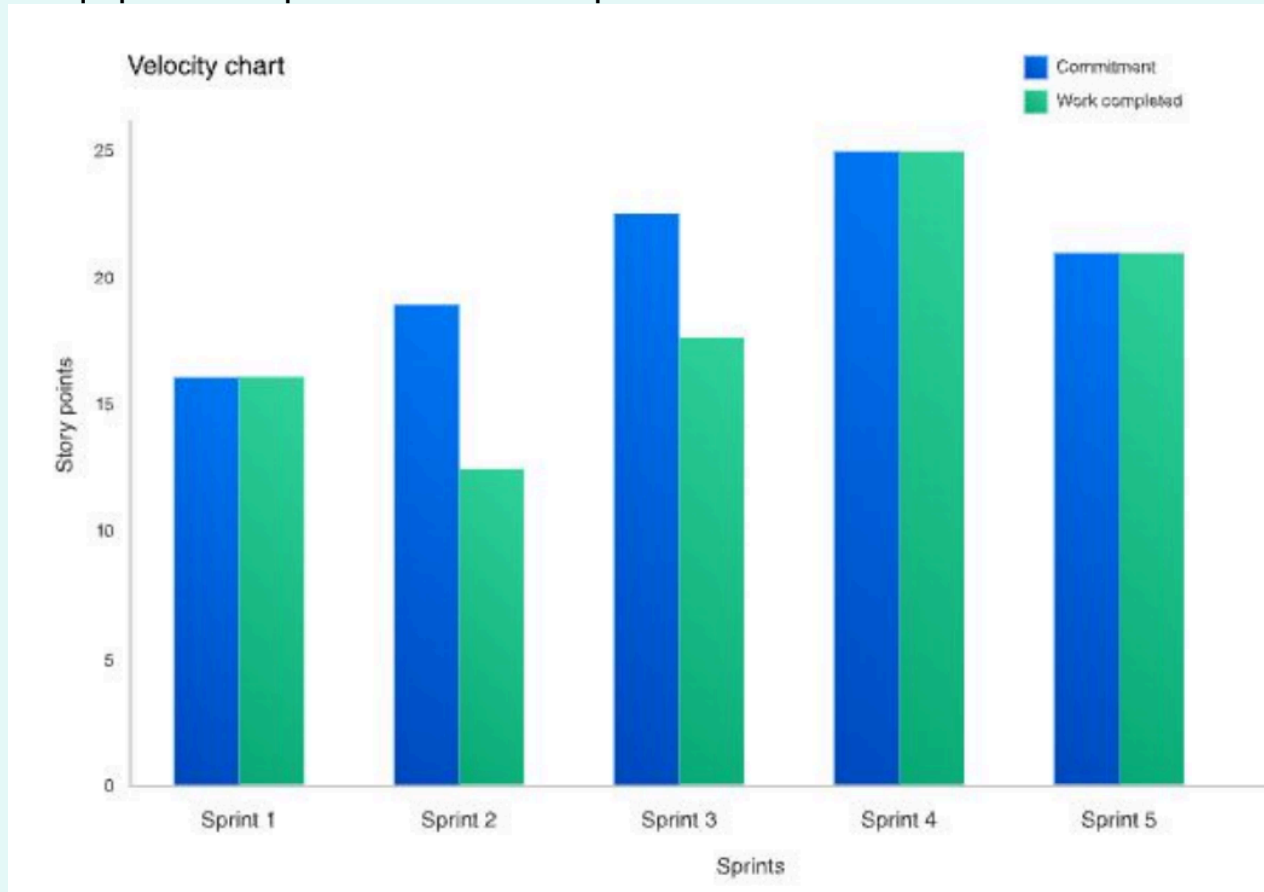
Epic burndown:

- variante del burndown chart, sigue el progreso de una épica
- x v/s y: tiempo (medido en sprints) v/s (story points)
- se puede medir el trabajo completado, el trabajo restante, trabajo agregado y el planificado.



Velocity chart:

- visualización del trabajo completado por un equipo en cada sprint. se usa para medir cómo evolucionó la velocidad del equipo.
- en el eje X tiene los sprints (discreto), y en el Y tiene una barra hasta la cantidad de story points alcanzados.
- adicionalmente lo compara con la cantidad de puntos de usuario con los que el equipo se comprometió en cada sprint.



NOTA: el *velocity chart* mide cuántos story point el equipo completo por sprint (el objetivo es *medir rendimiento y capacidad sostenida durante el proyecto*). El burndown chart muestra como van disminuyendo los puntos de usuario restantes en el sprint. (el objetivo es *ver si se llega a tiempo al sprint*)