

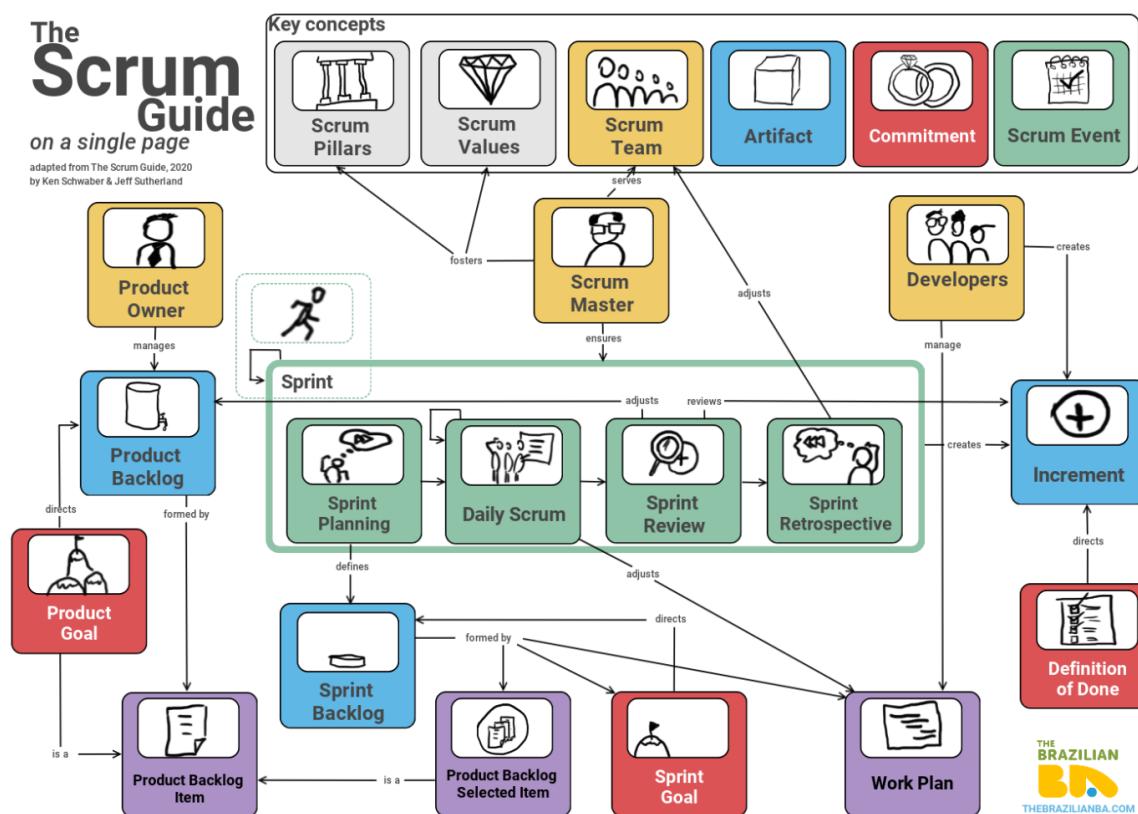
# Resumen Parcial 2. ISW

## Scrum

Se evalúa la guía Scrum y la información adicional de las profes. Si el parcial es a libro abierto, llevar la guía:

### Guía de Scrum

No aclarado en la guía: el refinamiento del product backlog es una **actividad continua**. No se hace durante un momento determinado, si no que puede surgir en cualquier momento.

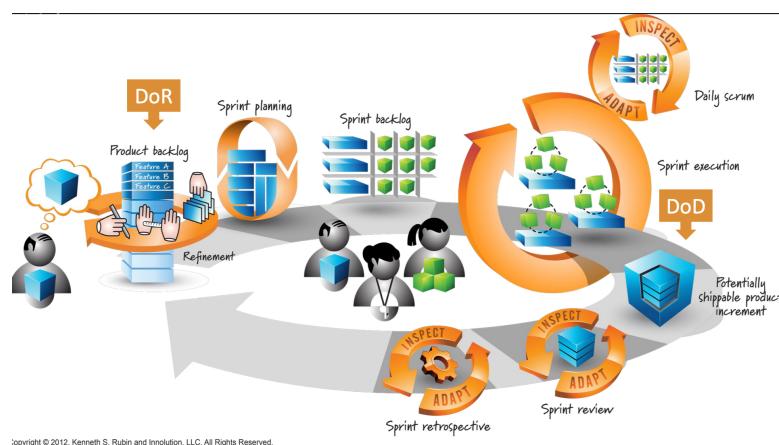


## Timebox

- Sprint: es una iteración. 1 mes o menos. La duración también se limita por:
  - No tanto como para que el riesgo sea inaceptable para el Product Owner.

- No tanto como para impedir sincronizar otros eventos del negocio con el equipo de desarrollo.
- Sprint planning: 8 horas máximo para un Sprint de un mes
- Daily Scrum: 15-30 minutos, depende del tamaño del equipo
- Sprint Review: 4 horas máximo para un Sprint de un mes
- Sprint Retrospective: 3 horas máximo para un Sprint de un mes (opcional para el PO)
- Refinamiento del PB: 10% del tiempo del Sprint

## Secuencia



## Características

- No implementar más de lo acordado
- No aceptar cambios durante la ejecución de una iteración
- Cuidar siempre la excelencia técnica del producto
- Utilizar el proceso que el equipo defina
- **Cimientos de Scrum:** Empirismo, Auto-organización, Colaboración, Priorización, Time Boxing.
- Nunca puede extenderse la duración de un Sprint.
- Daily Scrum: Asisten todos, Se responden las preguntas del scrum master/team
- Product Backlog:
  - Los ítems deberían agregar valor al cliente

- Los ítems están priorizados y ordenados según corresponda
- Los niveles de detalle dependen de la posición que ocupan
- Es un documento vivo
- No contiene ítems de acción o tareas de bajo nivel
- El Equipo SCRUM determina la cantidad de ítems del Product Backlog que podrán cumplimentar durante un Sprint.
- Puede contener errores a solucionar de iteraciones anteriores
- ¿Cuán frecuentes deben ser las reuniones de scrum de status del proyecto? **Diarias**
- El PO no es seleccionado por el Equipo de desarrollo y no asigna tareas.

## Definition of Done

El criterio de DoD (Definition of Done) establece cuándo una US se encuentra finalizada. Por lo general, consta de una checklist de actividades a cumplir antes de incorporar la historia en el próximo release de producto de software, pasando previamente por la aceptación del product owner. Por ejemplo:

- Diseño revisado
- Código Completo
  - Código refactorizado
  - Código con formato estándar
  - Código Comentado
  - Código en el repositorio
  - Código Inspeccionado
- Documentación de Usuario actualizada
- Probado
  - Prueba de unidad hecha
  - Prueba de integración hecha
  - Prueba de sistema hecha
- Cero defectos conocidos

Prueba de Aceptación realizada

En los servidores de producción

NO está relacionado al DoR ni al modelo INVEST.

## Definición de Listo (Definition of Ready)

Es una medida de calidad que constituye el equipo en conjunto para determinar que la US está en condiciones de entrar a una iteración de desarrollo (o sea, si entra a la Sprint Planning). Como *mínimo*, se requiere el modelo **INVEST**:

- Independent: no depende de otras US, se implementan en cualquier orden y son calendarizables.
- Negotiable: US escrita en términos de *qué* y no de *cómo*.
- Valuable: debe estar el *para qué*. Debe tener valor de negocio.
- Estimable: asignar un número (Story Points) que permita estimar tamaño relativo.
- Small: depende del equipo, pero deben ser consumidas en una iteración.
- Testable: relacionado a las pruebas de aceptación. Se debe poder demostrar que la US cumple con los criterios de aceptación.

Ejemplo:

Valor de negocio claramente expresada.

Detalles suficientemente comprendidos por el Equipo de forma tal que puedan tomar una decisión informada sobre si pueden completar el Product Backlog Item (PBI).

Dependencias identificadas y no hay dependencias externas que puedan impedir que el PBI se complete.

El equipo ha sido asignado adecuadamente para completar el PBI.

El PBI ha sido debidamente estimado y es lo suficientemente pequeño para ser completado en un Sprint.

Los criterios de aceptación son claros y testeables.

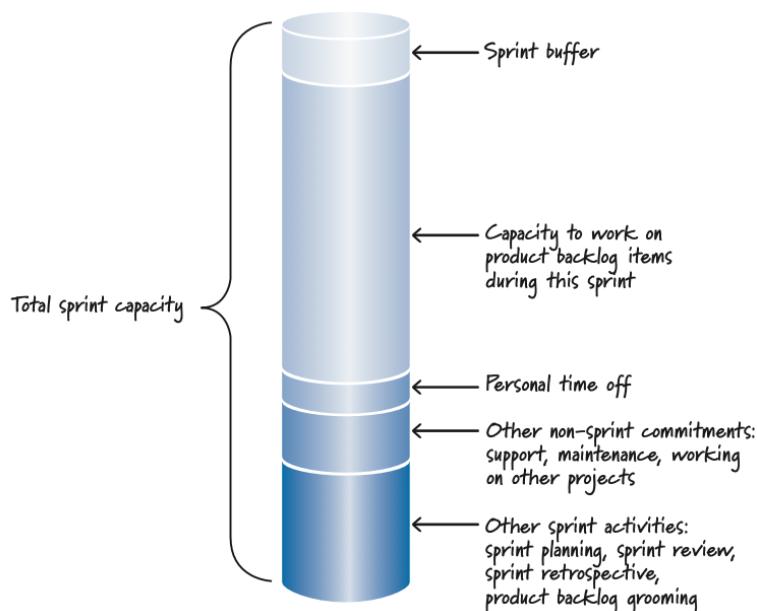
Los criterios de performance si hay, son claros y testeables.

El equipo comprende como mostrar el PBI en la Sprint Review.

# Capacidad

La única métrica que se estima, y se usa para ver cuánto compromiso se puede asumir como equipo en un determinado Sprint. Es una de las primeras cosas que se hacen en la Sprint Planning.

- Se determina en la Sprint Planning Meeting
- Se usa para estimar cuánto trabajo puede realizar un equipo durante un sprint
- Se puede medir en story points (puntos de historia)



Capacidad de desarrollo de un equipo en un Sprint

En equipos poco experimentados se mide en rango de horas ideales, en más experimentados en Story Points. Ejemplo:

Persona	Días disponibles (sin tiempo personal)	Días para otras actividades Scrum	Horas por día	Horas de Esfuerzo disponibles
Jorge	10	2	4-7	32-56
Betty	8	2	5-6	30-36
Simón	8	2	4-6	24-36
Pedro	9	2	2-3	14-21
Raúl	10	2	5-6	40-48

Persona	Días disponibles (sin tiempo personal)	Días para otras actividades Scrum	Horas por día	Horas de Esfuerzo disponibles
Total				140-197

Primera variable de definición: la duración del Sprint. Segunda: el cálculo de capacidad, Tercera: el objetivo del Sprint.

## Algunas preguntas

- *El Product Owner está en un workshop con el cliente. El cliente explica que el caso de negocio ha cambiado y algunos de los requerimientos ya no serán incluidos. Desafortunadamente, el equipo está trabajando en esas historias en el Sprint actual. ¿Qué debería hacer el Product Owner?*
  - El Product Owner debe llamar al equipo, terminar el Sprint y reiniciarlo con las prioridades cambiadas.
- *Suponiendo que elegimos el framework Scrum para gestionar el proyecto, El Product Owner está en un workshop con el Cliente. El Cliente le explica que el caso de negocio ha cambiado y que algunos de los requerimientos futuros ya no serán incluidos ¿Qué debería hacer el Product Owner?*
  - En el escenario donde el caso de negocio ha cambiado y algunos requisitos futuros ya no serán incluidos, el PO debe actualizar el **Product Backlog** para reflejar las nuevas prioridades y ajustar o eliminar los requisitos según sea necesario. El Product Owner es responsable de gestionar el Product Backlog, asegurándose de que sea transparente y refleje las necesidades y el valor más actual para los interesados. En esta situación, después de comprender el nuevo caso de negocio por parte del cliente, el Product Owner debe:
    - **Repriorizar el Product Backlog:** Eliminar los requerimientos futuros que ya no serán incluidos, y enfocarse en los elementos que ahora proporcionan el mayor valor para el negocio y los usuarios. Esto asegura que el equipo trabaje en los ítems más valiosos y relevantes.
    - **Colaborar con el Scrum Team:** Comunicar estos cambios al Scrum Team para garantizar que todos estén alineados con la nueva dirección y prioridades durante la próxima sesión de **Sprint**

**Planning** o reunión de **Refinamiento del Backlog**. Esto también ayuda a mantener la transparencia con el equipo de desarrollo y evitar que se trabaje en requisitos obsoletos.

- **Informar a los Stakeholders:** El Product Owner también debe gestionar las expectativas de los stakeholders, asegurándose de que estén al tanto de los cambios y cómo el producto evolucionará según las nuevas prioridades del negocio.
- *Suponiendo que elegimos el framework Scrum para gestionar el proyecto, ¿Qué debería hacer el Scrum Master en la siguiente situación? Durante una sesión de Sprint Review, mientras se mostraba una historia, aparece un mensaje de error en la aplicación y luego la aplicación se cae.*
  - Controla si el problema está relacionado con una nueva funcionalidad desarrollada en este Sprint, si es así la marca como no terminada.
  - Controla si el problema está relacionado con una nueva funcionalidad desarrollada en este Sprint, si no es así, la ingresa en la herramienta de seguimiento de defectos para su resolución.
- Pedro es el PO de un nuevo proyecto de software. El quiere completar el Product Backlog y crear la lista inicial de user stories y requerimientos. ¿Qué criterio debería considerarse mientras escribe la lista inicial de items?
  - User stories que describen explícitamente las funcionalidades requeridas por el cliente.
  - User stories requeridas para entender el concepto principal del producto.
- *Juan como Scrum Master de un equipo Scrum es invitado a una Scrum Daily Meeting a las 9 am. La reunión siempre excede el límite de tiempo de 15 minutos como todo el mundo habla al mismo tiempo. ¿Que debería hacer Juan?*
  - Asegurarse que el equipo conoce las reglas de la Daily Scrum Meeting; Interrumpir discusiones improductivas de inmediato; y también Introducir una posta (ej. una pequeña pelota). Sólo el miembro que la tenga puede hablar.
- *Juan como Scrum Master de un equipo Scrum es invitado a una Scrum Daily Meeting a las 9 am. Durante la reunion los miembros del equipo*

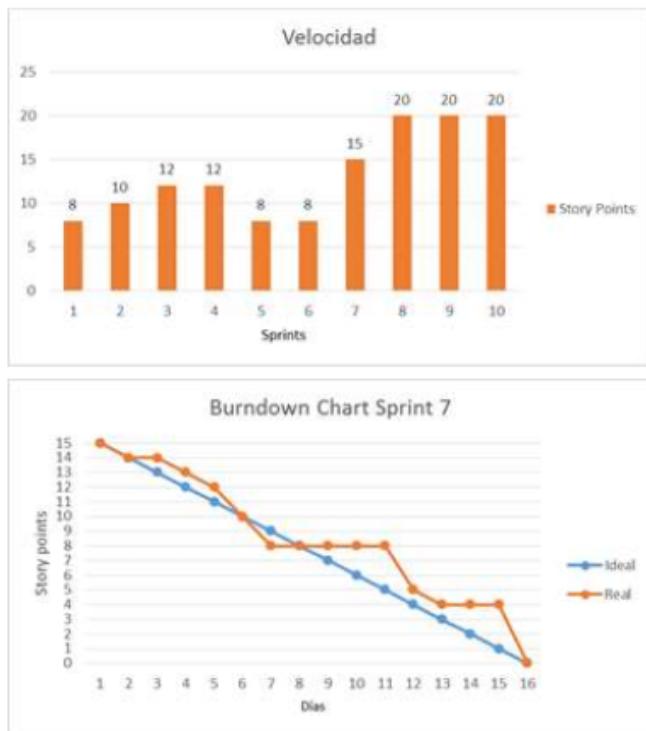
*actualizan los esfuerzos restantes de sus actividades. Que debería hacer Juan?*

- Tiene que sumar los valores y actualizar el gráfico de Burndown del Sprint.
- *Al final del Sprint, el Product Owner es invitado a una Sprint Review, el equipo explica que tuvo un problema con un build y que la versión oficial del build no está disponible en ese momento. En lugar de eso ellos prepararon una demo en los equipos de desarrollo. ¿Que debería hacer el Product Owner?*
  - Dar una mirada a las características, pero decirles que no están aceptadas.
  - Pedirles al Scrum Master una reunión adicional cuando el paquete oficial este disponible.
- *Durante la discusión con su cliente el Product Owner notó que se había olvidado de algunas historias importantes. ¿Que debería hacer el Product Owner?*
  - Agregar las nuevas historias en el Product Backlog.
- *En una discusión con uno de los miembros del equipo el Scrum Master advierte que hay aspectos sobre una user story que no están completamente definidos. El PO está muy ocupado y no está disponible para realizar una reunión y responder las preguntas. ¿Qué debería hacer el Scrum Master?*
  - Discutir el tema con el PO y decidir una forma de responder las preguntas.
- *Juan, el Scrum Master del equipo, organiza una Retrospective Meeting. Juan, el equipo y el Product Owner se reúnen en una habitación separada. La reunión empieza y Juan recuerda a cada uno los objetivos y reglas. ¿Cómo debería continuar Juan?*
  - Primero cada uno en la habitación debería decir una breve frase que represente como se sintió en el ultimo Sprint.
- *Las siguientes afirmaciones: (pregunta inicial)*
  - *Los proyectos utilizan estándares y procedimientos*
  - *Se realizan revisiones y auditorías independientes*

- Se produce documentación para dar soporte al mantenimiento y la mejora.
- Las desviaciones a los estándares y procedimientos son expuestas lo más pronto posible.
- El control de calidad en sí mismo es ejecutado con estándares establecidos

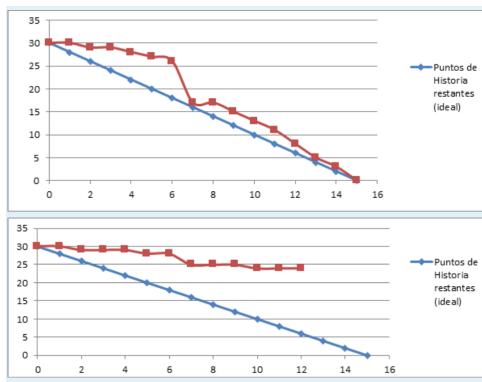
son aportes vinculados a: (pregunta termina)

- **Incorporar actividades de Aseguramiento de Calidad en la organización. (respuesta)**
- Análisis de gráficos:



- Ambos graficos resultan positivos, o al menos normales. El grafico de velocidad muestra en general una mejora en el desempeño a lo largo de los sprints, esto es esperable y deseable si el equipo de desarrollo aprende y crece como tal, mejorando su capacidad de generar incrementos. La caída en los Sprints 5 y 6 podría tranquilamente deberse a el faltante de algun miembro del equipo, o un problema inesperado, aun asi no es algo grave, dada la naturaleza cambiante del desarrollo de software y los humanos que lo desarrollan es esperable tener este tipo de "caídas" eventualmente.

- El grafico de la quema de puntos muestra tambien una historia positiva, eventualmente se completo el objetivo del sprint, en el dia 16, si bien hubo retrasos al principio, y otros dos retrasos grandes casi al final del sprint, el equipo logro llegar al objetivo. Tal vez esto podria indicar que no se fraccionaron correctamente las US, y habia alguna con demasiados puntos, que demoro mucho hasta ser completada.
  - Especialmente en el día 7 se puede ver que no se queman puntos durante 4 dias consecutivos y luego se hace una caída de 3 puntos en un solo dia.
- *Durante una Daily Meeting, uno de los miembros del equipo informa que tiene problemas para continuar las tareas porque una de las licencias del software que necesita ha expirado. ¿Qué debería hacer el Scrum Master?*
  - Preguntar al resto del equipo si alguien más está en la misma situación. Luego de la reunión llamar al proveedor y conseguir las licencias.
- *Juan como Scrum Master de un equipo Scrum es invitado a una Scrum Daily Meeting a las 9 am. Durante la reunión Juan nota que las lista de actividades abiertas crece y solo algunas pocas se han cerrado. ¿Qué debería hacer Juan?*
  - Controlar si hay un problema con el equipo o si solo esta relacionado a un miembro del equipo. Después de discutir el problema asegurarse de que el foco es cerrar las actividades iniciadas.
- Al final del sprint, Juan como Scrum Master de un equipo Scrum es invitado para una Sprint Review Meeting. Durante el Sprint el Product Owner estuvo la mayoría del tiempo ocupado y no estuvo disponible para discusiones. Cuando se revisaban las user stories el Product Owner reclamó que la mayoría de las historias no fueron implementadas como el Cliente esperaba. ¿Qué debería hacer Juan?
  - Después de la reunión discutir la situación con el Product Owner y decidir sobre las medidas para mejorar su disponibilidad durante los siguientes Sprints.
- Suponiendo un desarrollo ágil, analice en los siguientes gráficos el comportamiento del proyecto manifestado en los mismos y explique detalladamente la interpretación que hace luego del análisis:



- En estos graficos podemos observar la comparación entre quema de puntos de historia de manera ideal (en azul) y quema de puntos reales durante el sprint (rojo).
- Análisis primer gráfico: en el primer gráfico observamos que los puntos de historia se mantienen estables es decir, no se queman según lo esperado. En este caso podemos suponer que el equipo enfrentó problemas que dificultaron la quema de puntos. Vemos que a partir del sexto día se produce una quema importante de puntos (aproximadamente 10), probablemente en ese momento encontraron solución al problema y pudieron avanzar de la manera esperada. Otra suposición que podemos hacer es que los puntos fueron quemados de manera adecuada pero los integrantes realizaron la actualización en la herramienta que genera el burndown chart recién el sexto día (mal uso de herramienta)
- Análisis segundo grafico: en el segundo gráfico podemos observar que el equipo solo quemó 5 de los 30 puntos estimados para dicho sprint. Podemos suponer que las historias fueron mal estimadas, que surgieron dudas técnicas, que las user no estaban correctamente refinadas y detalladas, que hubo falta de recursos en el equipo (se enfermaron desarrolladores, no se consideraron licencias). Claramente este es un sprint que no va a cumplir con las expectativas de nuestro PO ya que se han quemado menos del 30% de los puntos estimados en la planning.
- Juan como Scrum Master de un equipo Scrum es invitado a una Scrum Daily Meeting a las 9 am. El equipo está reunido frente al tablero Scrum, pero uno de los miembros siempre llega tarde o ni siquiera aparece. Que debería hacer Juan?
  - Hablar con el miembro del equipo para encontrar una solución

- Juan como Scrum Master de un equipo Scrum es invitado a una Scrum Daily Meeting a las 9 am. Durante la reunión dos de los miembros del equipo discuten respecto de una posible solución para una de las tareas que no han iniciado todavía. Que debería hacer Juan?
  - Interrumpir la discusión y organizar una reunión de seguimiento inmediatamente después que termine la Daily Meeting.
- Juan como Scrum Master de un equipo Scrum es invitado a una Scrum Daily Meeting a las 9 am. Durante la reunión el equipo completo comienza una discusión profunda y finalmente comienzan a culparse entre ellos. Que debería hacer Juan?
  - Detener la discusión y continuar con la reunión.
  - Discutir la situación con el equipo y decidir juntos qué hacer para mejorar la situación.
- En una discusión, el arquitecto de software señala que hay tareas adicionales que deben implementarse independientemente de cualquier requerimiento del cliente. Que debería hacer el Product Owner?
  - Agregar las tareas al PB.
    - Justificación: pueden ser features, cambios de features, defectos, mejoras técnicas, trabajo de adquisición de conocimiento.
- Al final del sprint, Juan como Scrum Master de un equipo Scrum es invitado para una Sprint Review Meeting. La documentación para dos de las user stories no se han terminado conforme se pidió en el criterio de hecho (DoD). Que debería hacer Juan?
  - Reestimar y terminar esas dos user stories en un Sprint posterior.
- Juan como Scrum Master de un equipo Scrum es invitado a una Scrum Daily Meeting a las 9 am. Uno de los miembros del equipo reporta que necesita soporte para el testing de una de sus tareas. Sin ese soporte requerido la tarea no podrá terminarse hasta el final del Sprint. Un segundo miembro, el que tiene un conocimiento más amplio sobre testing, está actualmente ocupado con otras actividades de mayor prioridad. Que debería hacer Juan?
  - Controlar si algún otro miembro del equipo puede darle soporte en testing.

- Discutir las prioridades con el Product Owner para entregar el maximo valor para el final del Sprint, como esta situacion puede tener un impacto en la seleccion de los entregables finales del Sprint.
- En una reunion de retrospectiva el equipo reclama que tiene un grupo de user stories para implementar y que todavia no sabe mucho acerca de la estrategia de release detras de esas actividades. Que deberia hacer el Product Owner?
  - Crear un Plan de Release que muestra las prioridades de las user stories y sus fechas de release.
- En un workshop el arquitecto explica que la implementación de cierto conjunto de requerimientos requerirá un gran esfuerzo debido a las limitaciones del framework de desarrollo. ¿Qué debería hacer el Product Owner?
  - Estimar el retorno de inversion (ROI) considerando el valor de negocio, tiempo y costo. Priorizar las caracteristicas de acuerdo con eso.
- El Product Owner es invitado a un workshop para identificar y discutir riesgos de un proyecto. Durante la evaluación de riesgos conocidos, ellos descubrieron que uno de los riesgos previamente identificados ya no es valido. ¿Qué debería hacer el Product Owner?
  - Remover las actividades asociadas a la mitigación del riesgo del Product Backlog

## Taskboard



El tablero básico contiene sólo 3 columnas: To Do, Doing y Done. En equipos donde una US es realizada por más de un miembro, tablero con las tareas de las US discriminadas:

Story	To Do	In Process	To Verify	Done
As a user, I... 8 points	Code the... 9 Code the... 2 Test the... 8	Test the... 8 Code the... 8 Test the... 4	Code the... DC 4 Test the... SC 8	Test the... SC 6 Code the... DC 8 Test the... SC 8 Test the... SC 6
As a user, I... 5 points	Code the... 8 Code the... 4	Test the... 8 Code the... 6	Code the... DC 8	Test the... SC 6 Test the... SC 6

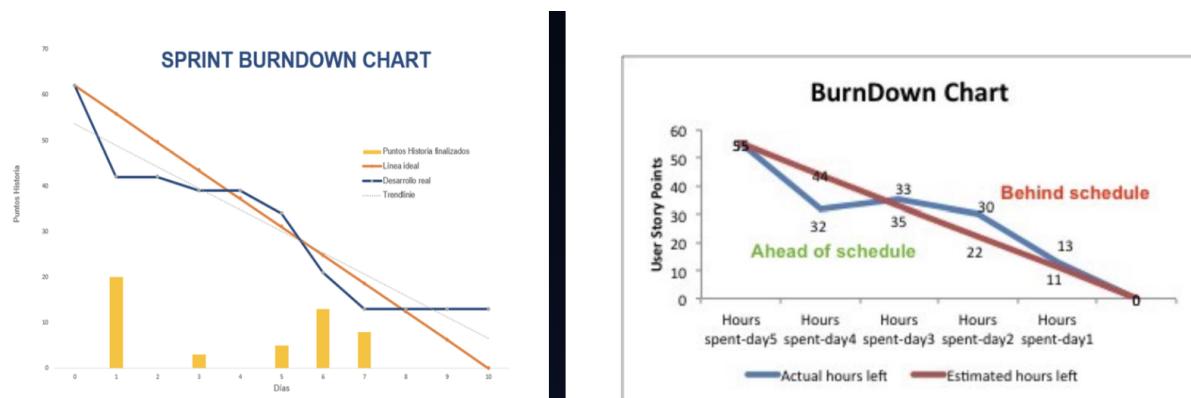
Tablero con tareas. También se suele agregar una columna "done done", que contiene las US cuyas tareas fueron completadas.

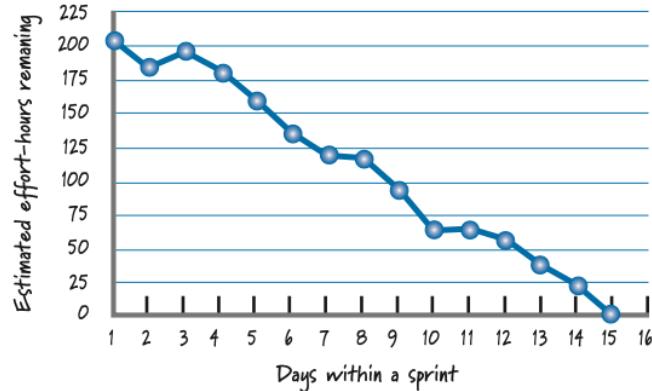
En Agile, las tareas se asignan con sistema **pull**: cada desarrollador elige las tareas con las que se compromete. Deben tener un nivel de granularidad alto.

## Sprint Burndown Charts

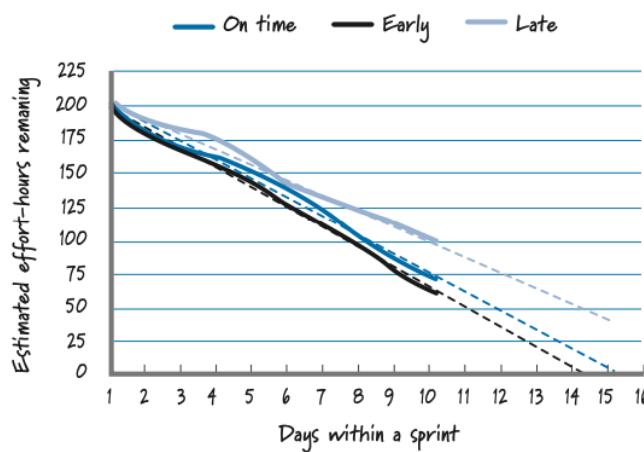
Un Burndown chart representa la tendencia del trabajo remanente a lo largo del sprint, el release o el producto.

Idealmente se actualiza en la Daily, por el Scrum Master. Existe una versión temporal y una permanente. Los **temporales** se limpian a final de cada Sprint, luego de calcular la velocidad del equipo según el gráfico:





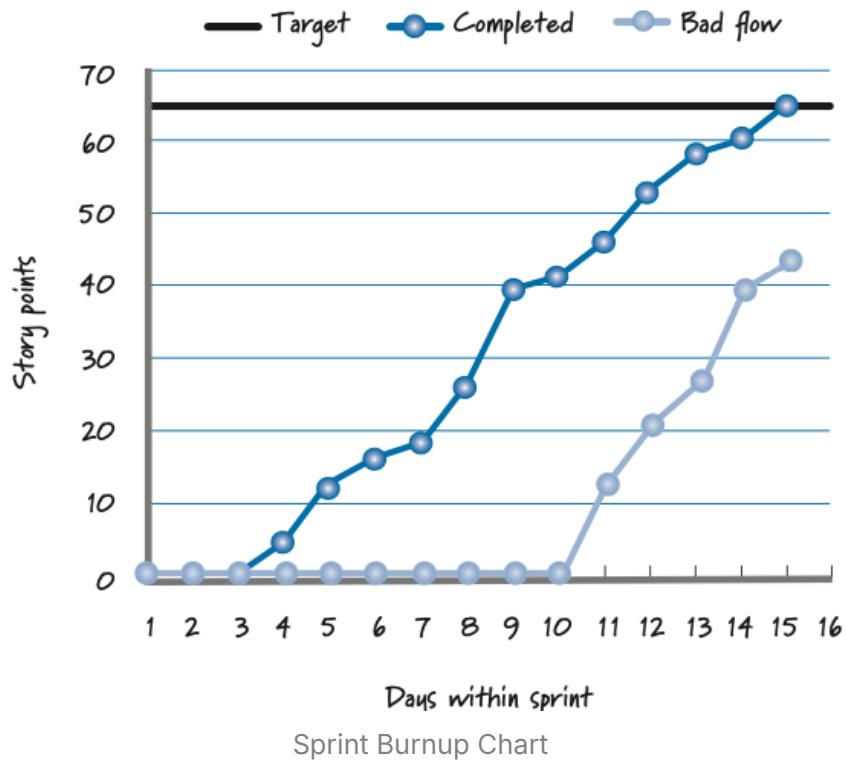
Otro Sprint Burndown chart temporal, que tiene un pico cerca de 200 en el día 3. Eso puede darse por una reestimación. (una US que pasó de 2 a 5 SP)



Sprint Burndown chart con líneas de tendencia

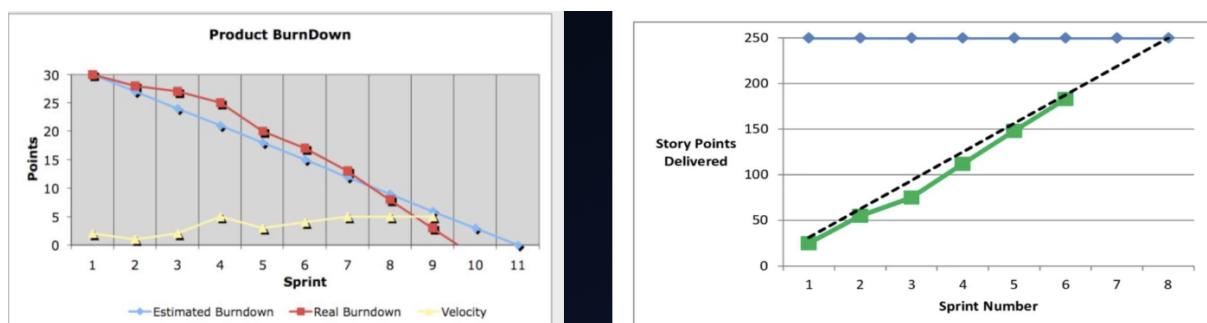
Si el equipo terminó el trabajo comprometido en el Sprint y le sobró tiempo, es recomendable terminar el Sprint en lugar de agregar otra funcionalidad.

La versión **permanente** hace el seguimiento sobre el producto a lo largo del tiempo en base a sprints.



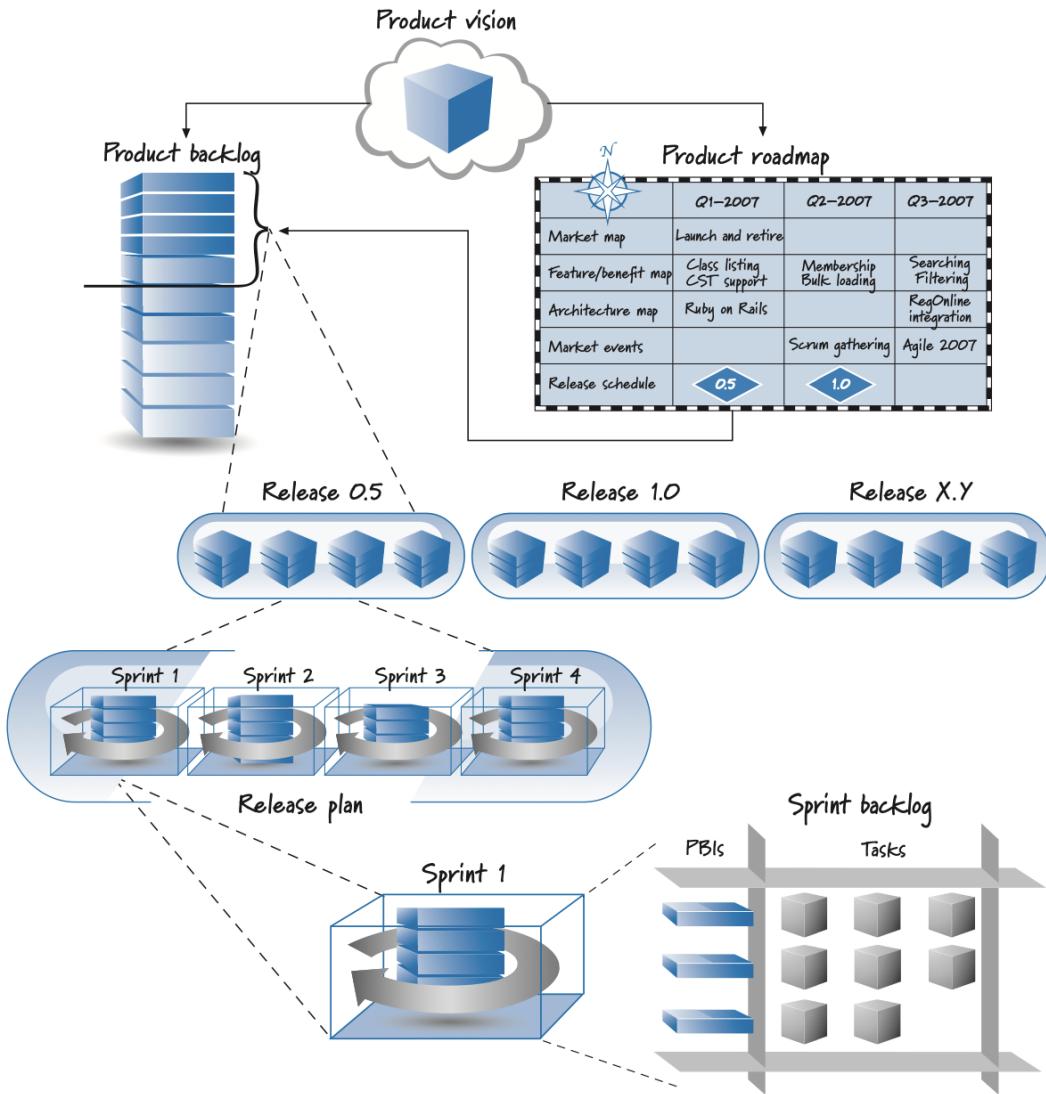
En los Sprint Burnup Chart, el trabajo puede representarse tanto en horas de esfuerzo (como en el gráfico de Sprint Burndown Chart) como en puntos de historia. Muchas personas prefieren usar Story Points en sus gráficos de avance, porque al final del sprint lo único que realmente importa al equipo Scrum es el trabajo de valor para el negocio que se completó durante el sprint, y esto se mide en puntos de historia (o días ideales), no en horas de tareas.

Otros gráficos:



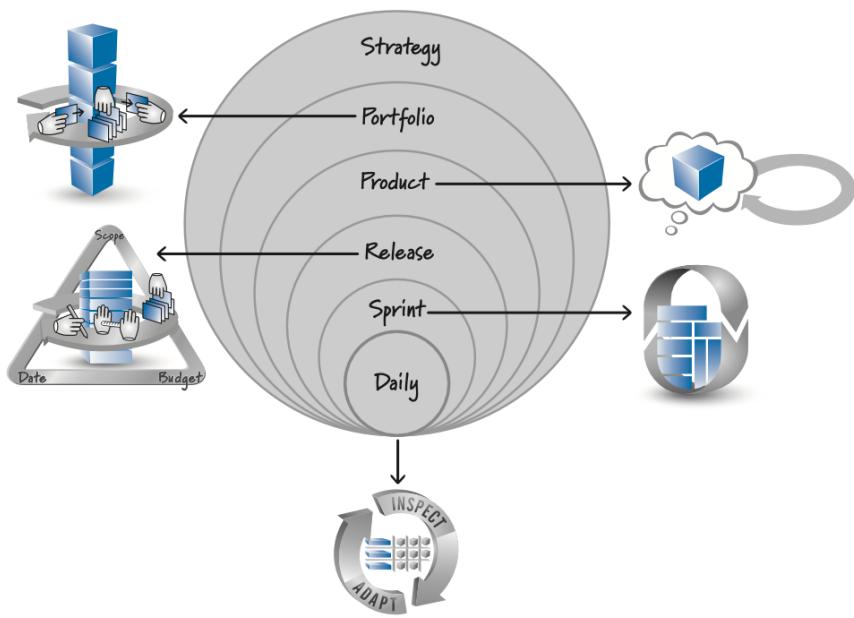
## Planificación

Se planifica a nivel jerárquico:



La visión de producto lleva a configurar el PB (no es necesario que esté completo, debe contener al menos funcionalidades que permitan iniciar el trabajo). La sección de Release 0.5, 1.0, X.Y es planificación de producto. La planificación de release es la parte de abajo, con la cantidad de sprints.

Existen distintos niveles de planificación:



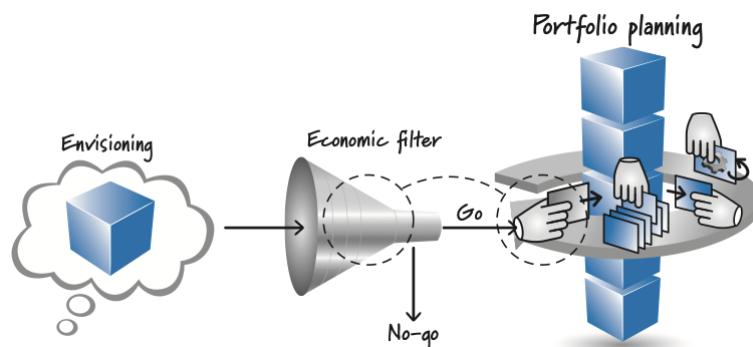
- **Diario:** Daily Scrum
- **Iteración:** Sprint Planning
- **Release:** Release Planning (cuántos Sprints se necesitan para completar las funcionalidades incluidas en el Release) - Triple Restricción
- **Producto:** Sumatoria de Releases a lo largo del tiempo.
- **Portfolio:** Productos de la empresa (Ej. Microsoft tiene Excel, Word, Powerpoint, etc.)
- **Estrategia**

Nivel	Horizonte	Quién	Foco	Entregable
Portfolio	1 año o más	Stakeholders y Product Owners	Administración de un Portfolio de Producto	Backlog de Portfolio
Producto	Varios meses o más	Product Owner y Stakeholders	Visión y evolución del producto a través del tiempo	Visión de Producto, Roadmap y características de alto nivel
Release	3 (o menos) a 9 meses	Equipo Scrum, Stakeholders	Balancear continuamente el valor de cliente y la	Plan de Release

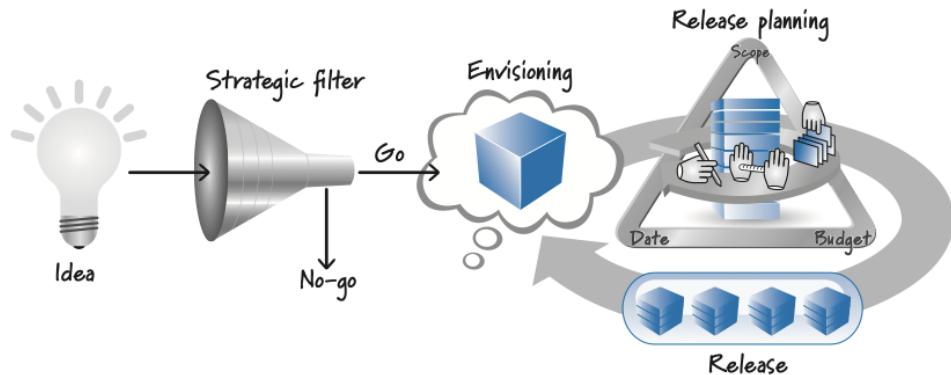
Nivel	Horizonte	Quién	Foco	Entregable
			calidad global con las restricciones de alcance, cronograma y presupuesto	
Iteración	Cada iteración (de 1 Equipo Scrum semana a 1 mes)	Equipo Scrum	Que aspectos entregar en el siguiente sprint	Objetivo del Sprint y Sprint Backlog
Día	Diaria	Equipo Scrum (al menos los que trabajan en IPB)	Cómo completar lo comprometido	Inspección del progreso actual y adaptación a la mejor forma de organizar el siguiente día de trabajo

El **Portfolio backlog** se estima en tamaños de remera (S, M, X, XL). El **Product Backlog** se estima generalmente en SP, y las **tareas del Sprint backlog** se estiman en horas ideales. No todos los equipos dividen las US en tareas.

En empresas también se **planifica el portfolio**:



## Planificación de Producto



Idea → Filtro estratégico → Envisioning (Product Planning) → Release Planning

La planificación de producto comienza con una idea para un producto generada por alguien o un equipo (ideación). La idea se evalúa en función de la dirección estratégica de la organización para ver si está alineada y vale la pena explorarla más a fondo. Si pasa esta evaluación, se inicia la planificación inicial para definir cuál debería ser la primera versión mínima, lo que permite una entrega rápida y de bajo costo. Esta primera versión se pone en manos de los usuarios para recopilar retroalimentación, que puede confirmar o desafiar las suposiciones iniciales sobre los clientes objetivo y las características deseadas. Según esta retroalimentación, el equipo puede continuar con la visión actual o pivotar hacia una nueva dirección.

## Participantes

El propietario del producto es el único participante requerido durante el proceso de visualización inicial, aunque a veces colabora con partes interesadas internas y especialistas en investigación de mercado, desarrollo de casos de negocio, diseño de experiencia de usuario y arquitectura de sistemas. Idealmente, el ScrumMaster y el equipo de desarrollo participan en este proceso para ofrecer retroalimentación y evitar transferir la visión a otro equipo. Una vez que comienza el desarrollo, el equipo Scrum completo debe ser incluido en cualquier esfuerzo de revisión de la planificación.

## Proceso

La entrada principal para la planificación inicial es una idea que ha pasado por el filtro estratégico, mientras que la re-planificación se basa en una idea revisada debido a la retroalimentación, cambios en el financiamiento, acciones de competidores u otros factores significativos. Otros insumos incluyen el horizonte de planificación, la fecha esperada de finalización de la visualización,