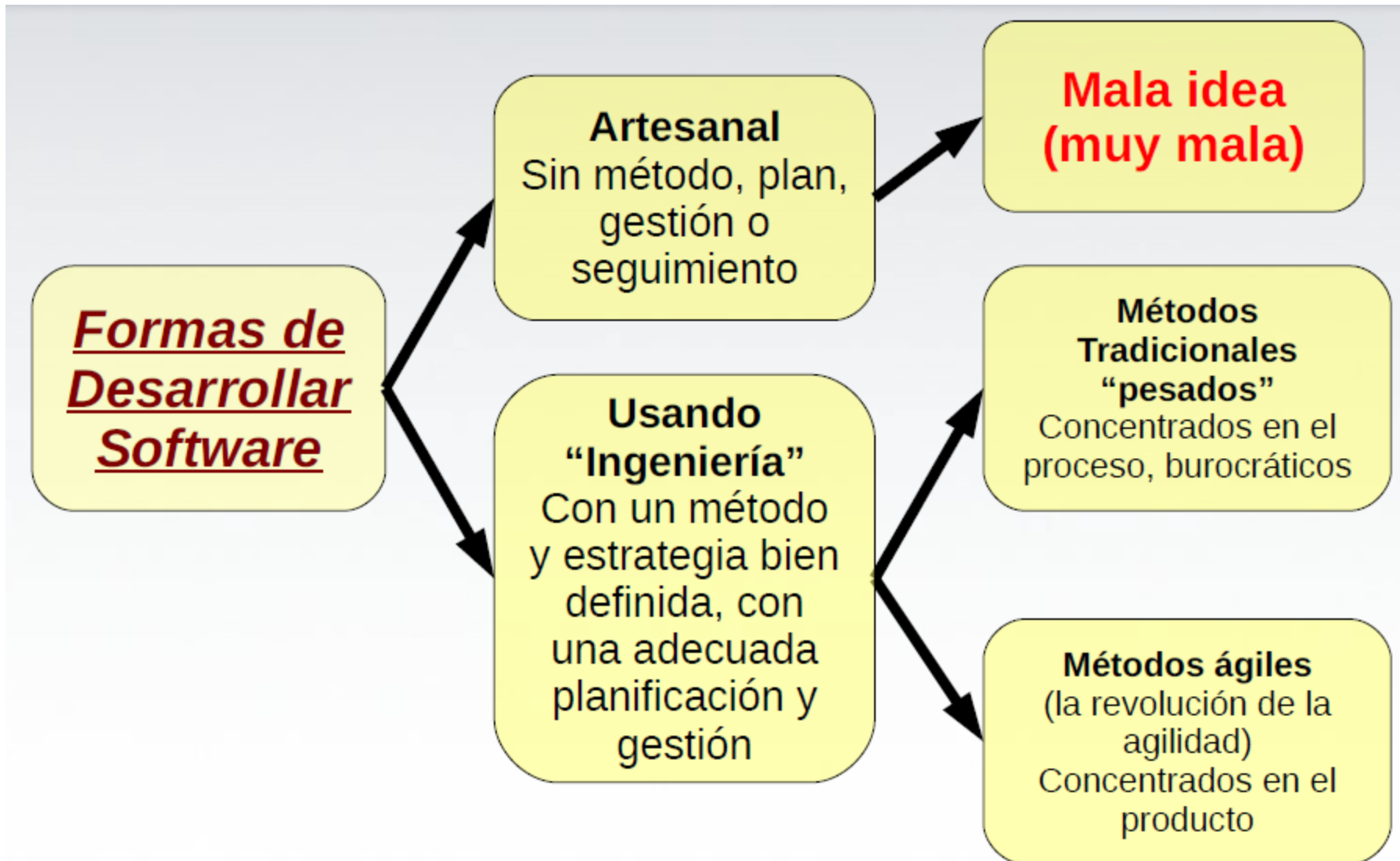


# SCRUM

MODELO DE DESARROLLO

# Formas de desarrollar software



# Métodos Tradicionales o Pesados

**Jens Østergaard,  
a fully qualified  
Certified Scrum  
Master Trainer,  
gives an introduction to  
Scrum and talks about  
why is Scrum so hard**

<http://www.youtube.com/watch?v=q3t8twm3aUk>

# modelos ágiles (SCRUM)

# Principios de SCRUM

## **la naturaleza del software es el cambio**

*“Dado que los cambios nunca van a dejar de existir, lo que necesitamos es ser capaces de gestionar los proyectos de una forma más ágil”*

# Principios de SCRUM

*Japón, años 80*

Takeuchi y Nonaka

***basado en las mejores prácticas de las  
empresas con buenos resultados de  
rapidez y flexibilidad en la producción  
de la época***

Xerox, Canon, Honda, NEC, Epson,  
Brother, 3M, Hewlet Packard

# Principios de SCRUM

inicia en el ámbito de desarrollo de  
productos tecnológicos  
(no necesariamente software)  
pero se puede aplicar con éxito a otras  
áreas de gestión

# Principios de SCRUM

***Es mejor tener equipos pequeños y auto-organizados:***

*formados por miembros de diferentes disciplinas  
capaces de organizarse por si mismos  
con una comunicación transparente*

*En general, de esta forma se suelen obtener mejor resultados.*

*De esta forma, todos los miembros se comprometen y están motivados*

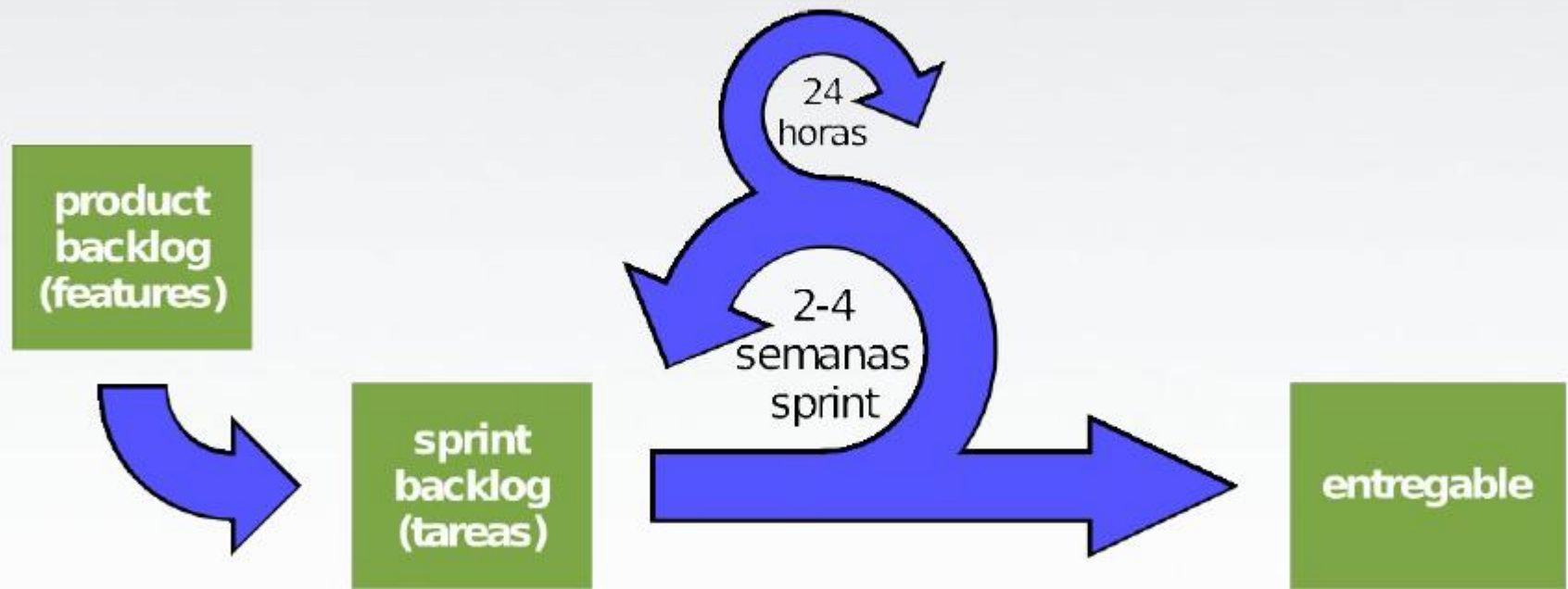


# Definición



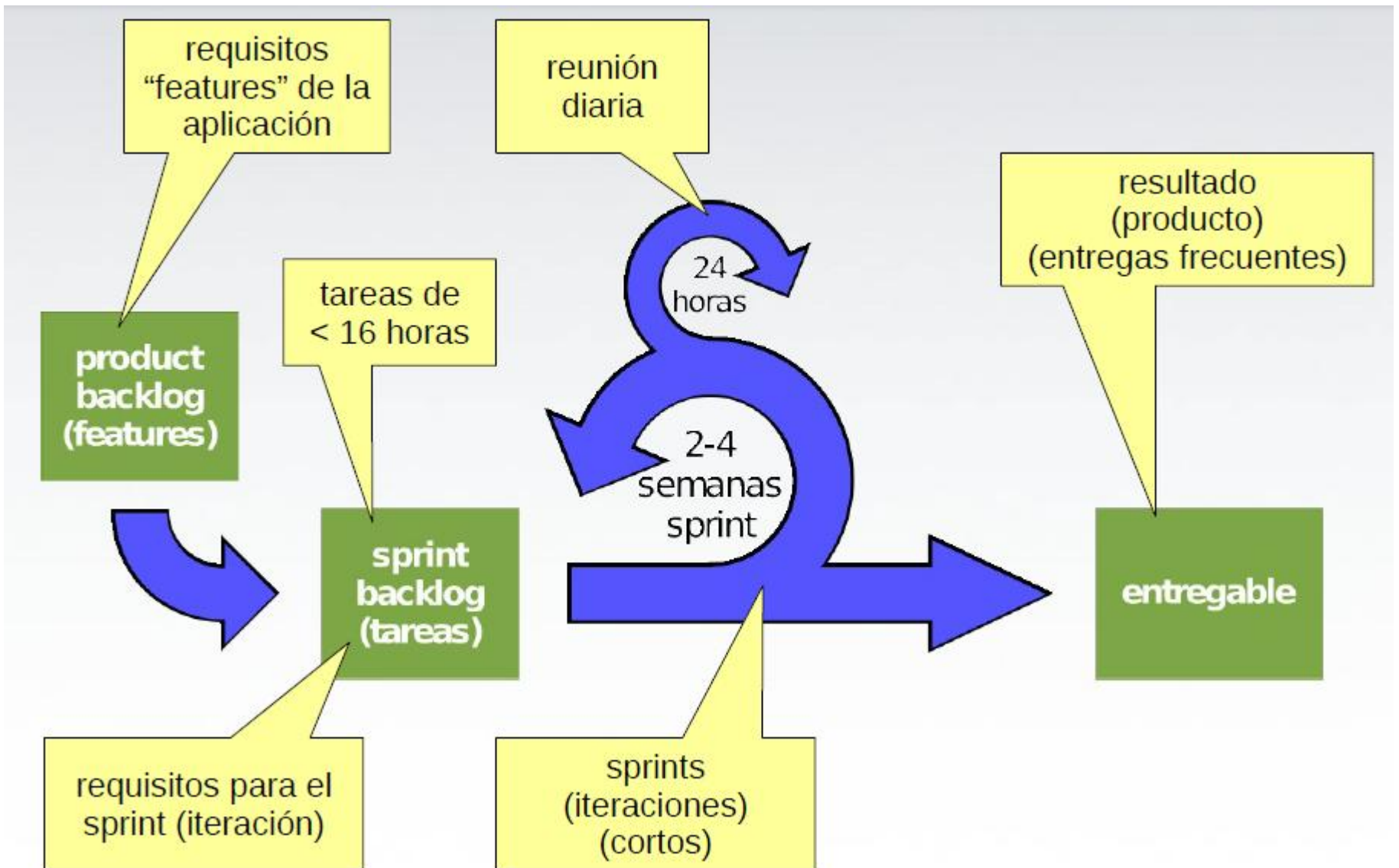
La palabra SCRUM procede del vocabulario del rugby y significa melé; es decir, esa “figura” en la que los compañeros del equipo se amontonan, forman una piña y empujan todos en la misma dirección

# Proceso

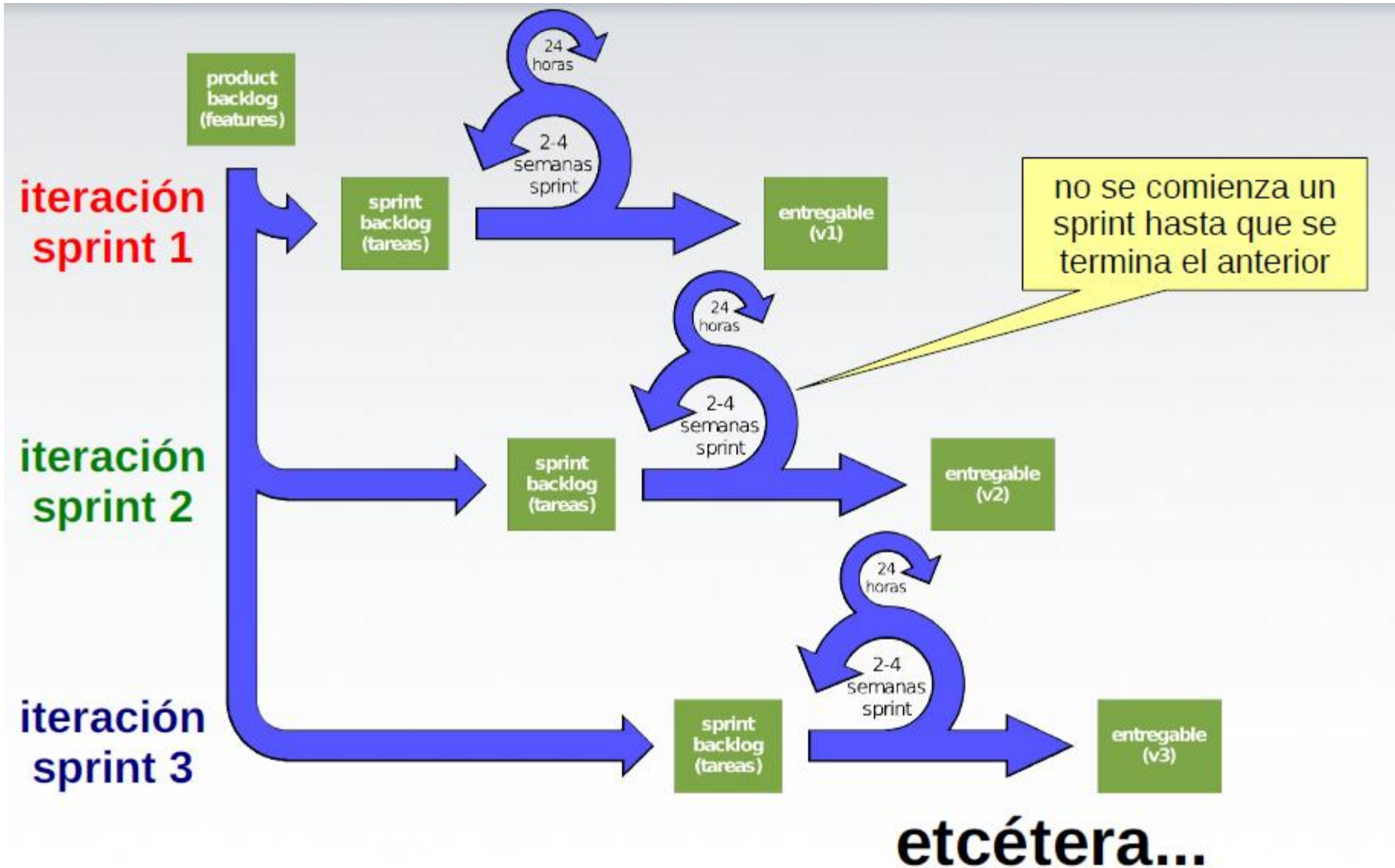


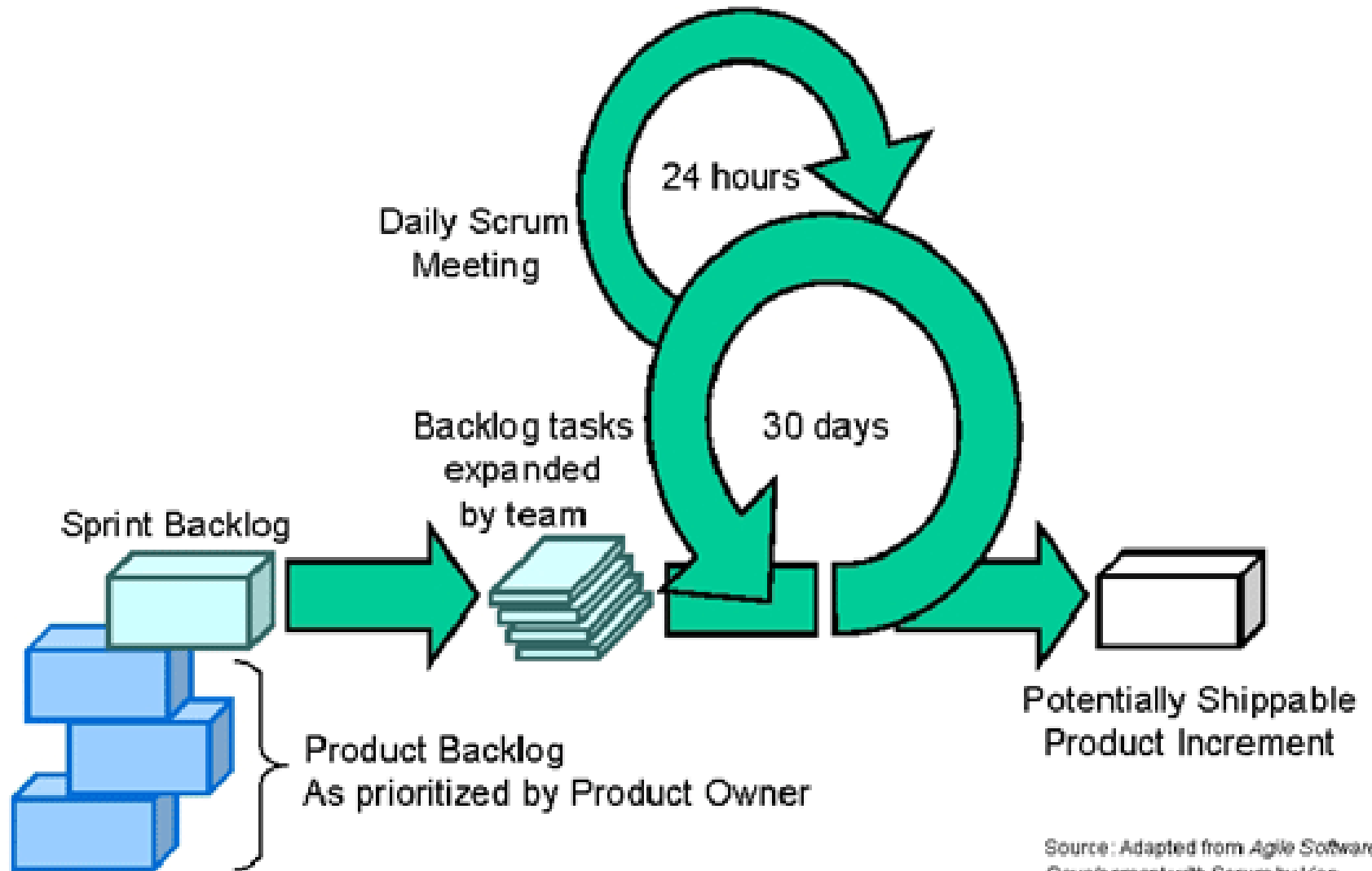
**¡SCRUM!**

# Proceso



# Proceso / SCRUM





Source: Adapted from *Agile Software Development with Scrum* by Ken Schwaber and Mike Beedle.



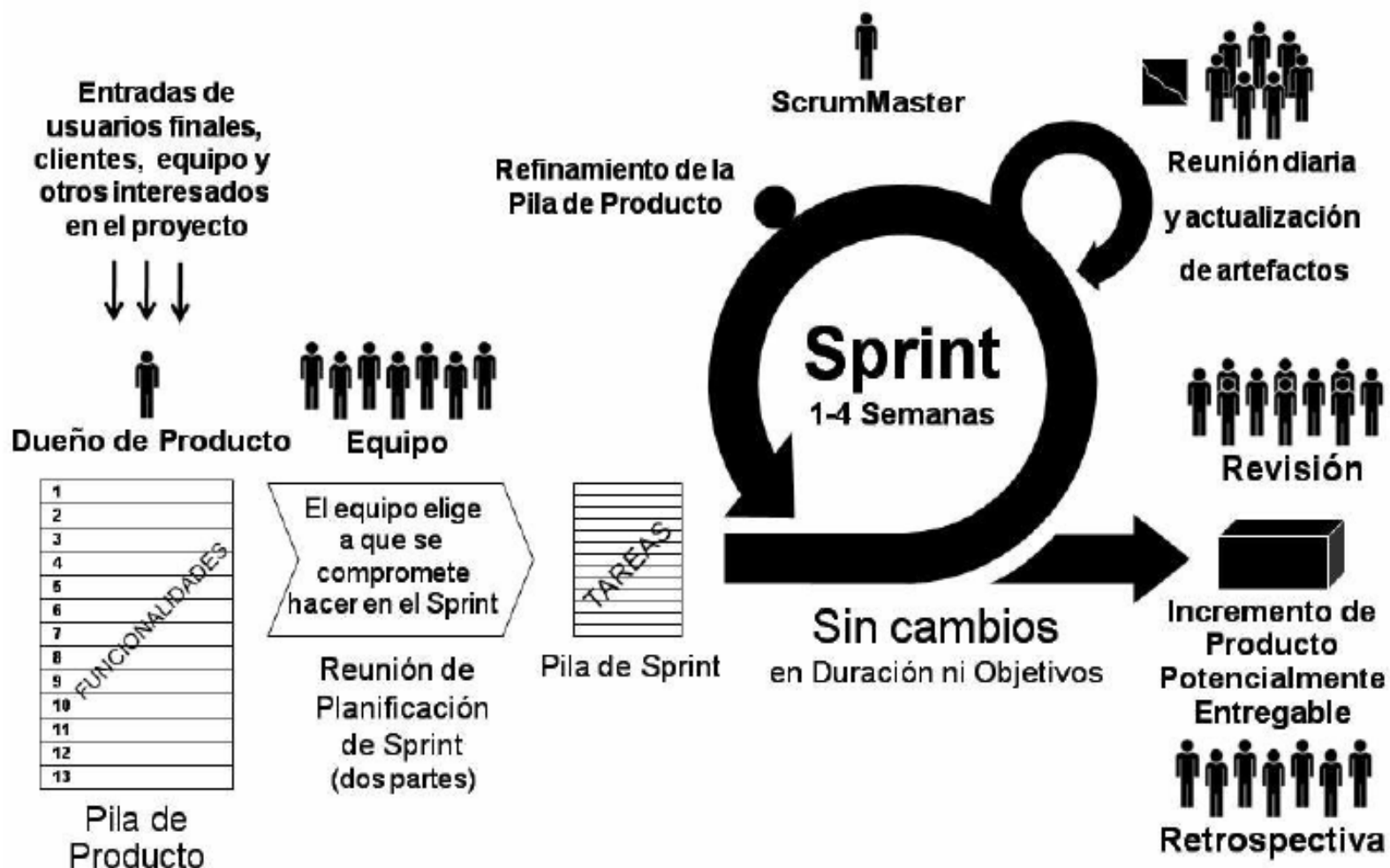


Figura 1. Scrum

# Levantamiento de Requisitos

Los requisitos del producto se capturan teniendo en cuenta la visión del cliente y del usuario

Para ello se utilizan **historias de usuario**, que son unas sencillas tarjetas en las que se recoge de forma esquemática, sencilla y en un lenguaje claro una interacción entre el usuario y el sistema

<i>Generación de Factura</i>
<i>El usuario introduce la información del cliente. Si el cliente ya está registrado con sólo introducir la cédula se deben cargar sus datos. Luego se ingresan los elementos a facturar y las cantidades de cada elemento.</i>
<i>Finalmente el sistema registra la factura y es capaz de imprimirla en la impresora local asociada al terminal del usuario</i>

# Historia de Usuarios

<i>Generación de Factura</i>
<i>El usuario introduce la información del cliente. Si el cliente ya está registrado con sólo introducir la cédula se deben cargar sus datos. Luego se ingresan los elementos a facturar y las cantidades de cada elemento.</i>
<i>Finalmente el sistema registra la factura y es capaz de imprimirla en la impresora local asociada al terminal del usuario</i>

Las historias de usuario sirven de “**recordatorio**” de un grupo de características que es necesario implementar en el sistema.

Antes de implementar una característica se produce una discusión con el usuario y **se refina y extiende la información de la historia de usuario**



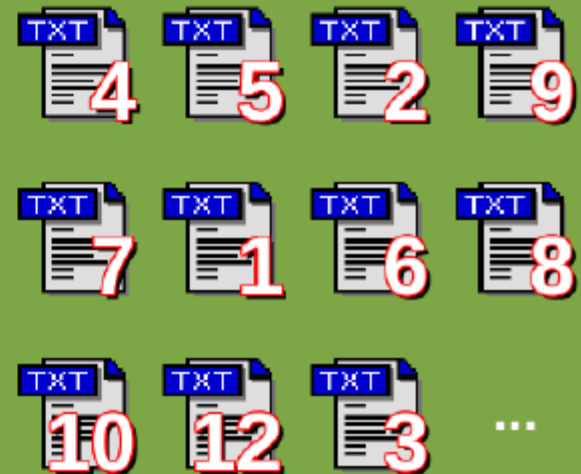
También es posible escribir un breve documento donde se describa la visión general del sistema, esto es útil para tener una visión global de lo que se está haciendo

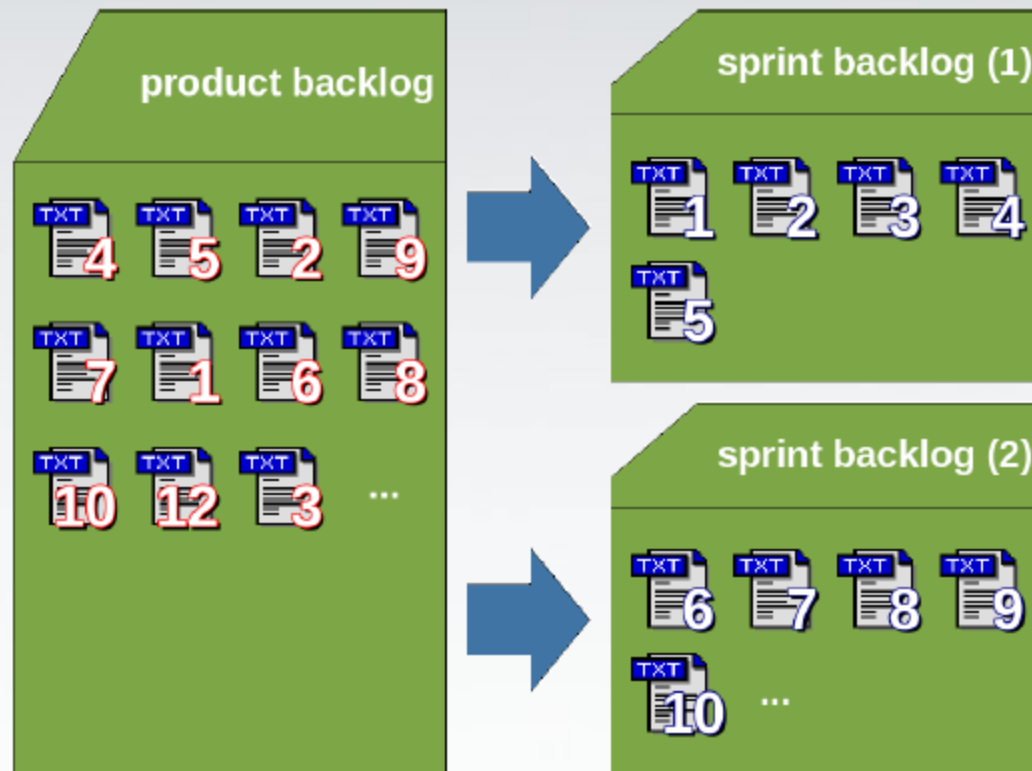
# Requisitos / SCRUM

El **product backlog** tiene todos los requisitos del producto

Se puede ver como una lista de ***todo lo que el usuario/cliente desea*** del producto (a wishlist)

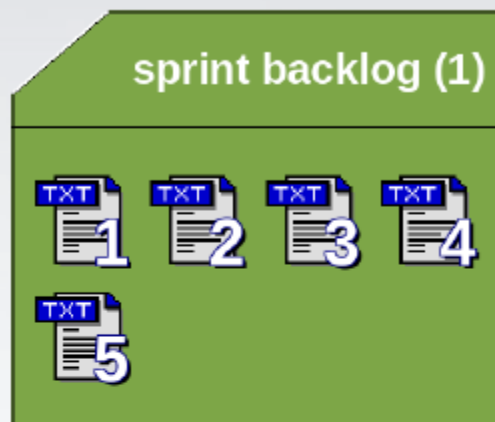
product backlog



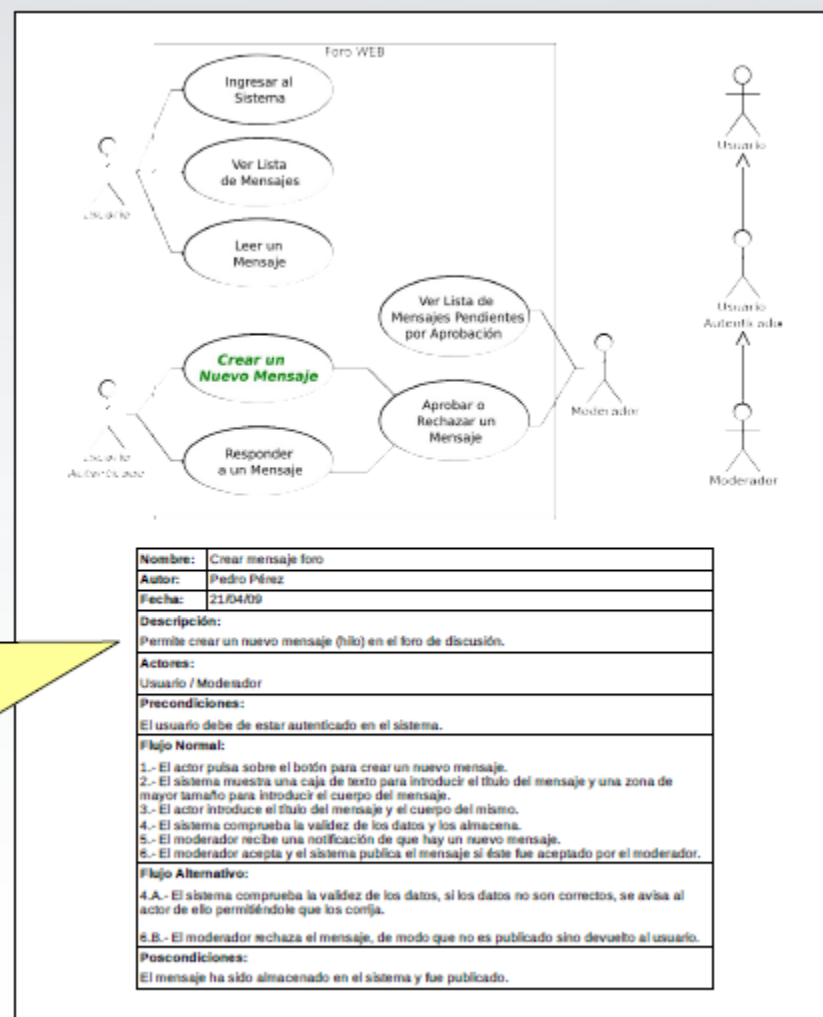


Los requisitos del **product backlog** se priorizan y se asignan a los distintos sprints planificados, es decir, al **sprint backlog** de cada sprint

# Refinamiento de Requisitos



Antes de comenzar el sprint, de ser necesario, se conversa con el cliente y **se refinan las historias de usuario**, usando cualquier técnica deseada para especificar requisitos



Sin embargo siempre es posible  
añadir nuevas historias de usuario al  
product backlog y/o cambiar las  
prioridades de las historias existentes

Una vez comenzado el  
sprint no se añaden  
nuevas historias de  
usuario al sprint backlog  
En general, no hay  
cambios, ni en duración  
ni en objetivos

product  
backlog  
(features)



sprint  
backlog  
(tarefas)



entregable

# Roles en SCRUM

## ***Rol:***

Las acciones o actividades asignadas o requeridas de una persona o grupo  
("La función del maestro",  
"El gobierno debe de...")

## ***Rol:***

Un personaje o ***parte escenificada por un actor***; El comportamiento esperado de un individuo en la sociedad. La función o posición de algo.

# Explicación de roles

Un cerdo y un pollo van caminando por la carretera. El pollo le dice al cerdo:\*

*-Oye, ¿por qué no abrimos un restaurante?*

El cerdo se vuelve y le responde:

*-Buena idea, ¿cómo quieres que lo llamemos?*

El pollo se lo piensa y propone:

*-¿Por qué no lo llamamos “**Huevos con jamón**”.*

*-No cuentes conmigo - responde el cerdo -. En ese caso, tú sólo estarías **IMPLICADO**, mientras que yo estaría realmente **COMPROMETIDO**.*

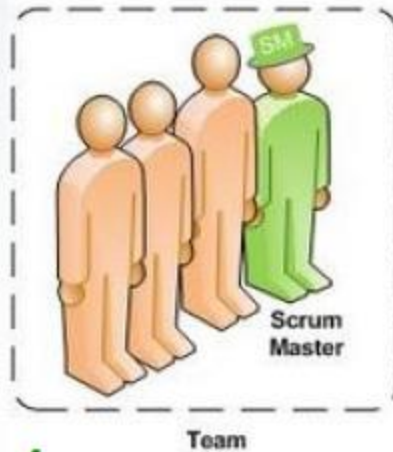
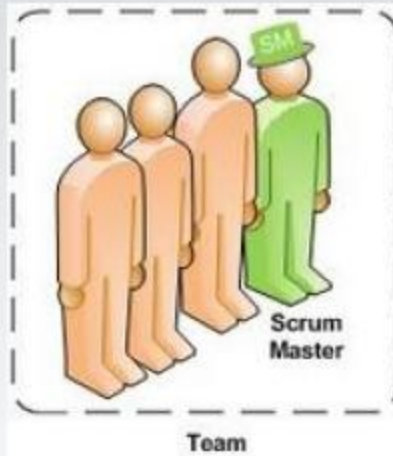


# Explicación de Roles



ese cuento ha sido inmortalizado de muchas formas





**cerdos  
(realmente  
comprometidos)**

**pollos  
(involucrados)**

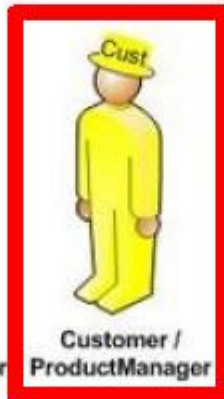
# Product Owner



Team



Team

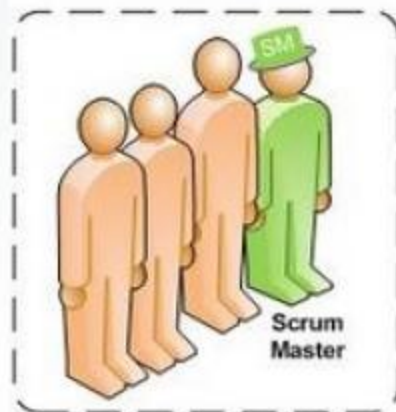


## ***Usuarios / Ejecutivos***

Son los que usarán y eventualmente pagarán el producto



Team



Team



Scrum  
Product Owner

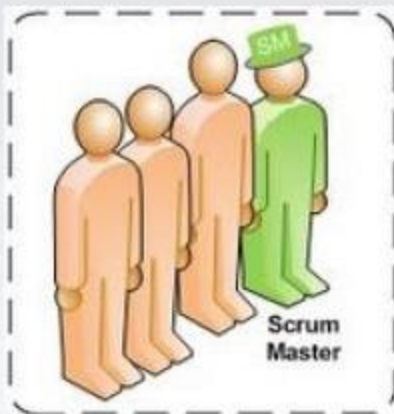


Customer /  
ProductManager

## ***Product Owner:***

Es el “dueño del producto”.

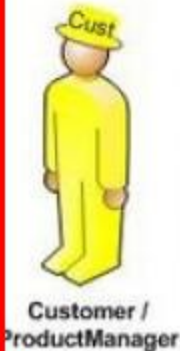
Representa la voz del cliente y aporta la visión de negocio. *Se encarga de escribir las historias de usuario, les da prioridad y las ubica en la lista de requisitos del producto. (Cerdos)*



Team



Team

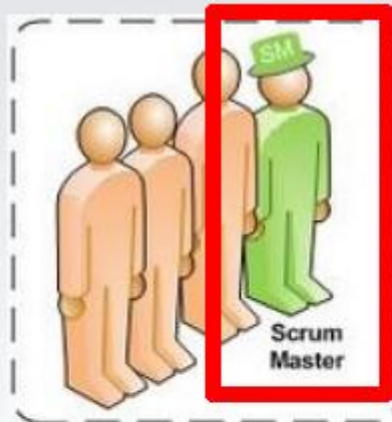


## ***Product Owner:***

Se encarga de decidir ***que va y que no va*** en el ***product backlog***, así como definir las prioridades de las distintas historias de usuario



# Scrum Master



Team

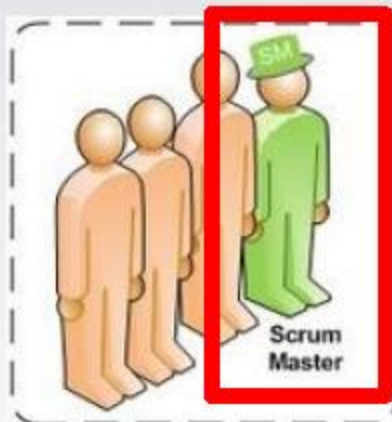


Team



**Scrum Master**

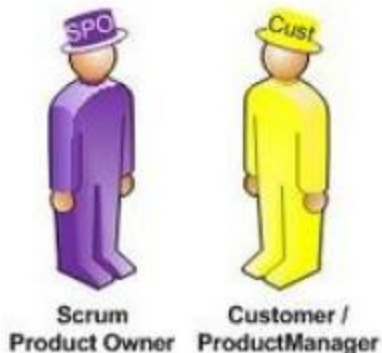
*Tiene como principal papel el de dejar el camino libre de obstáculos e impedimentos para que el resto del equipo consiga el objetivo del sprint.*



Team



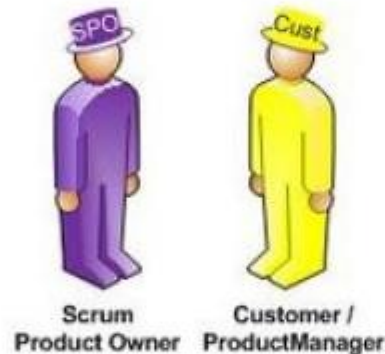
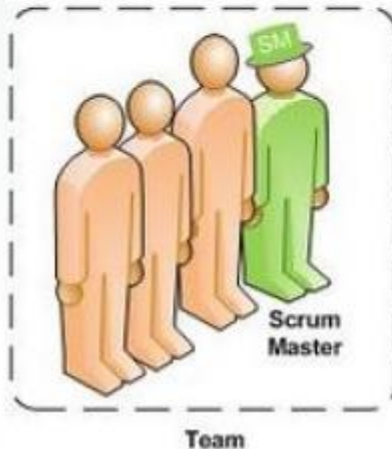
Team



## **Scrum Master**

*Organiza reuniones,  
hace seguimiento del  
trabajo que se está  
llevando a cabo y apoya  
en la planificación de  
los sprints / entregas*

# Equipo SCRUM



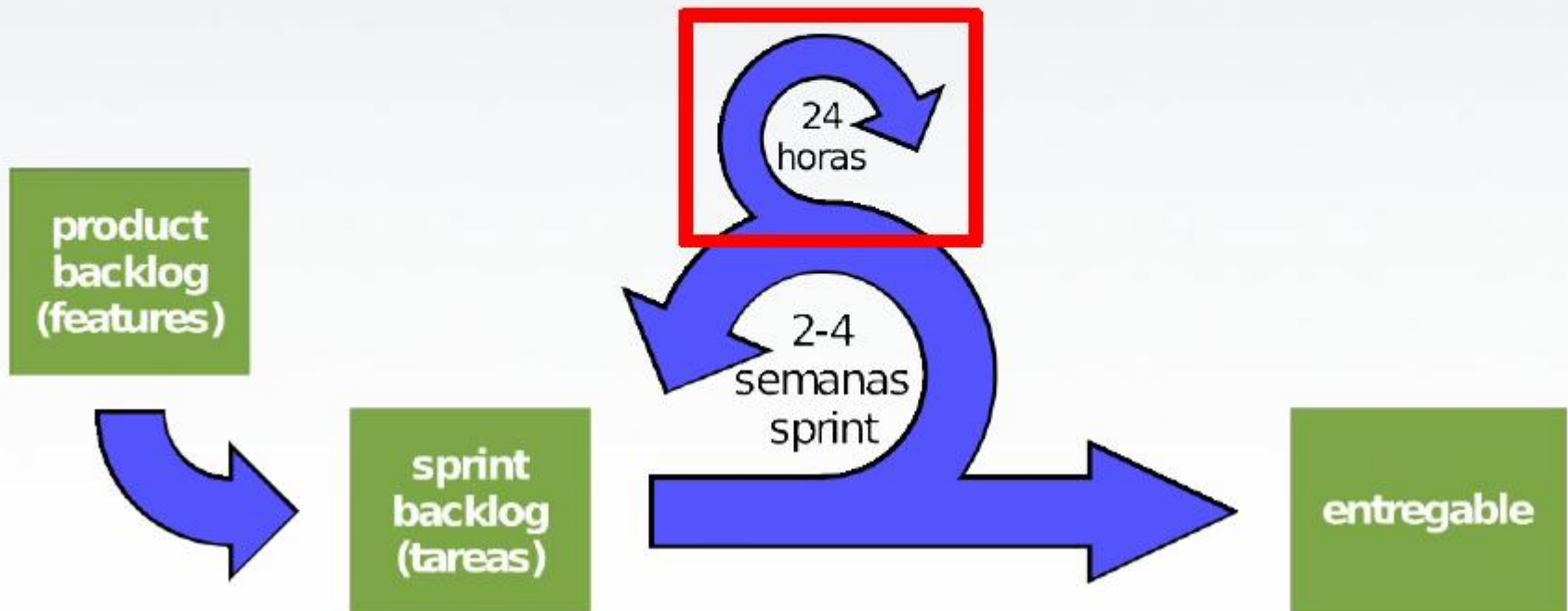
## ***Equipo***

*Está compuesto por los roles tradicionales: desarrolladores, probadores, etc.*

# Gestión, Control y Seguimiento

## *Reunión Diaria:*

Es una figura fundamental en SCRUM.  
Tiene que reunirse **TODO** el equipo y debe hacerse  
según ciertas reglas





## ***Reunión Diaria:*** **REGLAS:**

La reunión es diaria y se hace siempre a una hora predefinida, normalmente por la mañana

Es importante que todos los miembros del equipo acudan puntuales

***Reunión Diaria:***  
**REGLAS:**

La reunión debe durar **alrededor de 15 minutos**

Se realiza de pie, para mantener el máximo de concentración y atención

Todos los roles son bienvenidos, pero **sólo los cerdos** pueden hablar

## ***Reunión Diaria:***

### **REGLAS:**

En la reunión se realizan las siguientes  
3 preguntas clave:

***¿Qué has hecho desde ayer?***

***¿Qué tienes planeado hacer mañana?***

***¿Has encontrado algún problema para  
conseguir tu objetivo?***

**¿les suena conocido?**

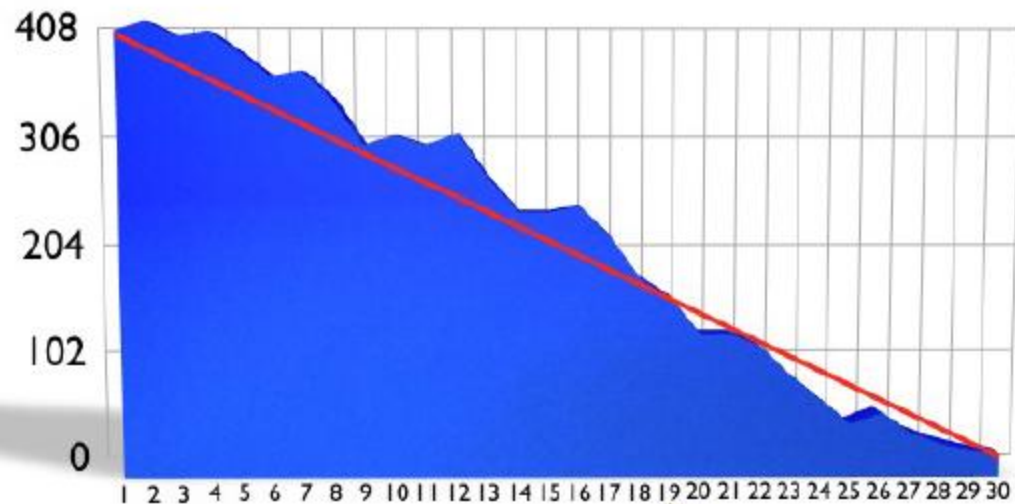
## ***Reunión Diaria:*** **REGLAS:**

La transparencia es un aspecto muy importante en estas reuniones

Todos los miembros saben que están haciendo los demás, y los problemas deben ser sacados a la luz en cuanto se detectan

**¿les suena conocido?**

# Scrum Burn Down



## EJE Y

Trabajo restante,  
horas, puntos de  
función u otra  
unidad de medida

## EJE X

Día o fecha del sprint

**Esto es responsabilidad  
del Scrum Master**

**Las siguientes hojas de excel fueron tomadas de:**

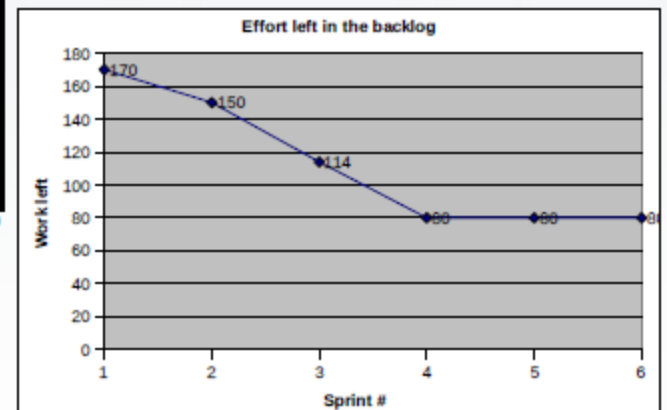
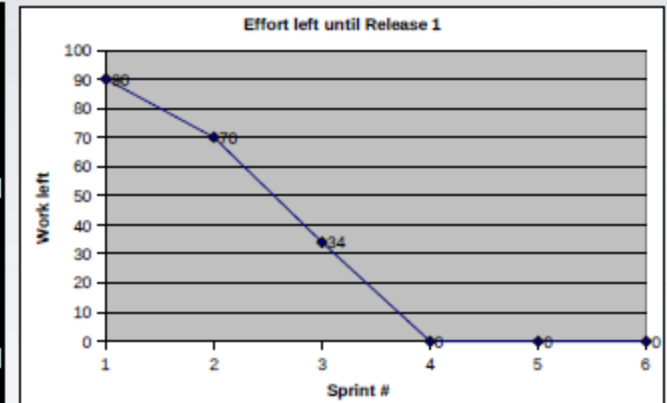
<http://agilesoftwaredevelopment.com/>

**Un video explicativo en:**

<http://www.youtube.com/watch?v=WqgZCiqiYAY>

## Weather on Mobile

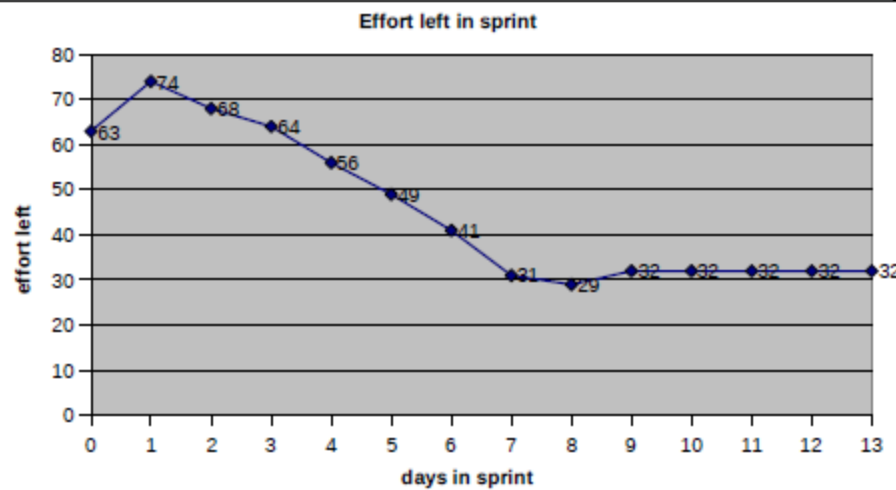
ID	Description	Sprint #	1	2	3	4	5	6
<b>Effort needed for Release 1 as in the beginning of the sprint</b>			90	70	34	0	0	0
	1 Set up continuous integration system		5	0	0	0	0	0
	2 Create compilable application skeleton		5	0	0	0	0	0
	3 Display current temperature in a simplest possible way		13	0	0	0	0	0
	4 Set up the web server for serving weather data		3	0	0	0	0	0
	5 Implement stubby WeatherML support on the server side		13	0	0	0	0	0
<b>Sprint 1 Make sample data go from server to device</b>								
	6 Graphics support on the client side		20	0	0	0	0	0
	16 Make the graphics library draw some icon and sample temperature text			13	0	0	0	0
	17 Draw the real weather screen			8	0	0	0	0
	7 Implement support for several days			8	0	0	0	0
	8 Implement support for rain, snow, etc. icons			2	0	0	0	0
	9 City changing support			5	0	0	0	0
<b>Sprint 2 Minimal working version</b>								
	10 Fetch one day temperature data from the weather provider system			13	13	0	0	0
	11 Fetch rain, snow, etc details from the provider			8	8	0	0	0
	12 Fetch several days data from the provider			5	5	0	0	0
	13 Auto-refresh feature			8	8	0	0	0
<b>Sprint 3 Plug in the real weather data</b>								
<b>Release 1 Sellable version</b>								
	14 Inject simulated ads from the test server			20	20	20	20	20
	15 Plug real ads in			20	20	20	20	20
	18 Change current city automatically according to the cell info			40	40	40	40	40
<b>Sprint 4 Advertisements support</b>								
<b>Release 2 Ad-supported version</b>								
<b>Effort in the whole backlog</b>			170	150	114	80	80	80



Backlog state taken after the end of sprint 3 = after release 1

<http://agilesoftwaredevelopment.com/scrum/simple-product-backlog>

Story ID	Story/task	days in sprint / effort left													
		0	1	2	3	4	5	6	7	8	9	10	11	12	13
		63	74	68	64	56	49	41	31	29	32	32	32	32	32
10	<b>Fetch one day temperature data from the weather provider system</b>														
	Make our server connect and authenticate to the provider system	4	16	12	8	3	3	3	3	3	3	3	3	3	3
	Read provider's data directory	8	7	7	7	4	0	0	0	0	0	0	0	0	0
	Parse the current temperature out of the data	6	6	4	4	4	1	1	1	1	1	1	1	1	1
	Push the temperature data to the client	16	16	16	16	16	16	8	2	0	0	0	0	0	0
11	<b>Fetch rain, snow, etc details from the provider</b>														
	Parse snow/rain data from the provider's data	4	4	4	4	4	4	4	0	0	0	0	0	0	0
	Push the snow/rain data to the client	4	4	4	4	4	4	4	4	4	0	0	0	0	0
	Redesign client screen a bit										3	3	3	3	3
	Refactor the server code										4	4	4	4	4
12	<b>Fetch several days data from the provider</b>														
	Parse the weather data in day packs	10	10	10	10	10	10	10	10	10	10	10	10	10	10
	Push several days data to the client	3	3	3	3	3	3	3	3	3	3	3	3	3	3
13	<b>Auto-refresh feature</b>														
	Make the client ping server once per 4 hours	6	6	6	6	6	6	6	6	6	6	6	6	6	6
	Make the server update the client	2	2	2	2	2	2	2	2	2	2	2	2	2	2



Backlog state taken after day 9

based on: <http://agilesoftwaredevelopment.com/scrum/simple-sprint-backlog>



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

**Se puede poner en una pared en la oficina y sirve para hacer el seguimiento de las tareas del sprint en curso**

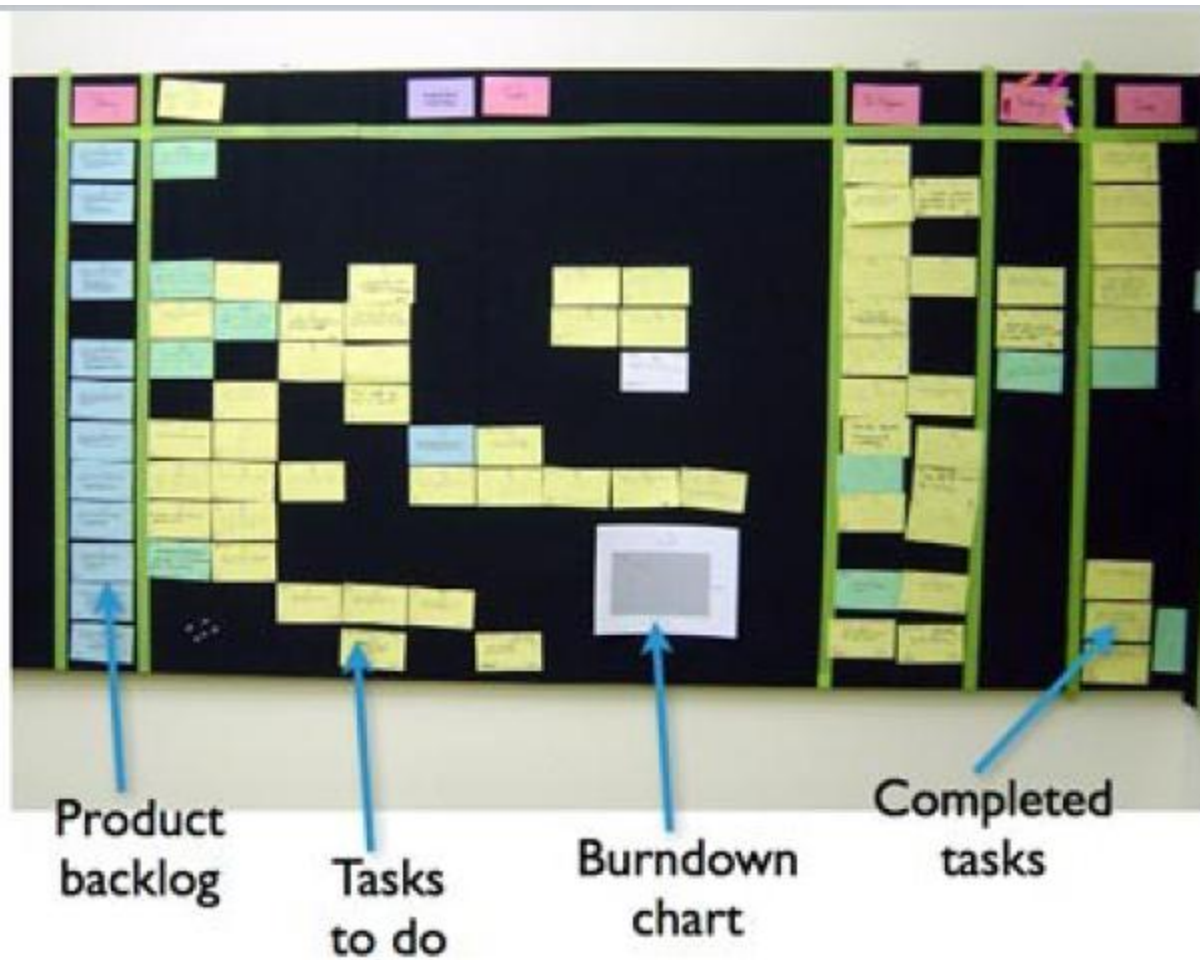
**(también se puede usar alguna herramienta de issue/task tracker)**

**(y si no me creen...)**

<http://www.mountangoatsoftware.com/scrum/task-boards>



<http://www.mountangoatsoftware.com/scrum/task-boards>



<http://www.mountangoatsoftware.com/scrum/task-boards>

Video recomendado  
Scrum en 10 minutos