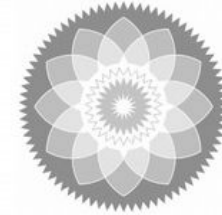




*Universidad Nacional de Tucumán*



2016 AÑO DEL  
BICENTENARIO  
DE LA DECLARACIÓN  
DE LA INDEPENDENCIA  
NACIONAL

Universidad Nacional de Tucumán  
Facultad de Ciencias Exactas y Tecnología

Ingeniería de Software II

**Agile / SCRUM básico**

Mg. Héctor A. Valdecantos

# Manifiesto Ágil

- Febrero del 11 al 13 del 2001
  - 17 desarrolladores de software
    - <https://agilemanifesto.org/>
  - Discuten temas de desarrollo de software
  - Producen el manifiesto

# Manifiesto Ágil: 4 valores

*“Estamos descubriendo formas mejores de desarrollar software tanto por nuestra propia experiencia como ayudando a terceros. A través de este trabajo hemos aprendido a valorar:”*

**Individuos e interacciones** sobre procesos y herramientas

**Software funcionando** sobre documentación exhaustiva

**Colaboración con el cliente** sobre negociación de contratos

**Respuesta a los cambios** sobre seguimiento de un plan

<https://agilemanifesto.org/>

<https://agilemanifesto.org/iso/es/manifesto.html>

# Manifiesto Ágil: 12 principios

Guían las metodologías ágiles

1. Satisfacer las necesidades del cliente
2. Describen una cultura donde el cambios es bienvenido
3. Se enfoca en tener un software siempre funcional
4. Ponen el foco de trabajo en colaboración con el cliente
5. Equipos con individuos motivados
6. Enfatiza las interacciones humanas
7. Software funcional como medida de progreso
8. Fomenta un ritmo constante y sostenible de desarrollo
9. Busca la excelencia técnica
10. Busca eliminar desperdicio
11. Clama por equipos auto-organizados para obtener calidad
12. Un proceso activo y adaptativo con intervalos regulares

<https://agilemanifesto.org/iso/es/principles.html>

# Objetivos metodologías ágiles 1/2

- Poner a las personas en un papel principal
- Liberar el software (release) tan rápido como sea posible
- Obtener una buena visibilidad del proyecto y su progreso
- Responder al cambio y reducir el costo a cambiar



# Objetivos metodologías ágiles 2/2

- Motivación y compromiso
- Transparencia en el proceso
- Predictibilidad de costos y tiempos
- Centrarse en el valor de negocio
- Calidad a través de revisiones y pruebas
- Riesgo limitado por incrementos pequeños
- Flexibilidad ante el cambio

# Desarrollo de software ágil

“Una forma fructífera de pensar el desarrollo de software es considerándolo como un juego cooperativo de invención y comunicación.” [Cockburn, 2001]

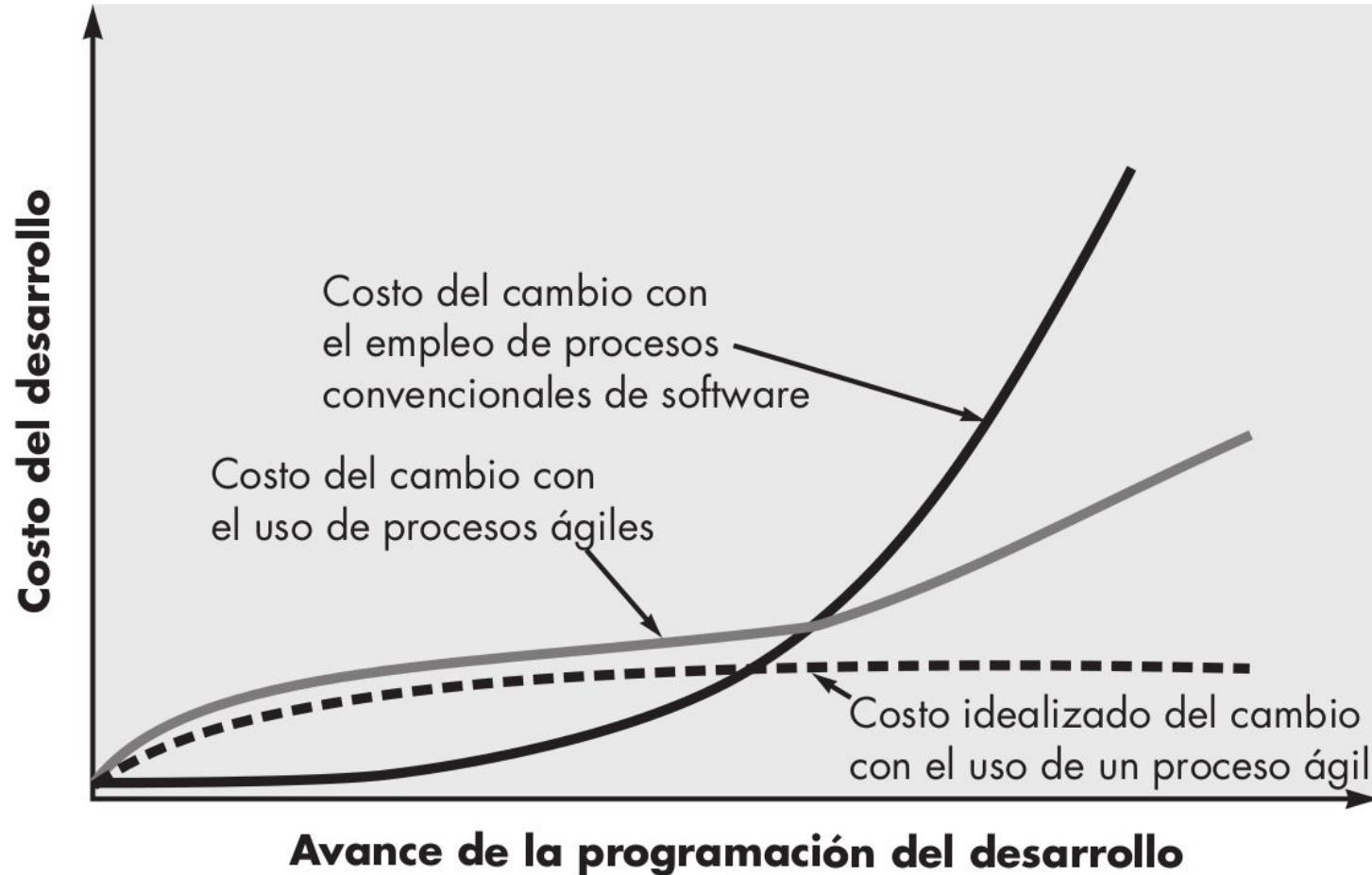
- Los procesos prescriptivos **no** toman en cuenta las debilidades de los desarrolladores
  - Queremos procesos que impartan más disciplina o más tolerancia?

# ¿Qué es la agilidad?

- Hacer más fácil la comunicación entre stakeholders
- Desenfatar los productos intermedios
- Incluir al cliente en el equipo (eliminar el nosotros/ellos)
- Reconocer la incertidumbre de un plan
- Enfatizar una entrega incremental
- Aceptar el cambio
- Y más...



# Costo del cambio



# ¿Qué es Scrum?

Es un framework, metodología, o proceso?

“Scrum es un [marco de trabajo](#) liviano que ayuda a las personas, equipos y organizaciones a generar valor a través de soluciones adaptativas para problemas complejos.”

[Sutherland, Jeff, and Ken Schwaber. "La guía de Scrum.", (2020), pag. 3]

- Aparece a principios de los años 90s (1993)
- Para desarrollar y entregar productos complejos
- Basado en teoría empírica del control de procesos
- Se usa en otras áreas, no sólo para desarrollar SW

# Tres pilares en el control de procesos empírico

- **Transparencia:**

- Aspectos del procesos visibles para todos
- Entendimiento común

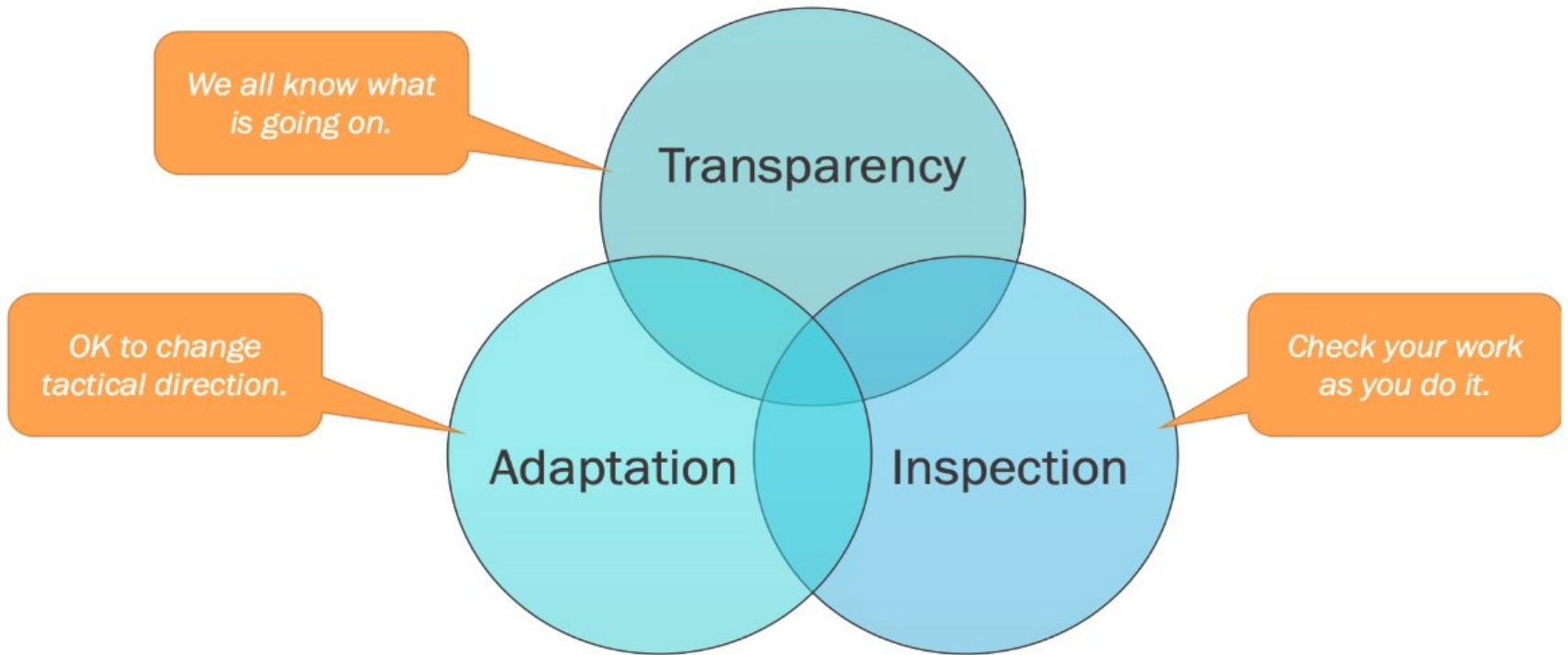
- **Inspección**

- Inspecciones frecuente del progreso realizada por cada uno
- Realizada con rapidez, no debe ser una carga

- **Adaptación**

- Ante una desviación detectada se hacen ajustes
- Prescribe cuatros formas de inspecciones formales:
  - Sprint planning
  - Daily Scrum
  - Sprint review
  - Sprint retrospective

# Los tres pilares del empirismo

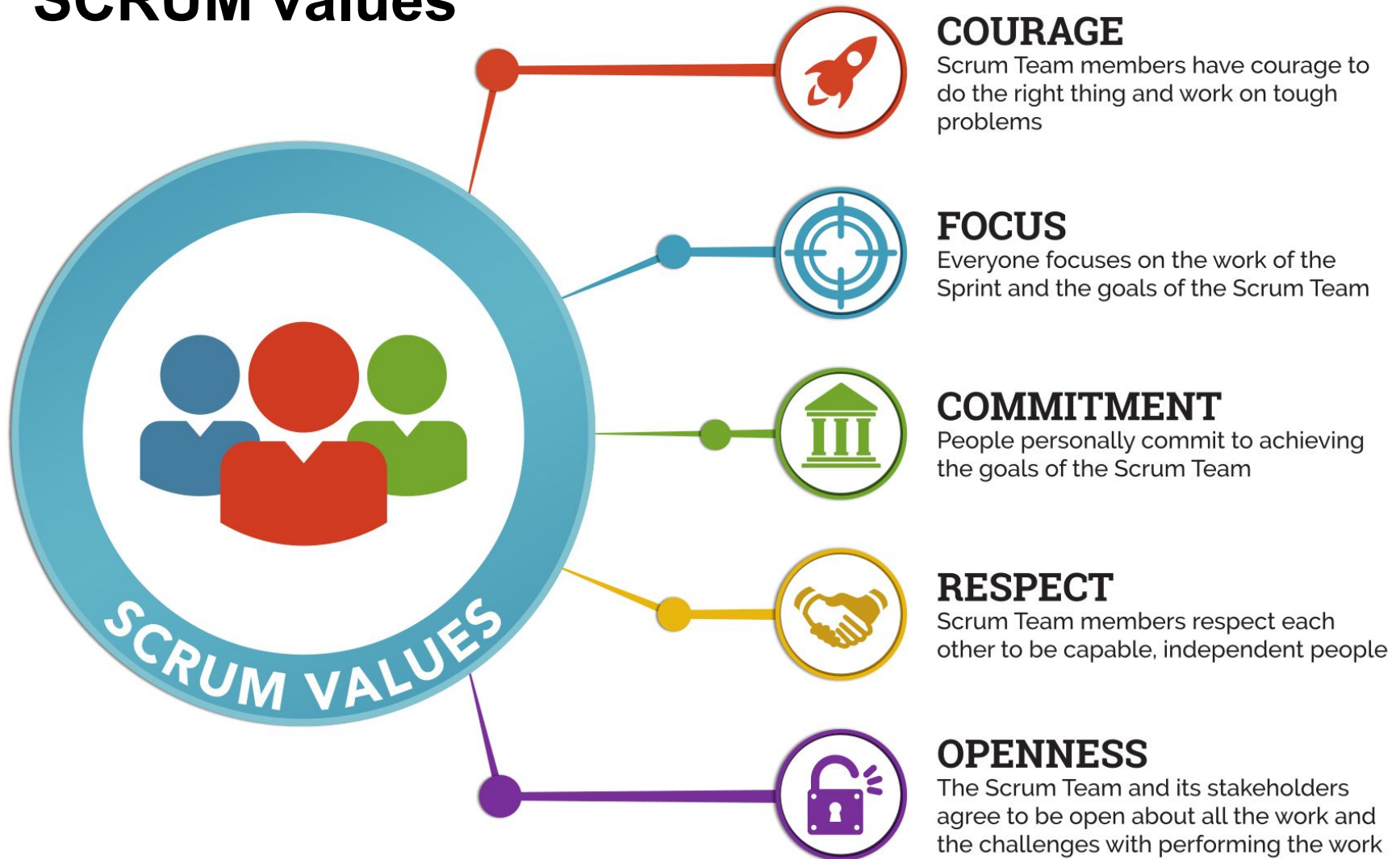


[<https://www.scrum.org/resources/blog/three-pillars-empiricism-scrum>]

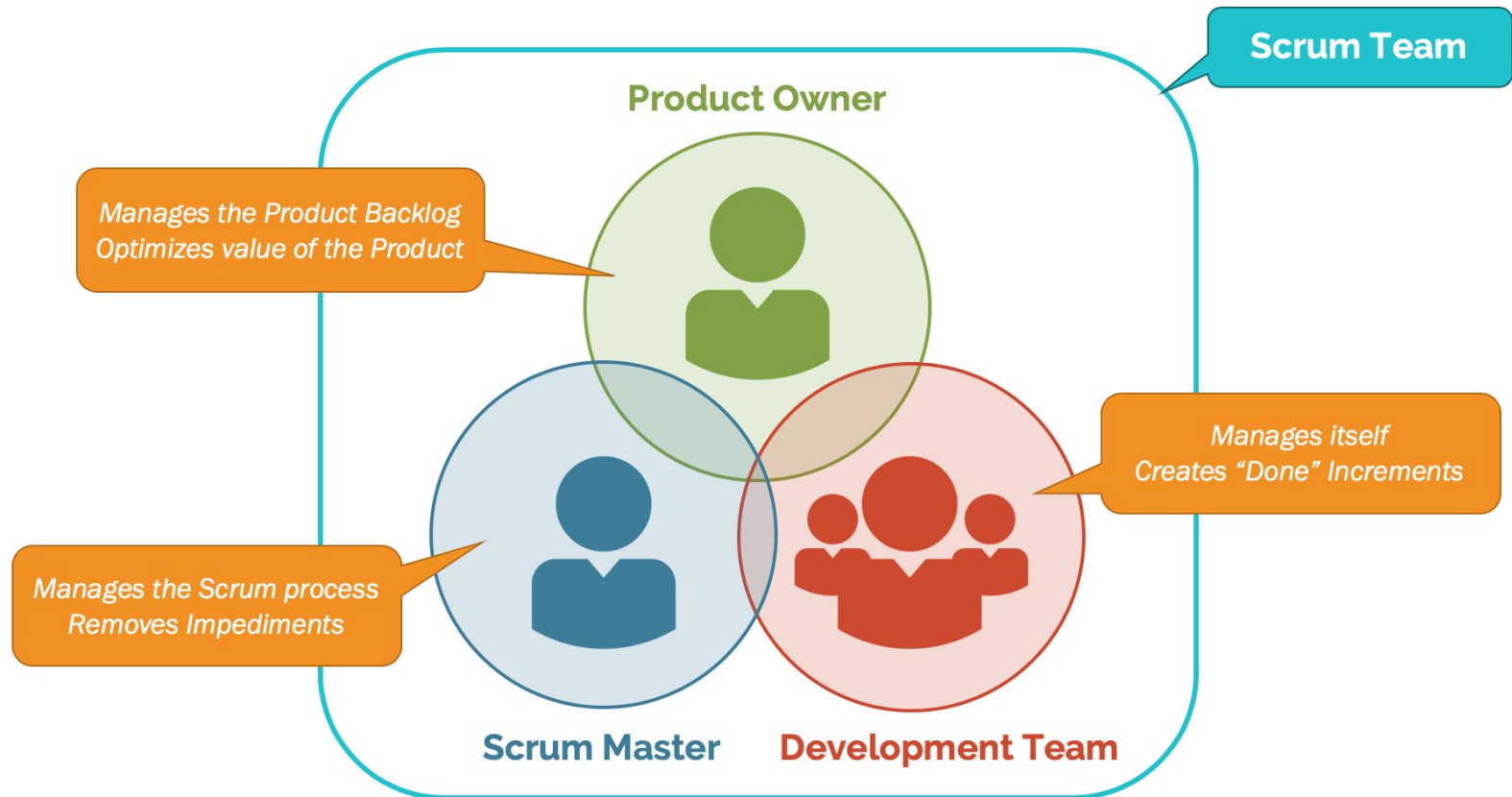
# Valores de SCRUM

- Compromiso
  - Las personas se comprometen a alcanzar la meta
- Coraje
  - de informar buenas y malas noticias
  - de hacer lo correcto y trabajar en problemas difíciles
- Foco
  - para revisar constantemente las actividades
  - para alcanzar las metas del sprint
- Apertura
  - a todo trabajo que se presente
- Respeto
  - al otro para ser personas capaces e independientes

# SCRUM values



# SCRUM Team



# SCRUM roles: Product owner

- El dueño del producto (Product owner)
  - Define las funcionalidades del producto
  - Decide sobre las fechas y contenidos de los releases
  - Es responsable por la rentabilidad del producto (ROI)
  - Prioriza funcionalidades de acuerdo al valor del mercado/negocio
  - Ajusta funcionalidades y prioridades en cada iteración cuando es necesario



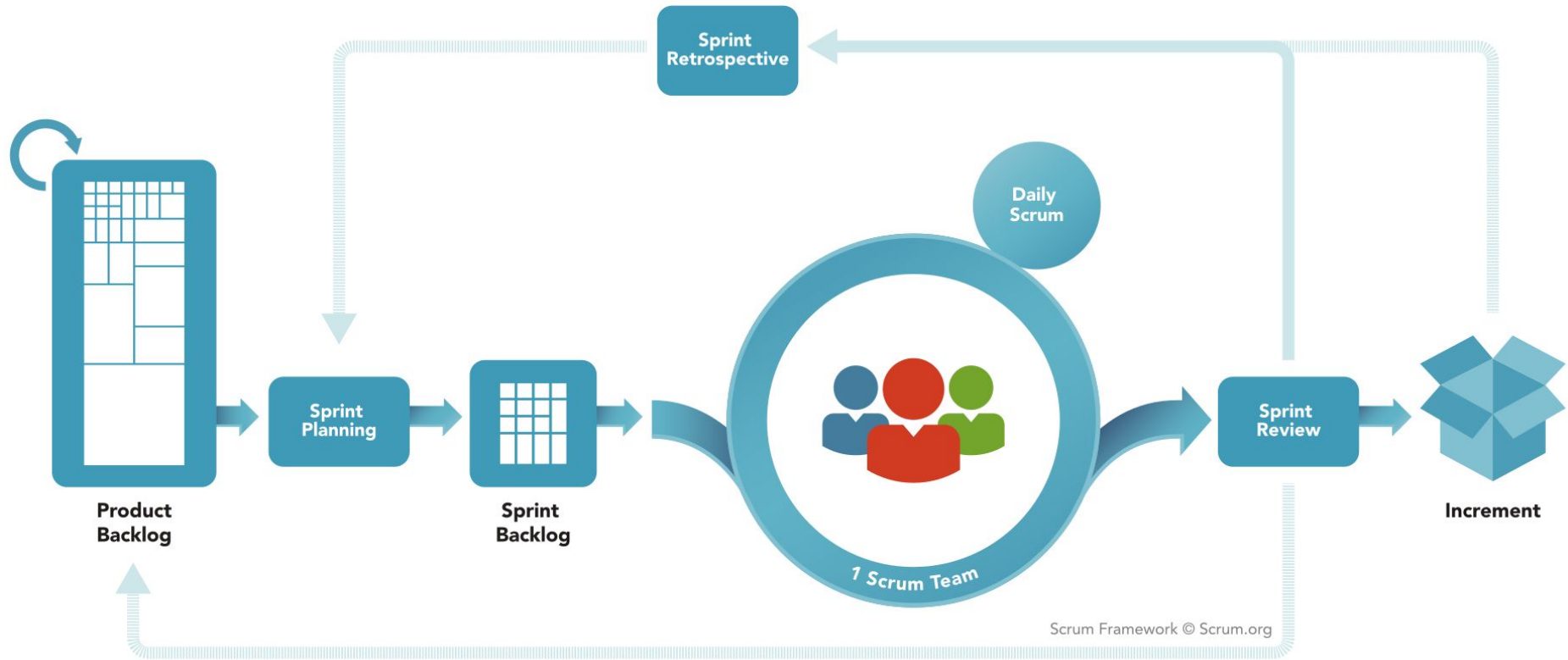
# SCRUM roles: **Development Team**

- El equipo de desarrollo (Development Team)
- Típicamente de 5 a 9 personas
  - Multi-funcional:
    - Programadores, testers, analistas, diseñadores, etc.
  - Los miembros tienen que reportar su dedicación
- Los equipos son auto-organizativos
  - Idealmente, no existen títulos pero a veces se utilizan de acuerdo a la organización
  - Solo puede haber cambio de miembros entre los sprints

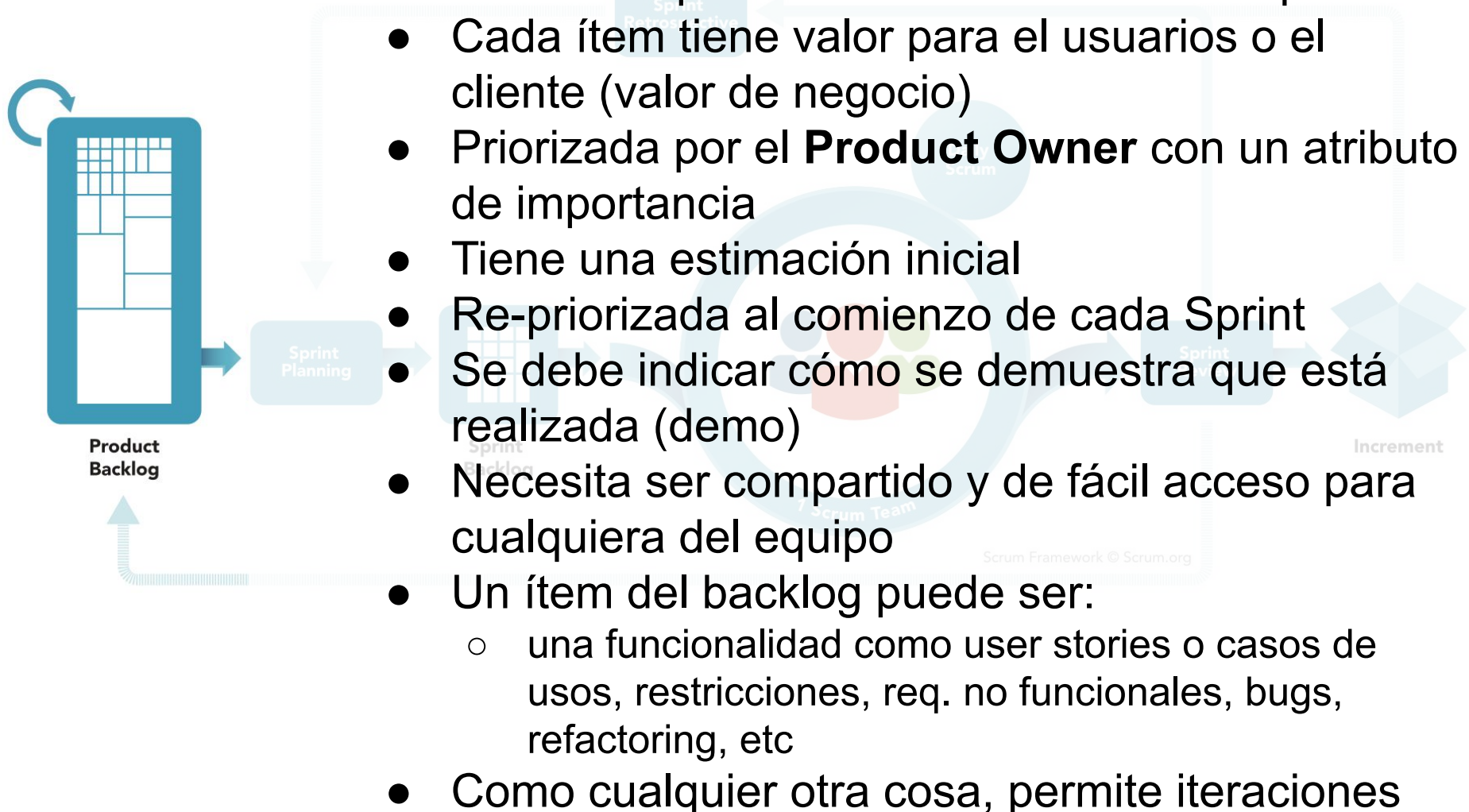
## **SCRUM roles: SCRUM Master**

- Representa a la gestión del proyecto
- Responsable de promover los valores y prácticas de Scrum
- Remueve impedimentos
- Se asegura de que el equipo es completamente funcional y productivo
- Permite la estrecha cooperación en todos los roles y funciones
- Escudo del equipo de interferencias externas

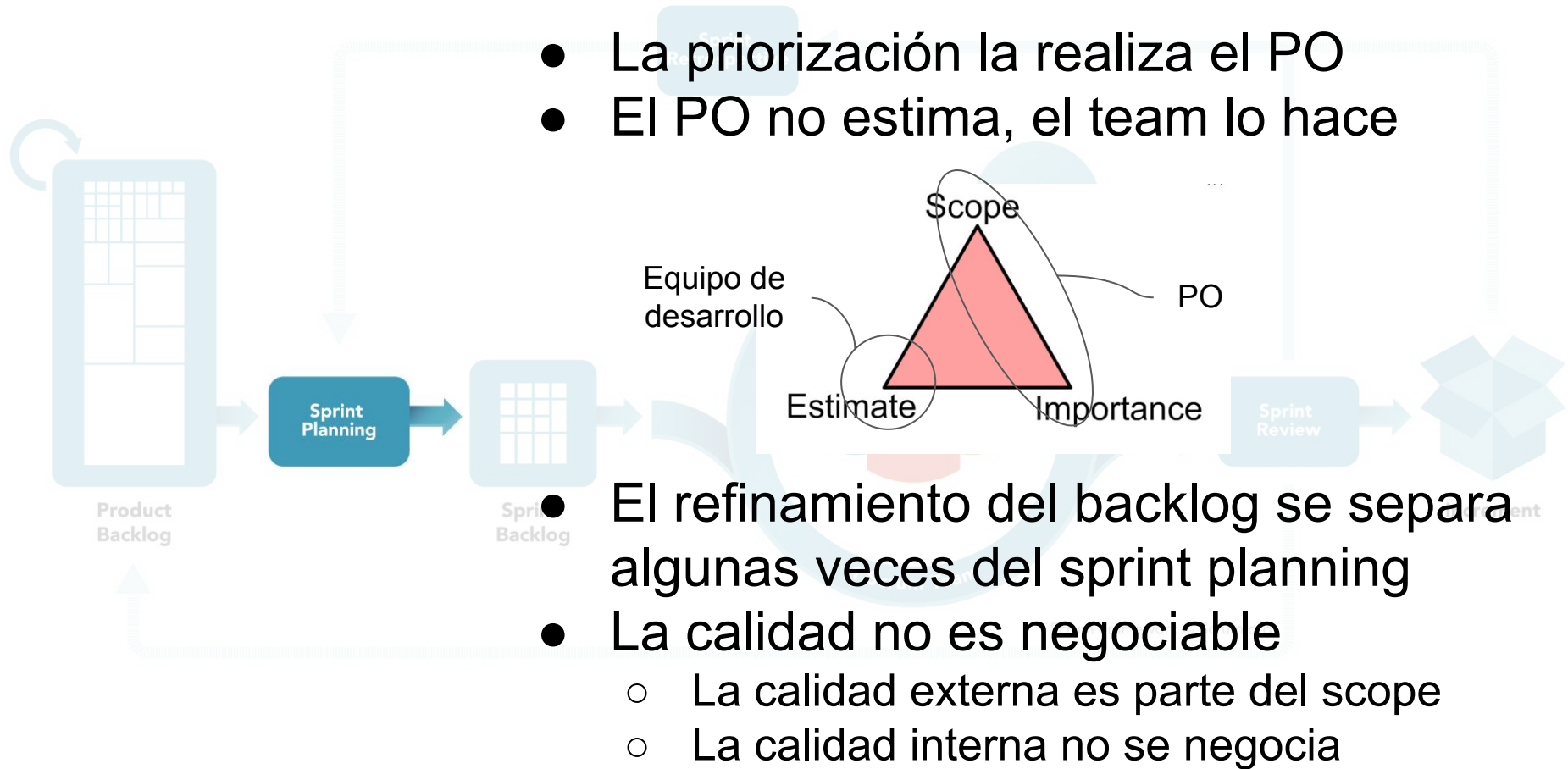
# SCRUM framework



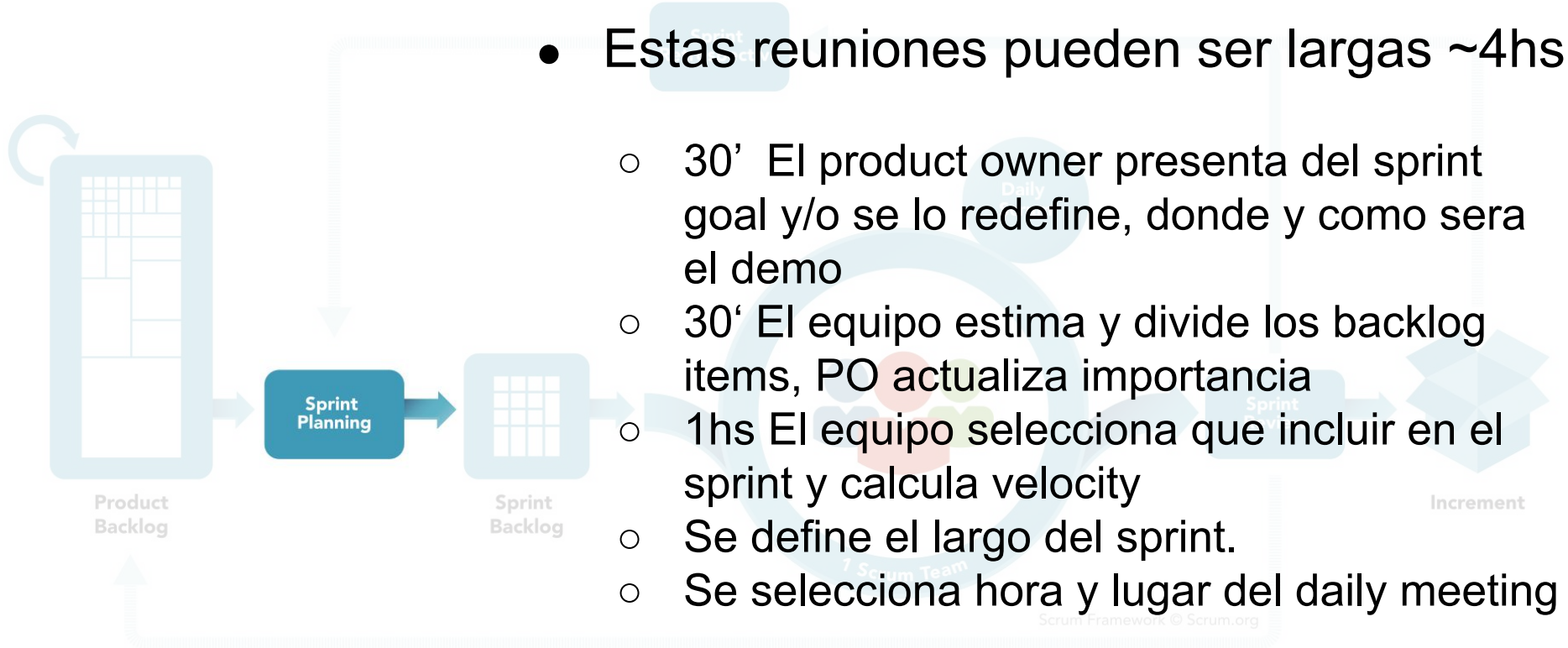
# SCRUM artifact: **product backlog**



# SCRUM event: **Sprint planning (1/3)**



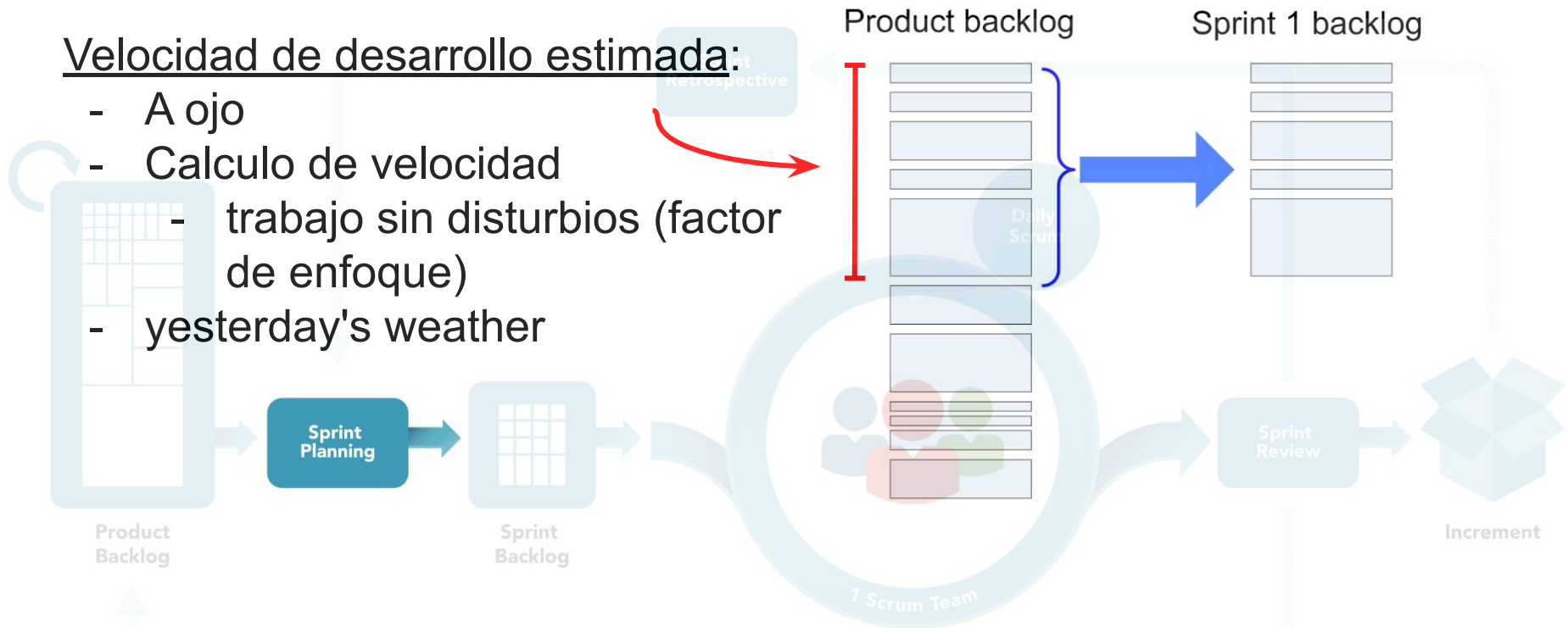
# SCRUM event: **Sprint planning (2/3)**



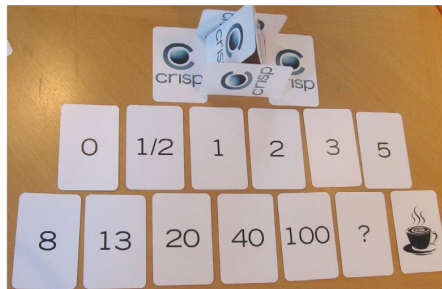
# SCRUM event: Sprint planning (3/3)

## Velocidad de desarrollo estimada:

- A ojo
- Calculo de velocidad
  - trabajo sin disturbios (factor de enfoque)
- yesterday's weather



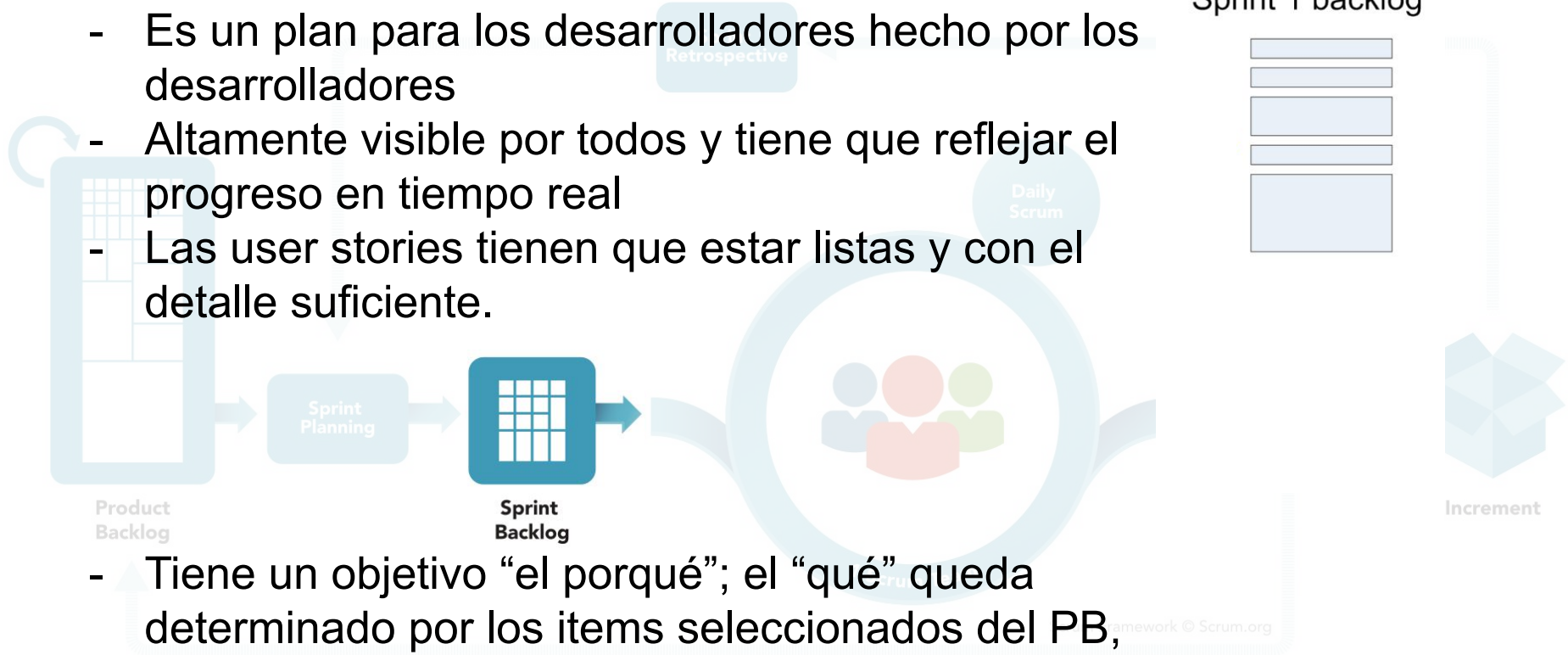
- Planning poker



- El equipo decide cuántas stories incluir, no el PO

# SCRUM artifact: Sprint backlog (1/2)

- Es un plan para los desarrolladores hecho por los desarrolladores
- Altamente visible por todos y tiene que reflejar el progreso en tiempo real
- Las user stories tienen que estar listas y con el detalle suficiente.

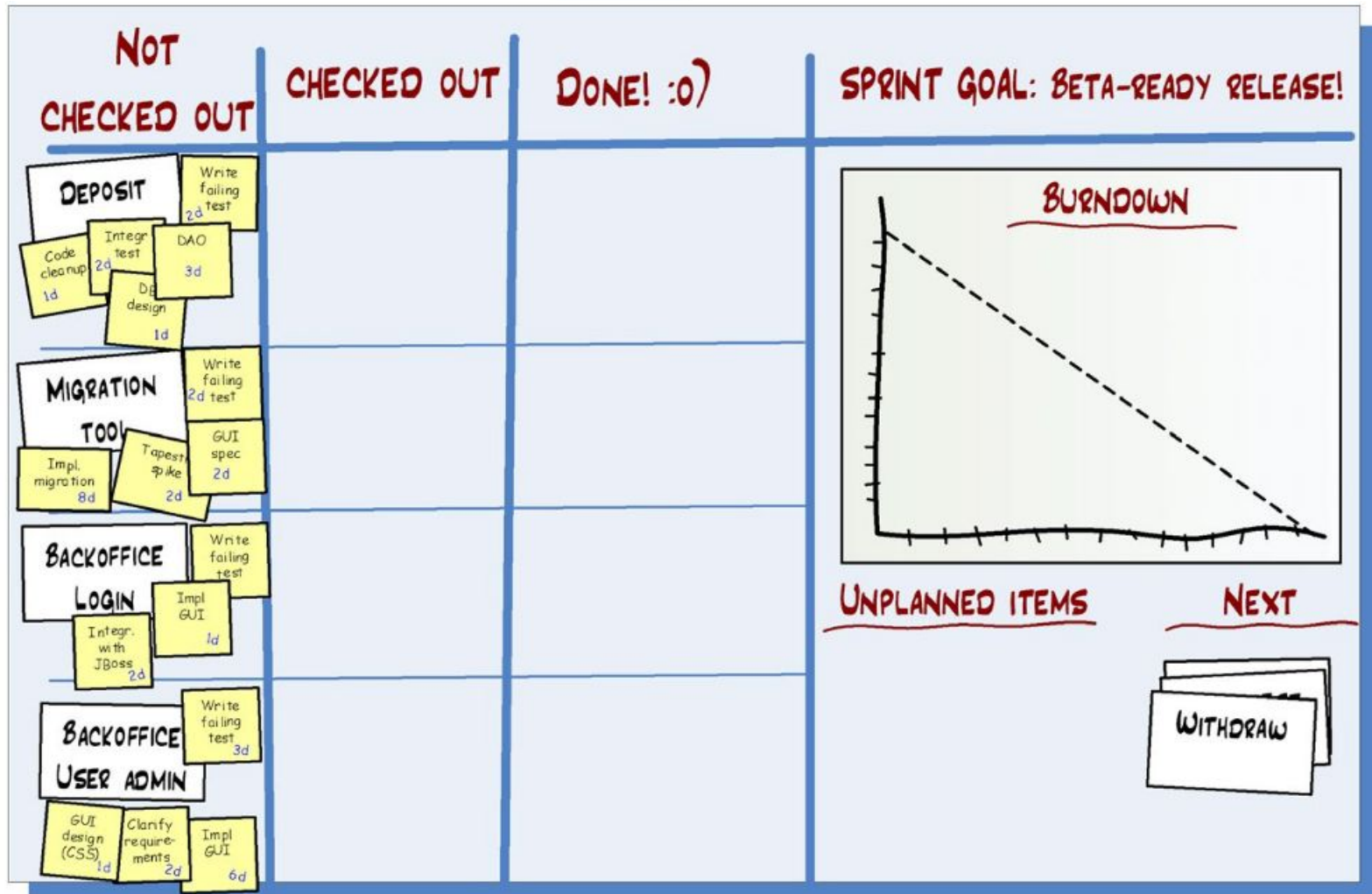


- Tiene un objetivo “el porqué”; el “qué” queda determinado por los items seleccionados del PB, y un “cómo” que es un plan de entrega del incremento.

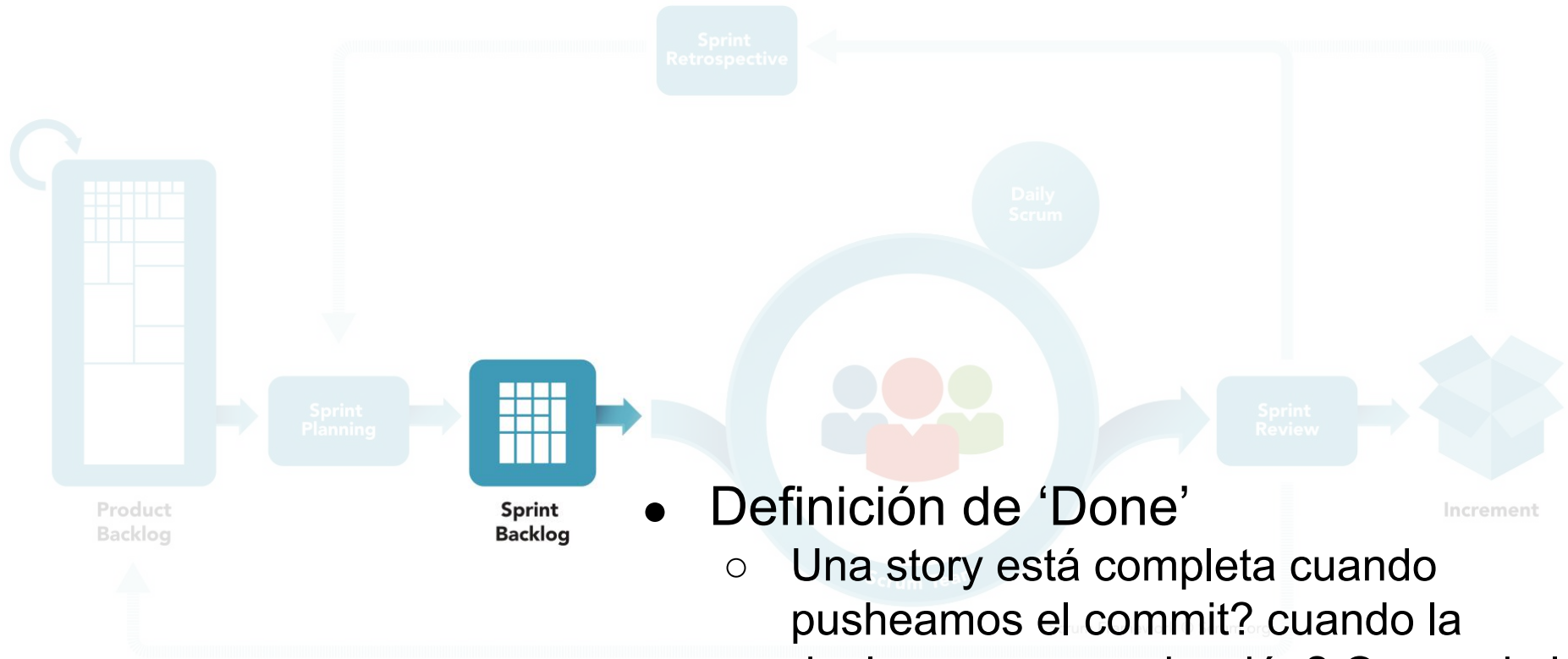
- Se usan fichas en un tablero kanban  
-> ver siguiente



# Tablero kanban + burndown chart



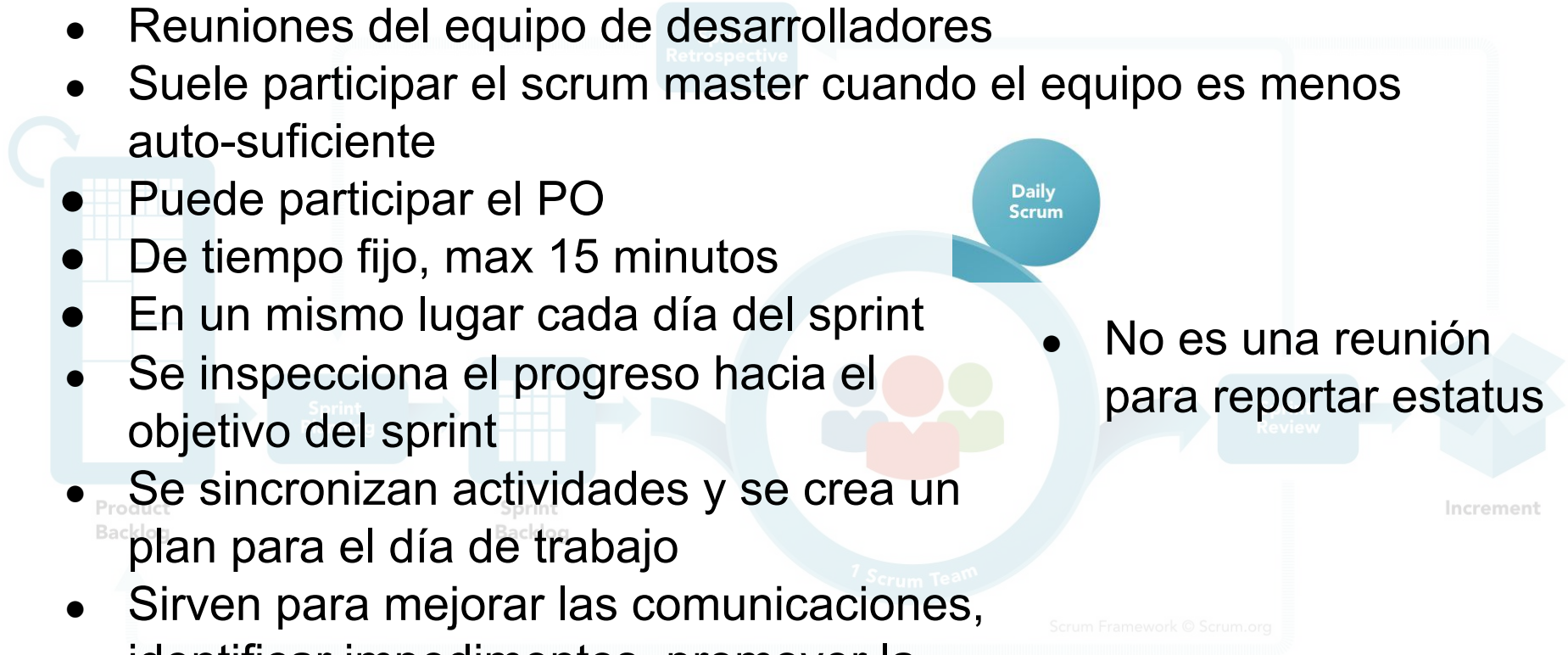
# SCRUM artifact: Sprint backlog (2/2)



- Definición de 'Done'
  - Una story está completa cuando pusheamos el commit? cuando la deployamos a producción? O cuando la deployamos para un test de aceptación? etc
- Si hay confusión, una historia puede tener su propia DOD

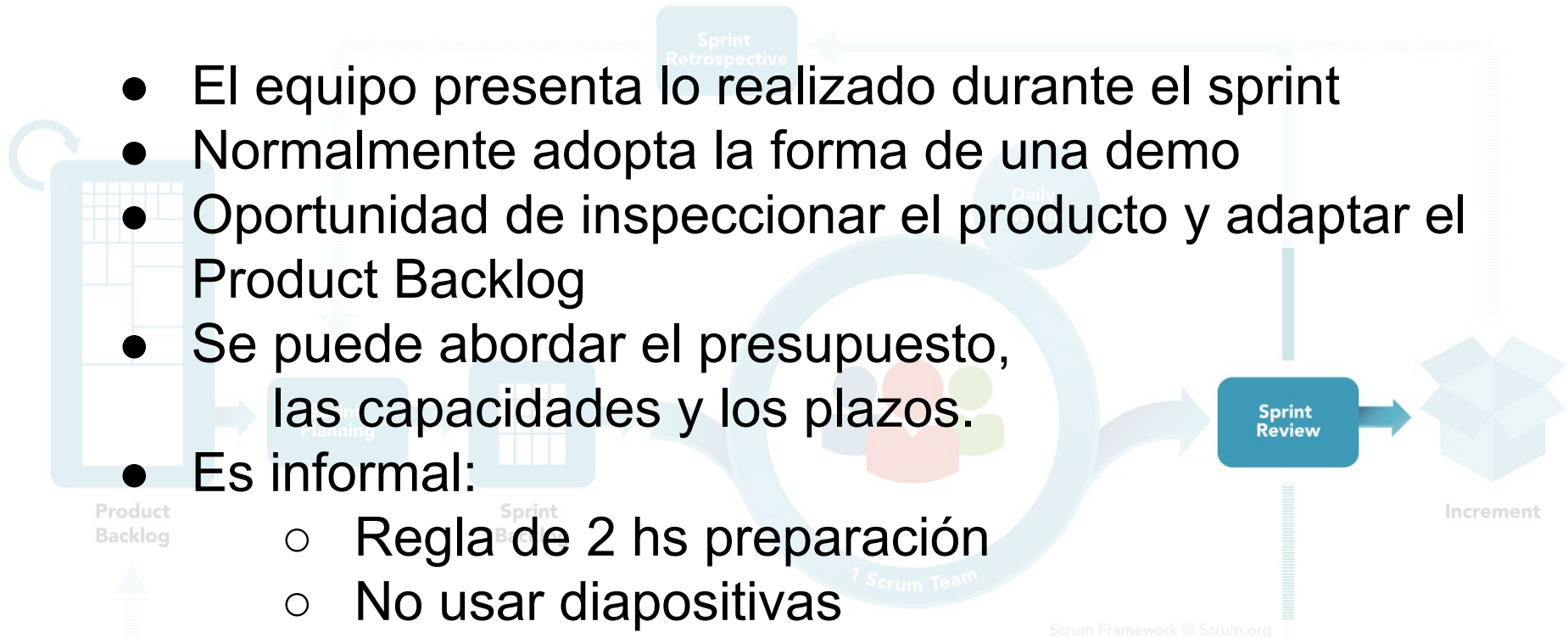
# SCRUM event: **daily scrum**

- Reuniones del equipo de desarrolladores
- Suele participar el scrum master cuando el equipo es menos auto-suficiente
- Puede participar el PO
- De tiempo fijo, max 15 minutos
- En un mismo lugar cada día del sprint
- Se inspecciona el progreso hacia el objetivo del sprint
- Se sincronizan actividades y se crea un plan para el día de trabajo
- Sirven para mejorar las comunicaciones, identificar impedimentos, promover la toma de decisiones rápidas, para adaptar o replanificar el resto del trabajo
- No es una reunión para reportar estatus

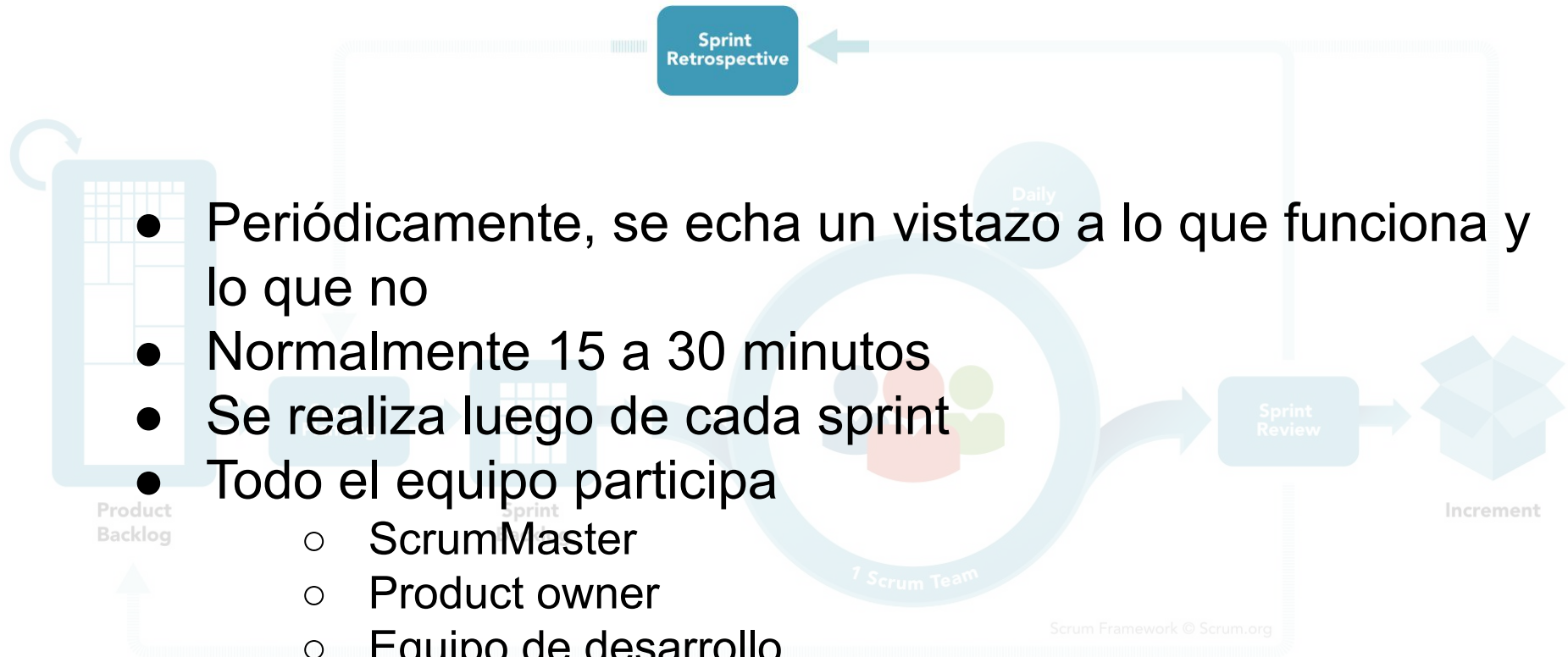


# SCRUM event: **Sprint review**

- El equipo presenta lo realizado durante el sprint
- Normalmente adopta la forma de una demo
- Oportunidad de inspeccionar el producto y adaptar el Product Backlog
- Se puede abordar el presupuesto, las capacidades y los plazos.
- Es informal:
  - Regla de 2 hs preparación
  - No usar diapositivas
- Todo el mundo participa: PO, el equipo Scrum, la gerencia, los clientes y es posible que participen los desarrolladores de otros proyectos.
- Sin cliente, no hay review



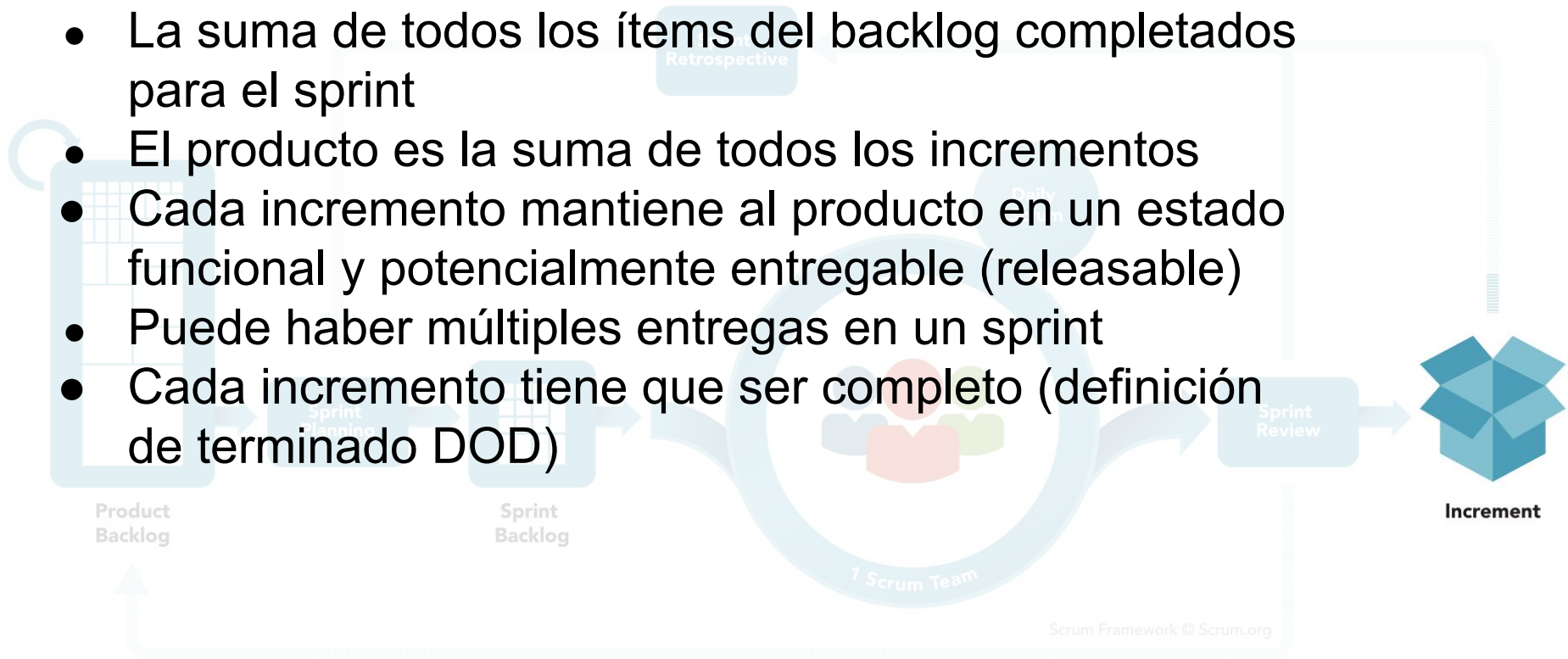
# SCRUM event: **sprint retrospective**



- Periódicamente, se echa un vistazo a lo que funciona y lo que no
- Normalmente 15 a 30 minutos
- Se realiza luego de cada sprint
- Todo el equipo participa
  - ScrumMaster
  - Product owner
  - Equipo de desarrollo
  - Posiblemente puede participar algún stakeholder
- Todo el equipo se reúne y discute lo que hay que:
  - Comenzar a hacer, Dejar de hacer, Continuar haciendo

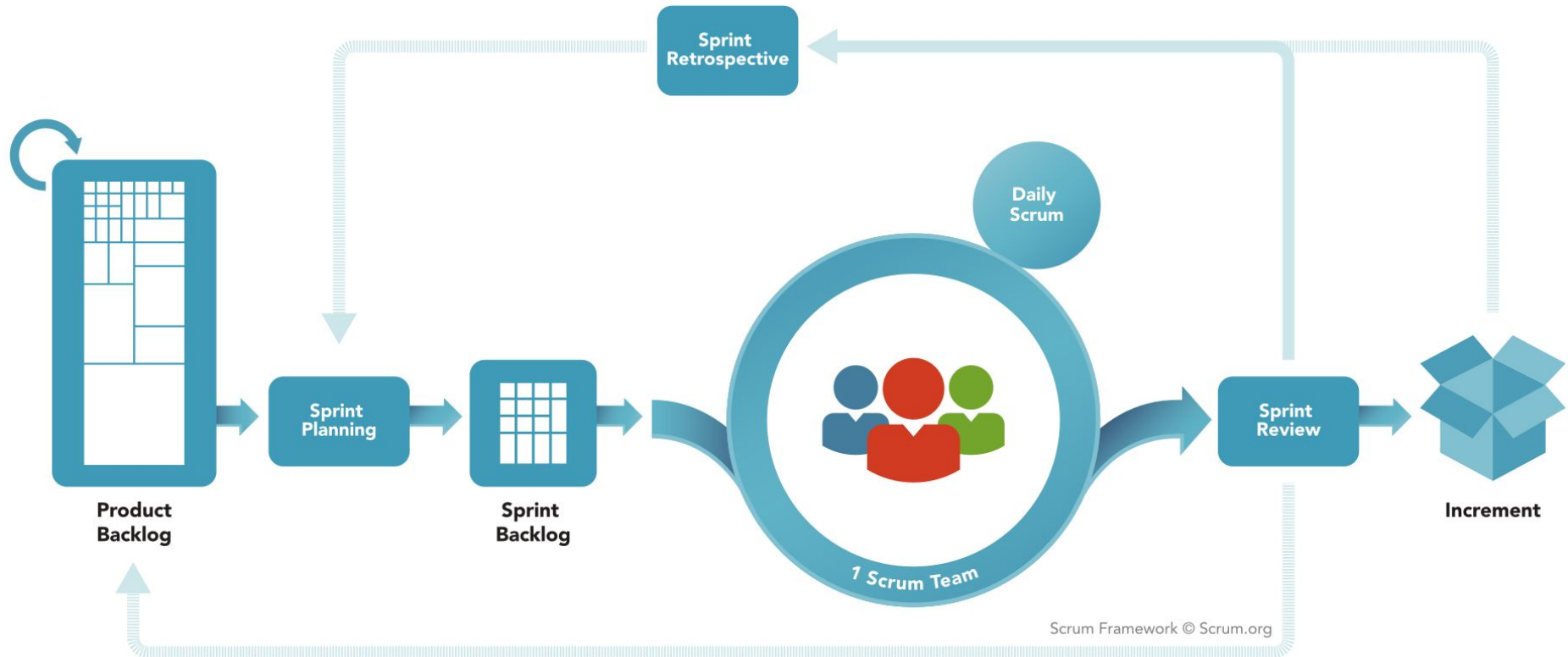
# SCRUM artifact: **incremento**

- La suma de todos los ítems del backlog completados para el sprint
- El producto es la suma de todos los incrementos
- Cada incremento mantiene al producto en un estado funcional y potencialmente entregable (releasable)
- Puede haber múltiples entregas en un sprint
- Cada incremento tiene que ser completo (definición de terminado DOD)



# SCRUM event: **sprint**

- En un contenedor de todas las actividades y todos los eventos scrum



- Sirve para enfocarse en desarrollar actividades
- Comienza con el sprint planning
- Termina con el sprint retrospective
- de 30 días o menos de duración para tener una retroalimentación regular



# SCRUM framework

- **Roles**

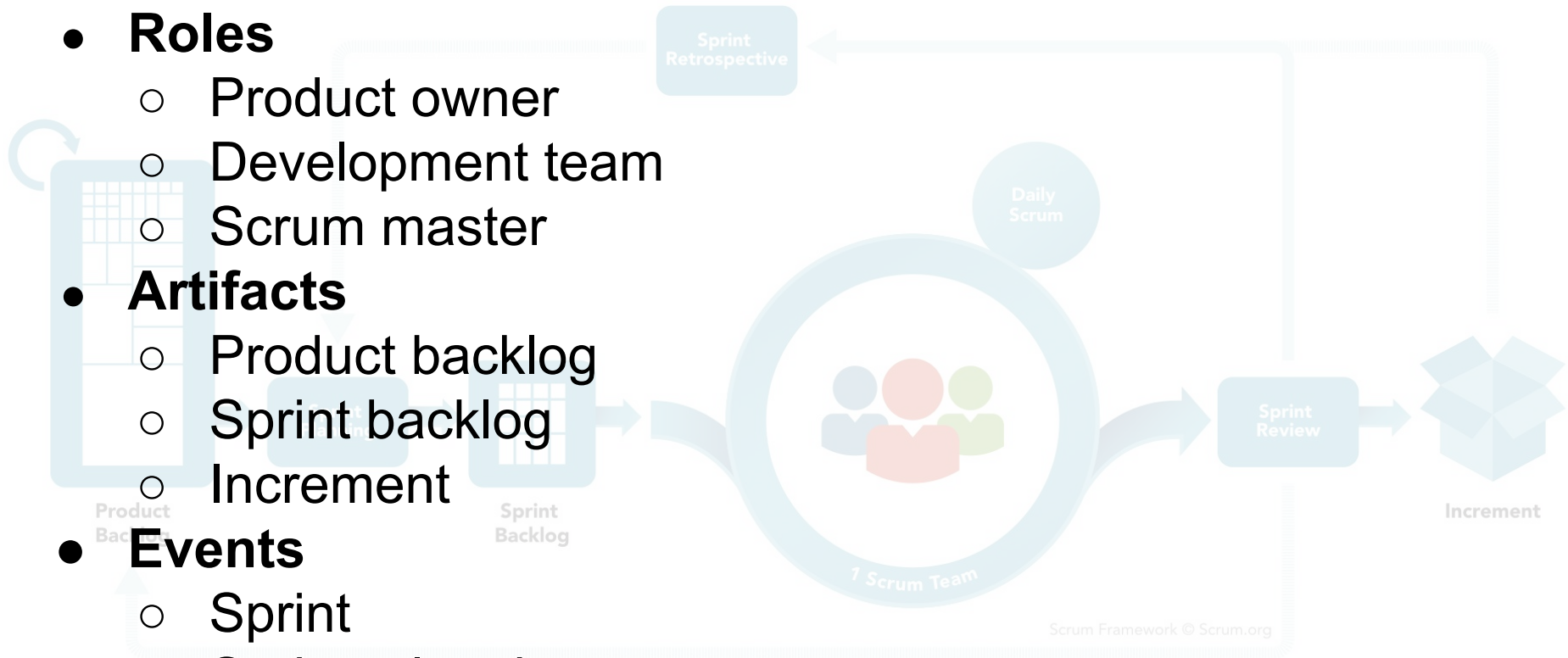
- Product owner
- Development team
- Scrum master

- **Artifacts**

- Product backlog
- Sprint backlog
- Increment

- **Events**

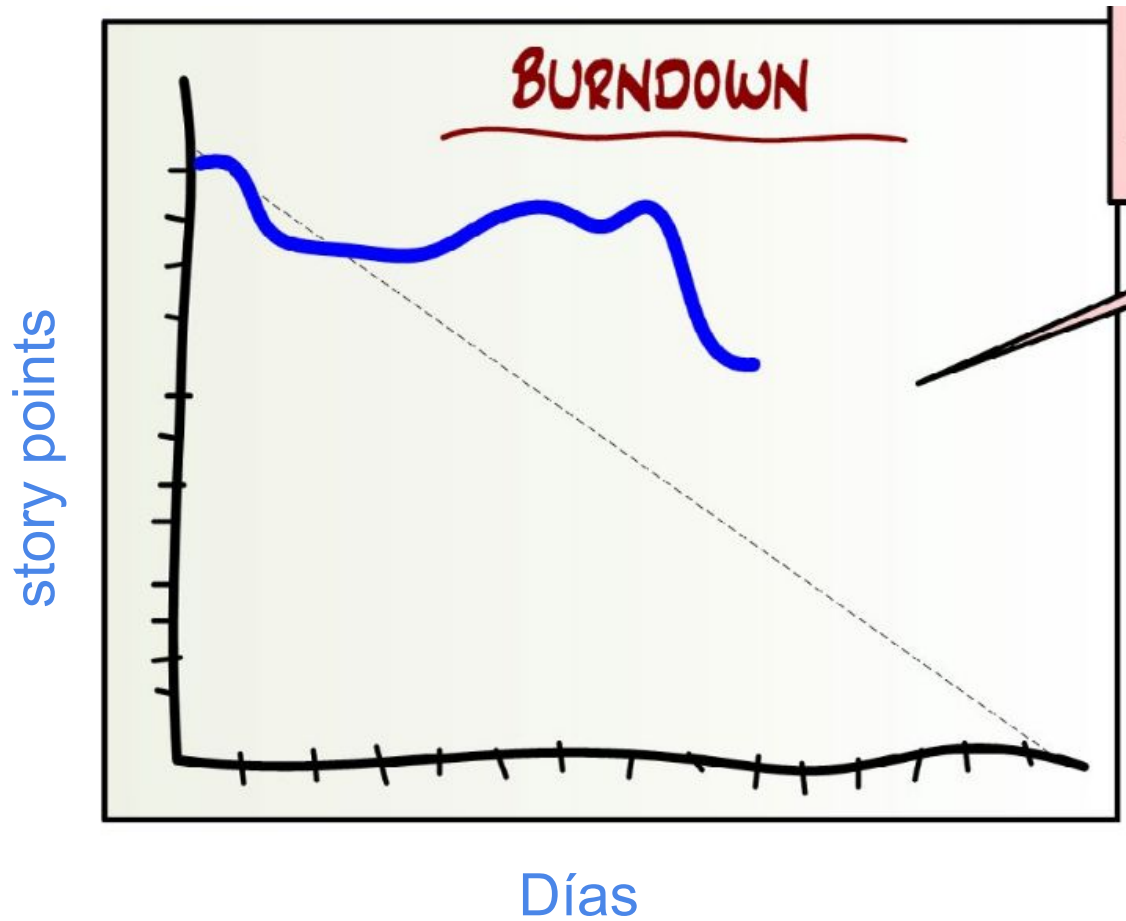
- Sprint
- Sprint planning
- Daily scrum
- Sprint review
- Sprint retrospective



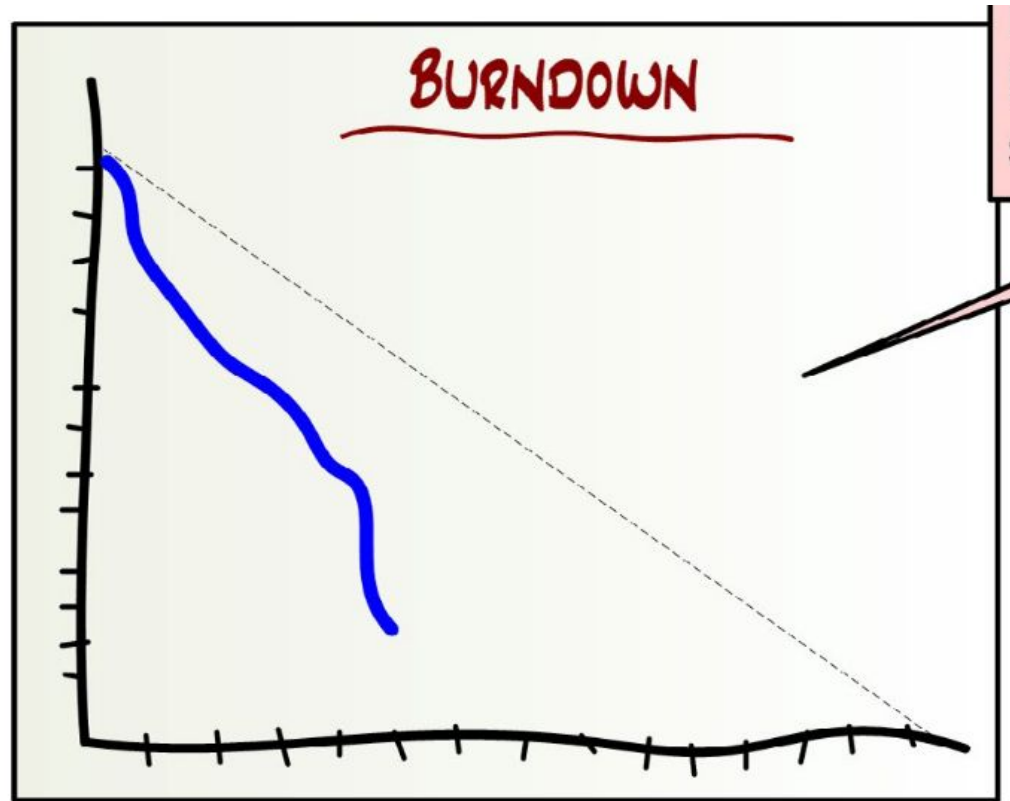


¿Cómo se limita el riesgo con los elementos que define el framework Scrum?

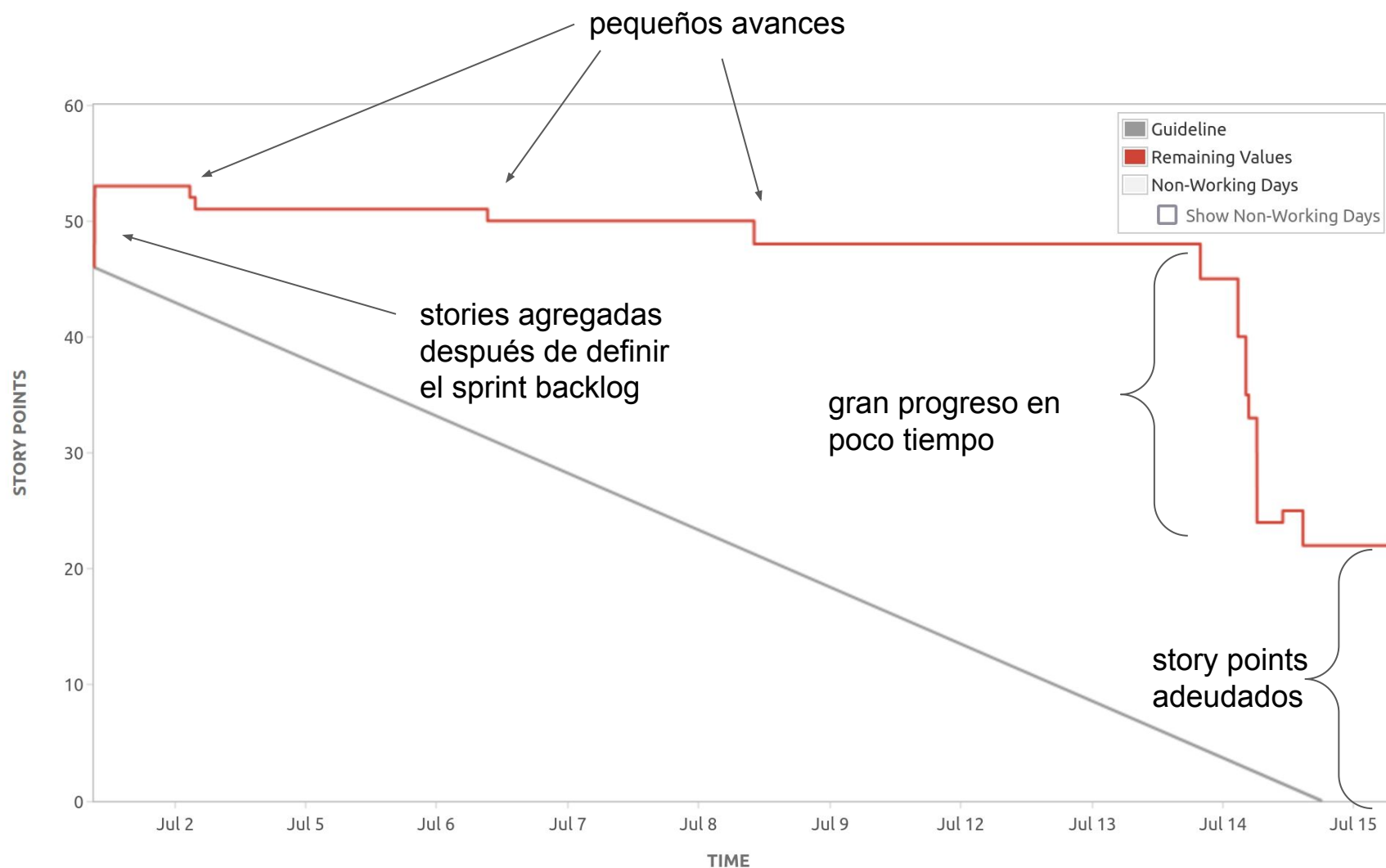
# Burndown chart y velocidad



# Burndown chart y velocidad

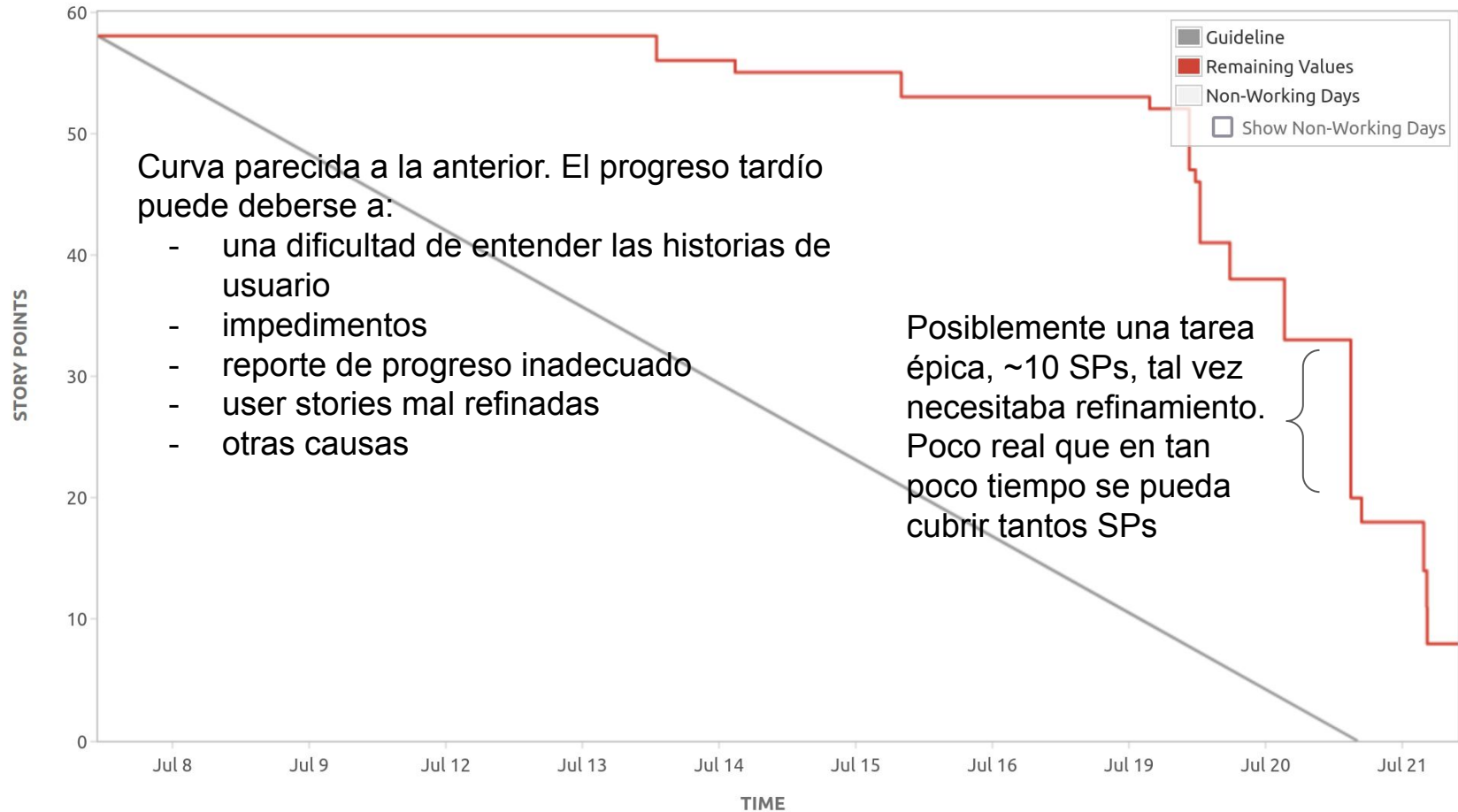


# Burndown chart y velocidad: ejemplo real



entre dos  
semanas y dos meses,

# Burndown chart y velocidad: ejemplo real



# Burndown chart y velocidad: ejemplo real



Tareas cumplidas fuera del sprint, tal vez por los problemas del planning. Esas tareas debieron pasar al sprint siguiente

¿Podemos considerar a Scrum un SDLC?

# Bibliografía

Sutherland, Jeff, and Ken Schwaber. "The scrum guide." The definitive guide to scrum: The rules of the game. Scrum. org 268 (2013).

<https://www.scrum.org/resources/what-is-scrum>

Kniberg, Henrik. "Scrum y XP desde las trincheras." Estados Unidos: C4Media (2007).

<https://www.mountaingoatsoftware.com/agile/scrum>.

[www.agilemanifesto.org](http://www.agilemanifesto.org)