

IIC 2143 – Ingeniería de Software

# SCRUM

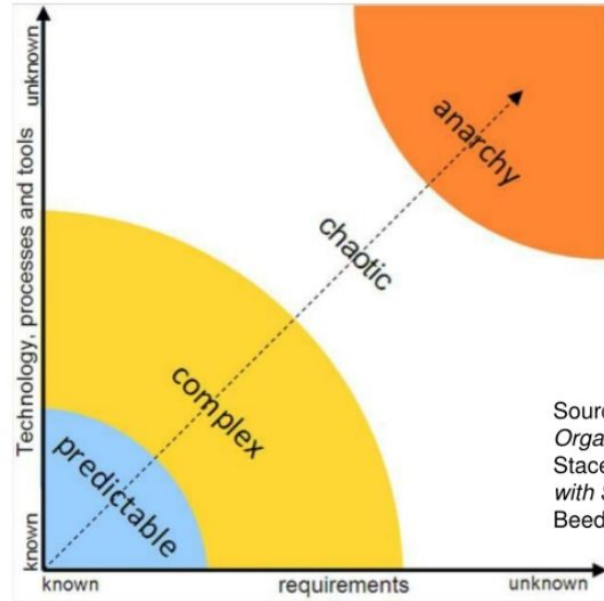
M. Trinidad Vargas  
mtvargas1@uc.cl

# ¿Quienes usan SCRUM?



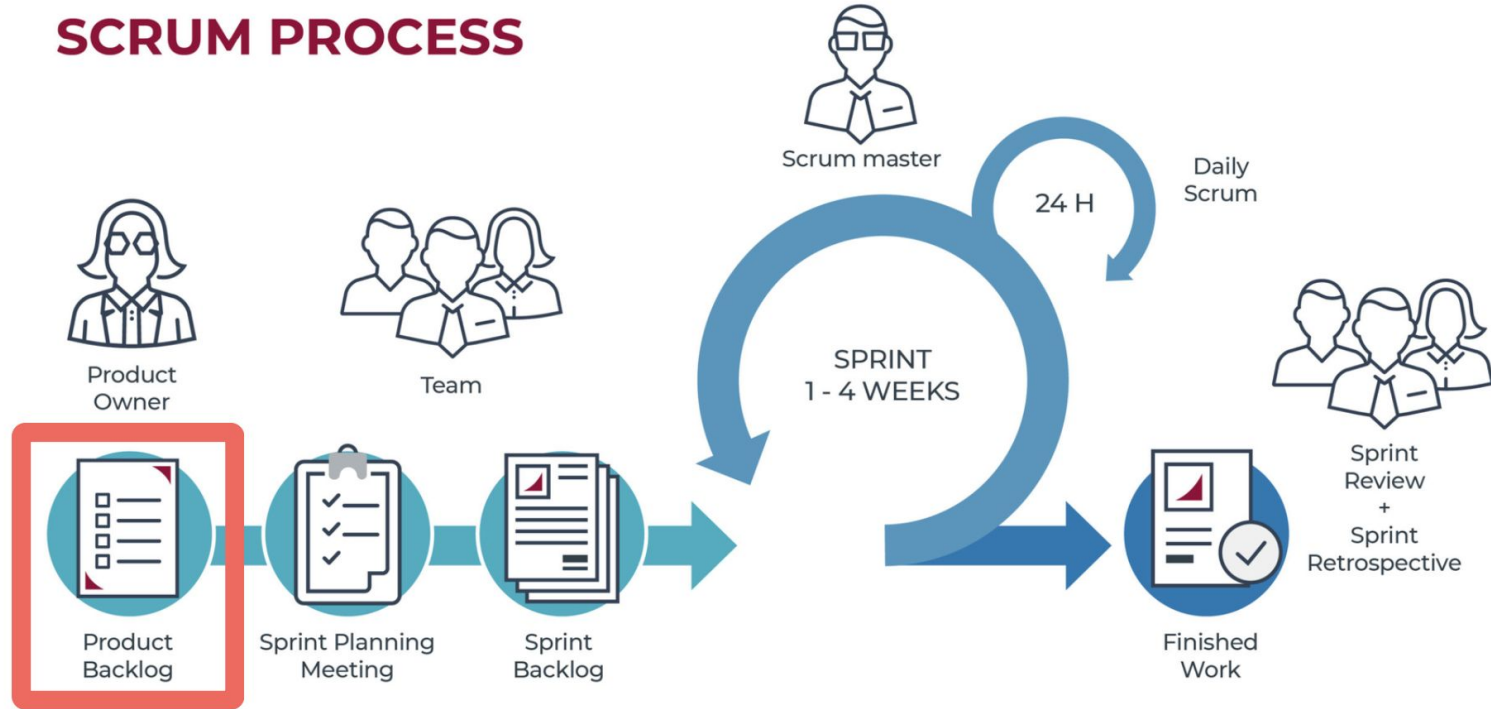
# ¿Cuándo es más útil?

En ambientes complejos  
o caóticos

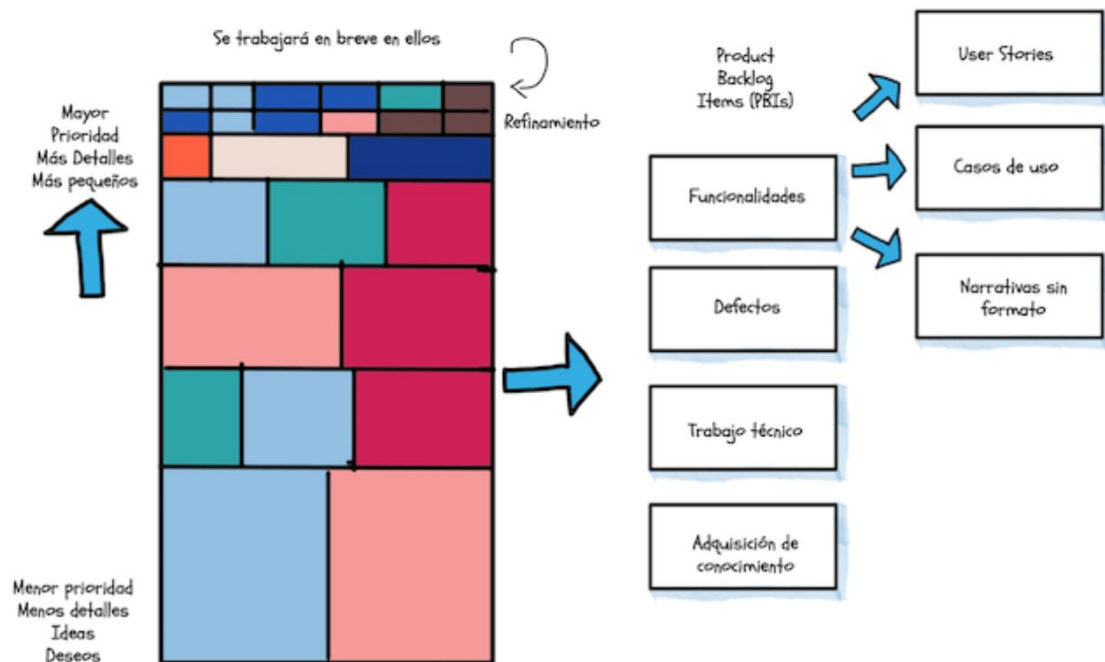


Source: *Strategic Management and Organizational Dynamics* by Ralph Stacey in *Agile Software Development with Scrum* by Ken Schwaber and Mike Beedle.

# SCRUM PROCESS



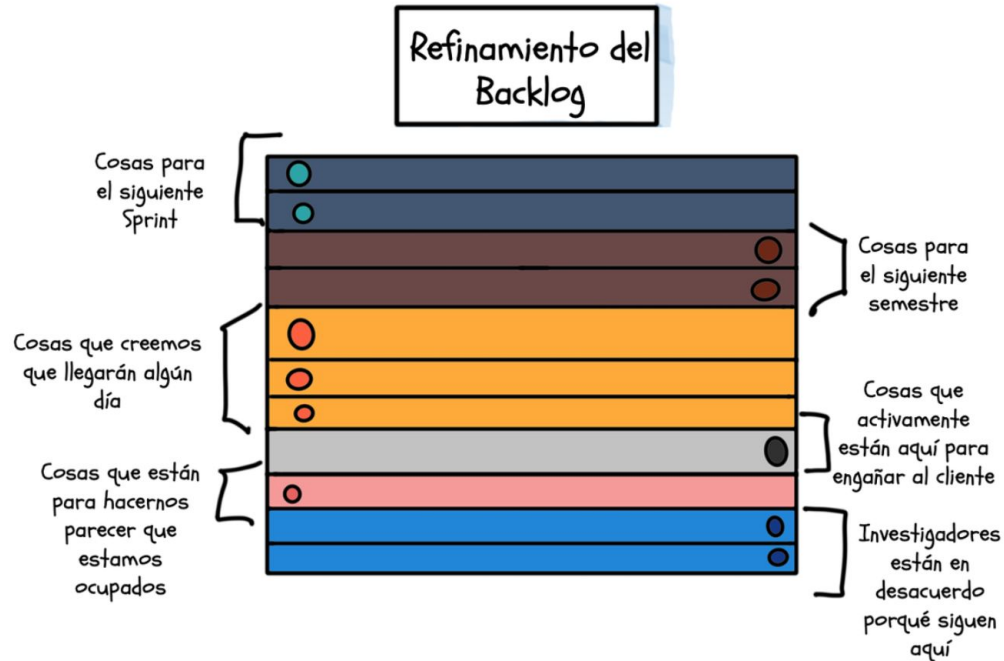
# Product Backlog



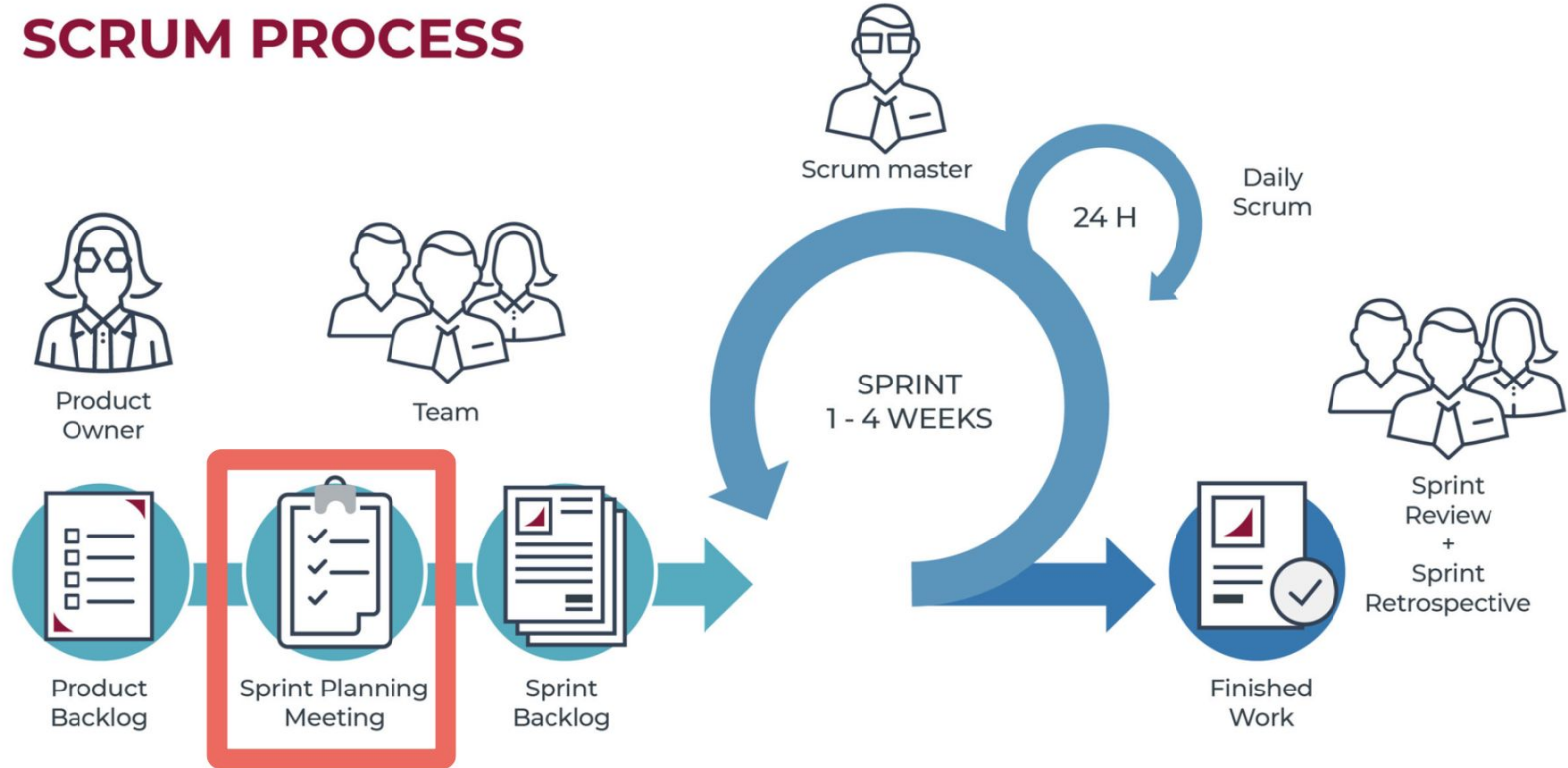
# Product Backlog

Backlog item	Estimate
Allow a guest to make a reservation	3
As a guest, I want to cancel a reservation.	5
As a guest, I want to change the dates of a reservation.	3
As a hotel employee, I can run RevPAR reports (revenue-per-available-room)	8
Improve exception handling	8
...	30
...	50

# Product Backlog



# SCRUM PROCESS





# Sprint Planning

¿QUÉ HACEMOS?

Planificar el trabajo necesario  
para entregar un **INCREMENTO**

que cumpla el  
Definition of Done



El Product Owner dice el QUÉ

Los Desarrolladores deciden el CÓMO



¿QUIÉNES PARTICIPAN?

El **Scrum Team**

Product Owner



Scrum Master



Desarrolladores



# Sprint Planning

1

Definir el  
OBJETIVO DEL SPRINT



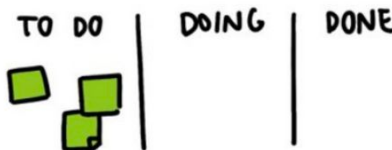
¿Por qué?

Para ayudar al equipo  
a poner el **FOCO**  
en lo que es  
más importante!



2

Crear el  
SPRINT BACKLOG



- Priorizando los elementos de trabajo
- Estimando el esfuerzo del trabajo a realizar
- Acordando entre todos el alcance del Sprint

3

Elaborar un  
PLAN



- Dividiendo los elementos de trabajo en trozos más pequeños
- Identificando posibles dependencias, bloqueos o impedimentos
- Debatiendo cuál será la solución técnica

# Sprint Planning

1

Definir el  
OBJETIVO DEL SPRINT



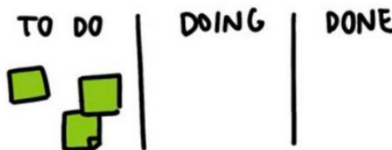
¿Por qué?

Para ayudar al equipo  
a poner el **FOCO**  
en lo que es  
más importante!



2

Crear el  
SPRINT BACKLOG



- Priorizando los elementos de trabajo
- Estimando el esfuerzo del trabajo a realizar
- Acordando entre todos el alcance del Sprint

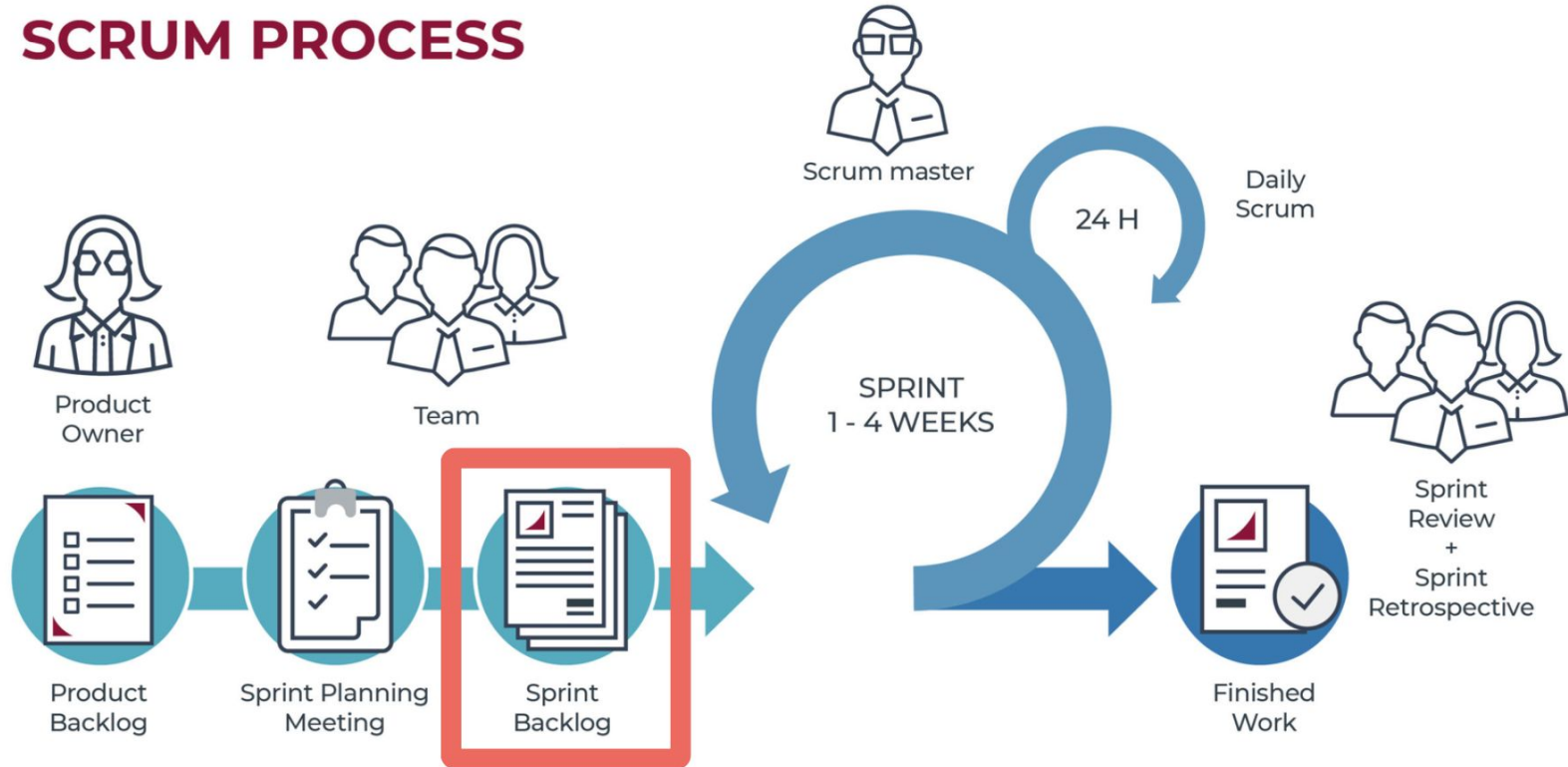
3

Elaborar un  
PLAN

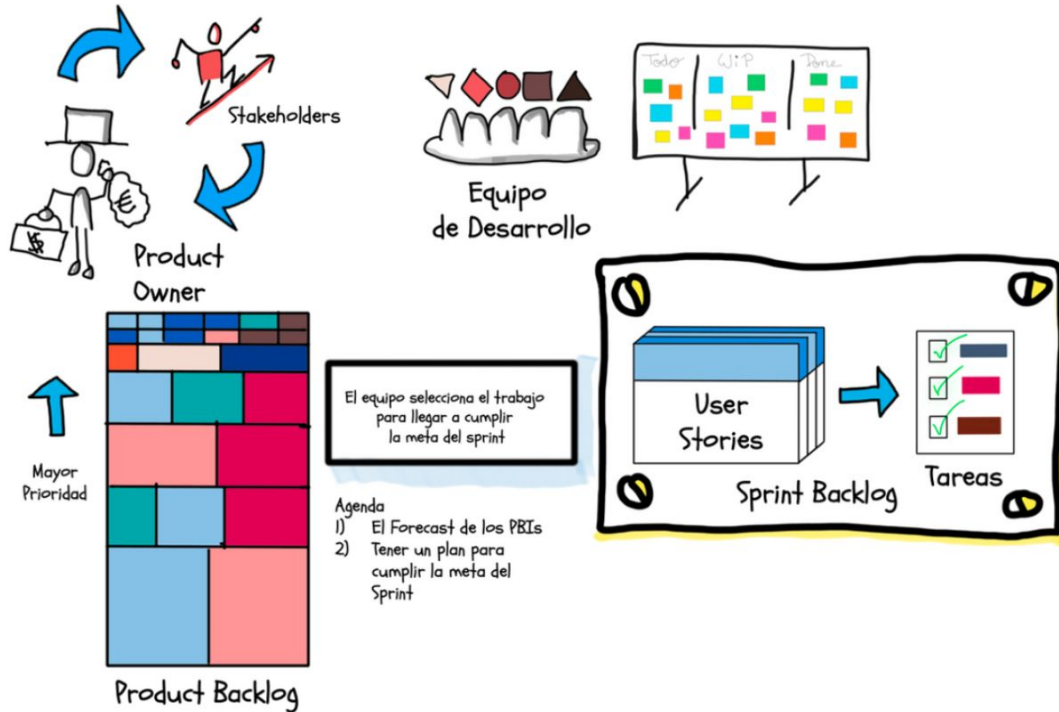


- Dividiendo los elementos de trabajo en trozos más pequeños
- Identificando posibles dependencias, bloqueos o impedimentos
- Debatiendo cuál será la solución técnica

# SCRUM PROCESS



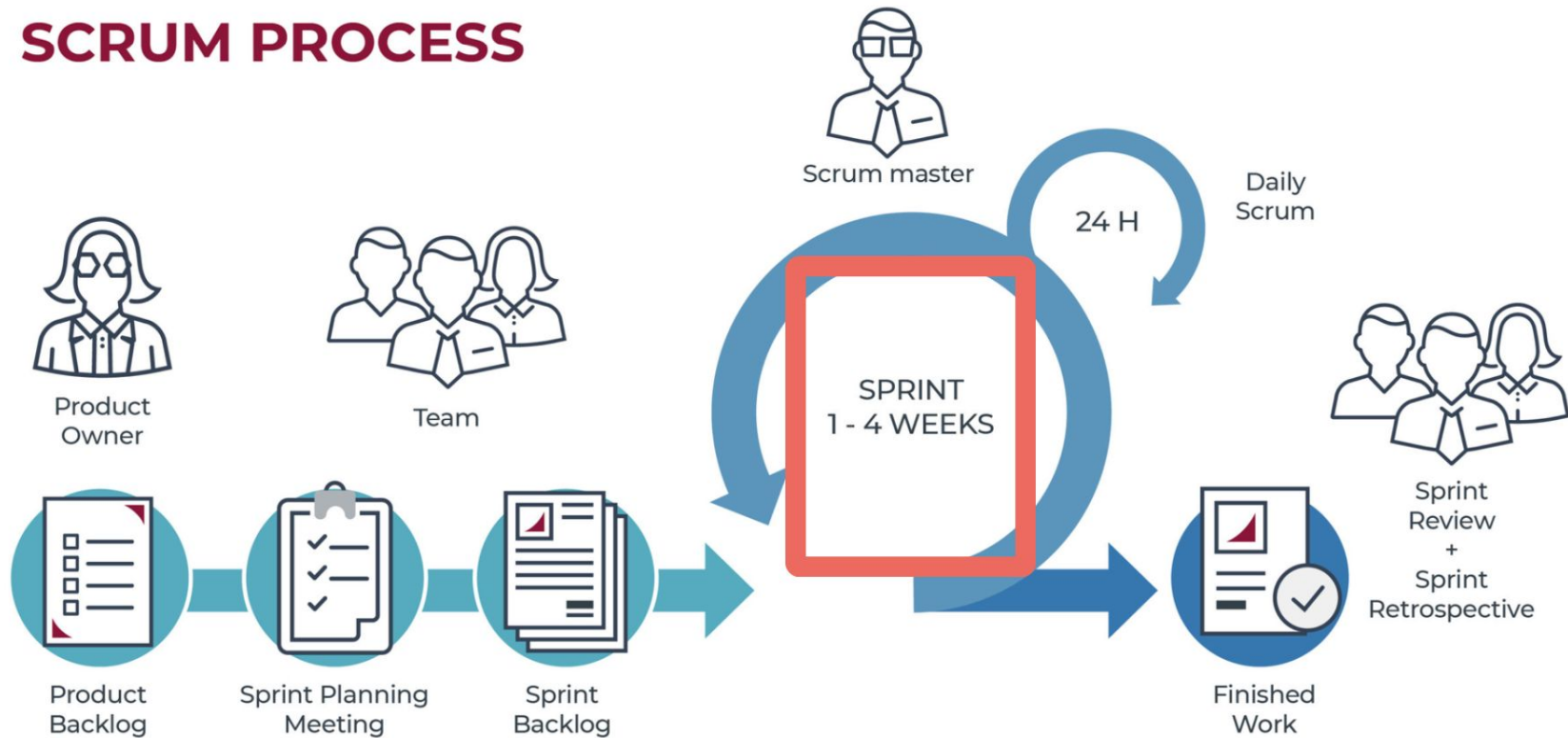
# Sprint Backlog



# Sprint Backlog

Tasks	Mon	Tues	Wed	Thur	Fri
Code the user interface	8	4	8		
Code the middle tier	16	12	10	4	
Test the middle tier	8	16	16	11	8
Write online help	12				
Write the foo class	8	8	8	8	8
Add error logging			8	4	

# SCRUM PROCESS



# Sprint

Es un evento de longitud fija: Generalmente entre 2 y 4 semanas

## Input:

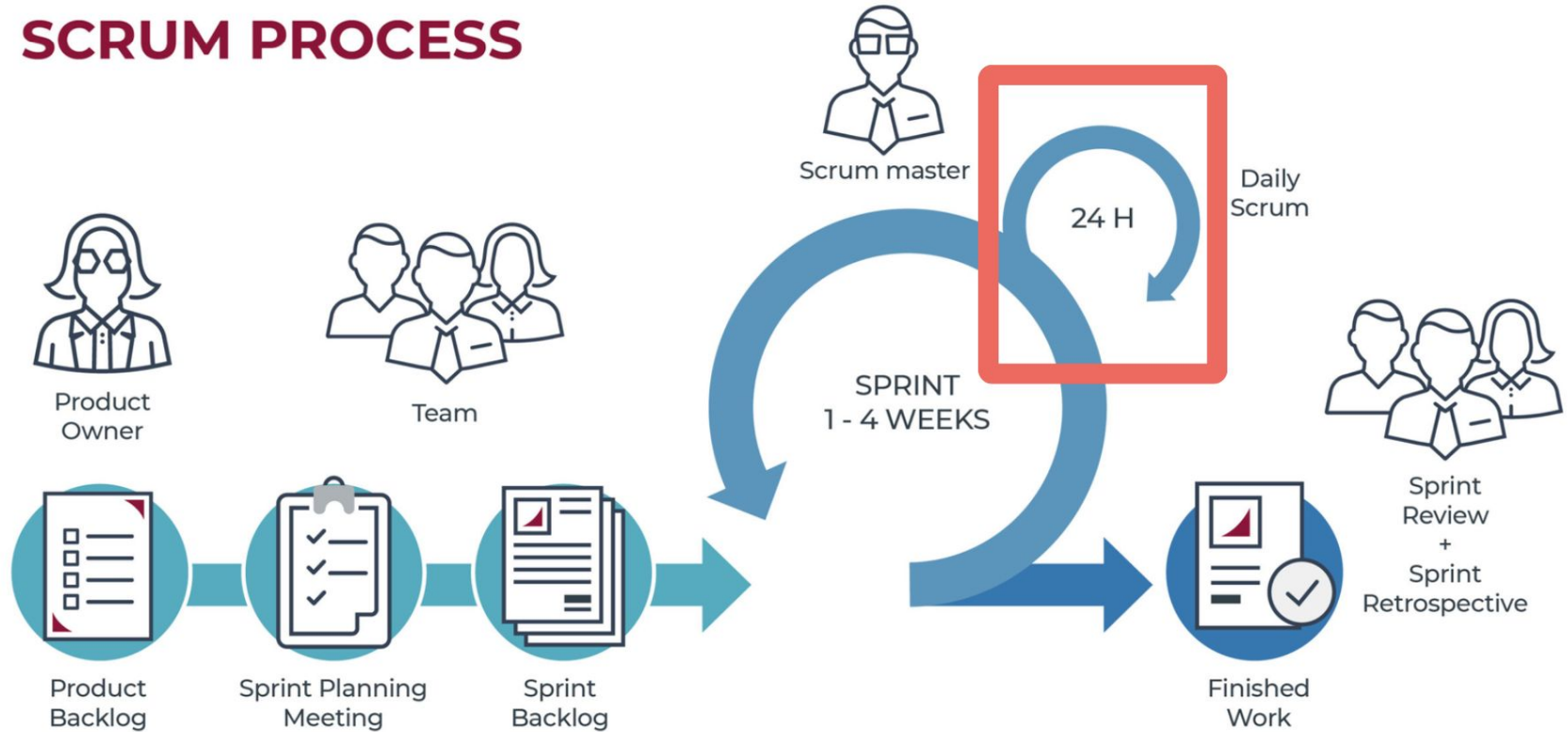
- El objetivo del Sprint
- Ítems seleccionados del product backlog
- Conjunto de tareas del sprint backlog

## Output:

- **Incremento** al producto o valor agregado existente
- Paso listo (cumple DoD) que contribuye al Product Goal



# SCRUM PROCESS



# Daily Meeting



¿Qué considerar?

- Riesgos
- Impedimentos
- Necesidad de refinar



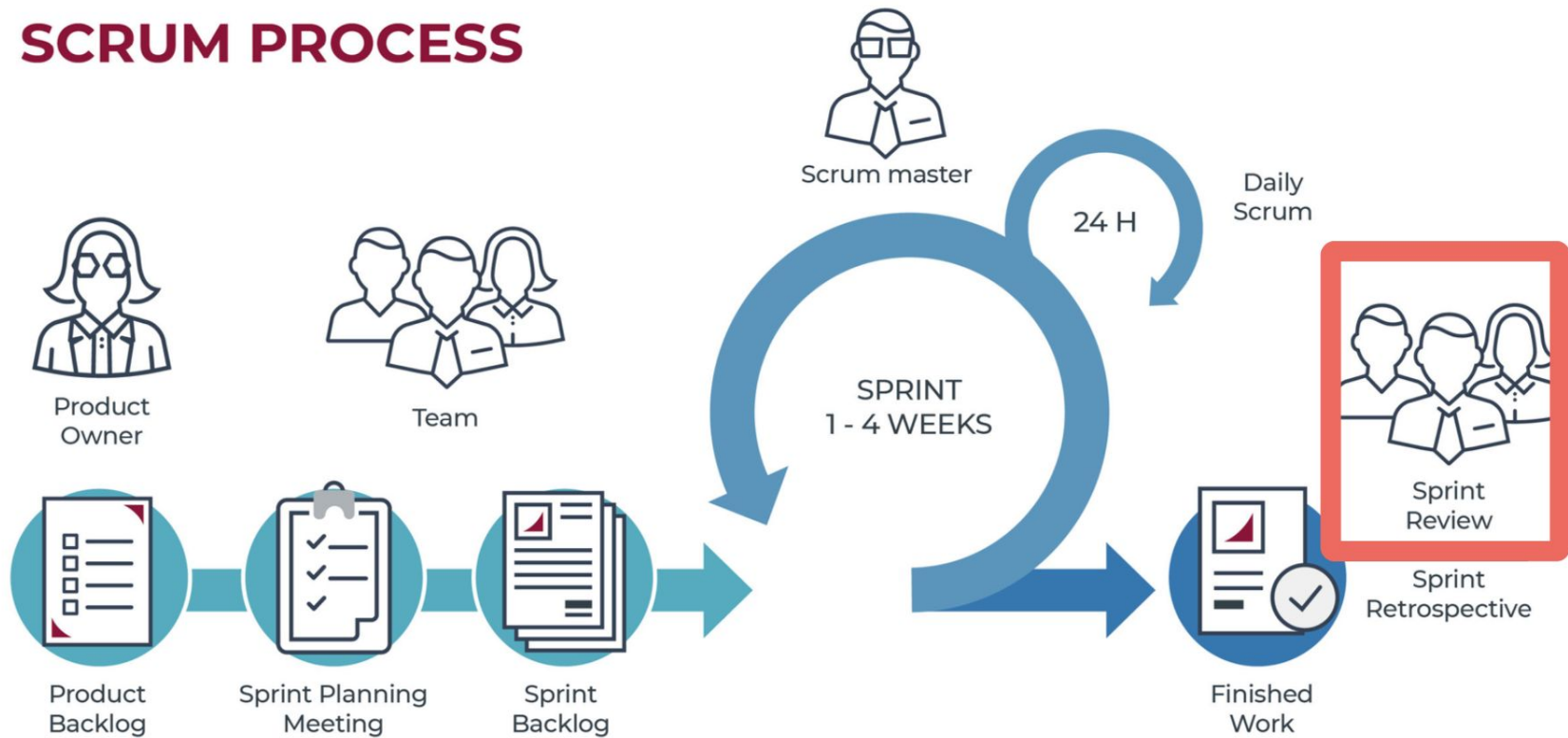
Todos los días, en el mismo sitio y a la misma hora



Después, el Product Owner y el Scrum Master pueden ayudar a aclarar dudas y eliminar impedimentos

¿Qué hice ayer?, ¿qué haré hoy?, ¿tuve algún impedimento o complicación (stopper)?

# SCRUM PROCESS

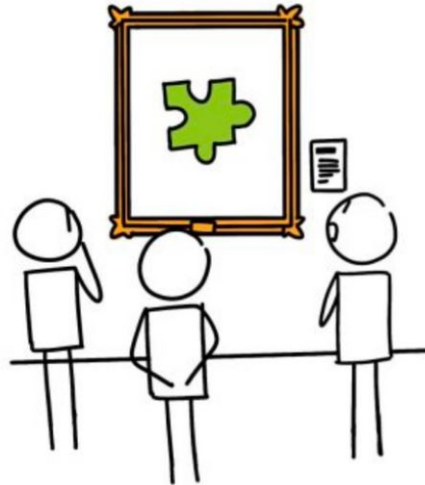


# Sprint Review

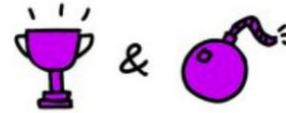
Invitamos a los Stakeholders  
para recibir su FEEDBACK



MOSTRAMOS y comentamos  
el trabajo terminado



Compartimos los LOGROS e  
IMPEDIMENTOS del Sprint



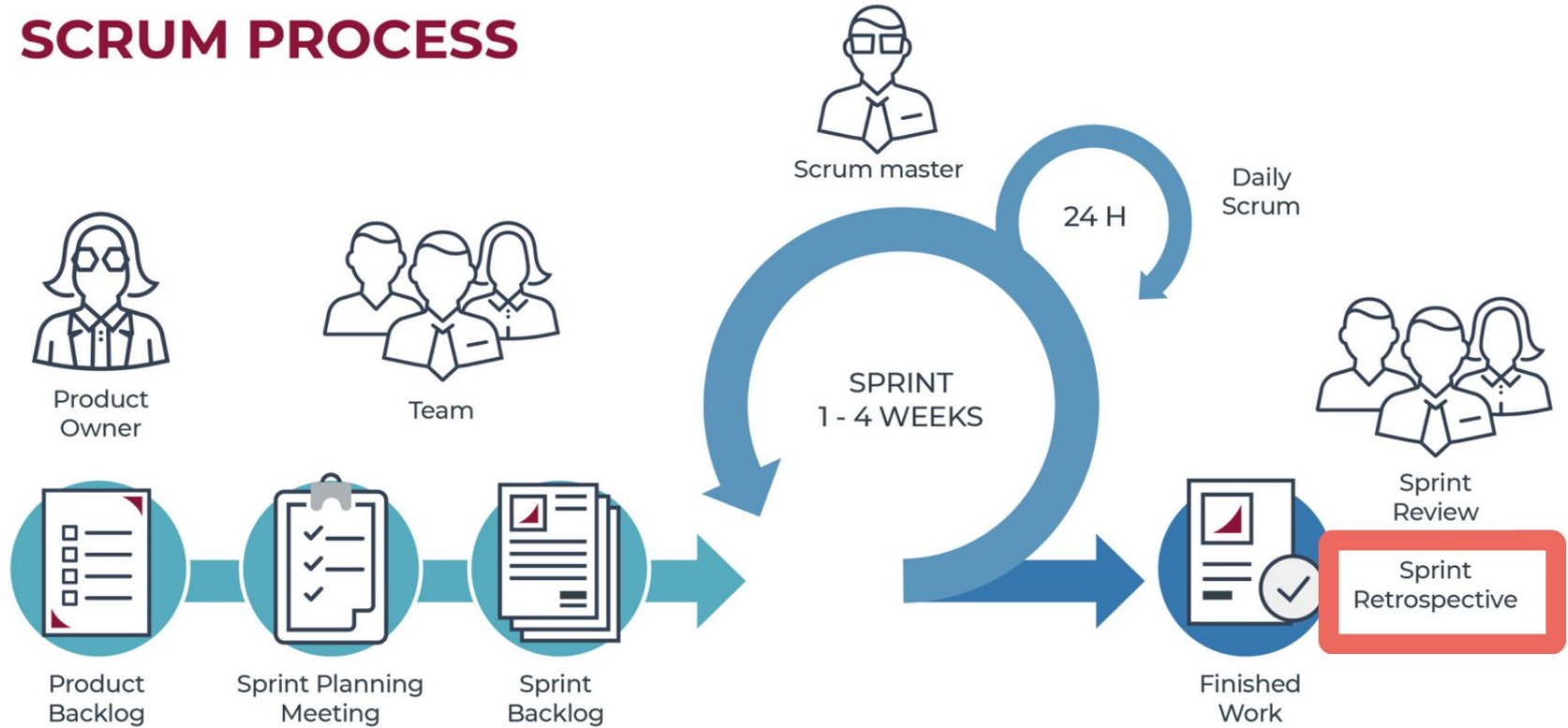
Comentamos los PRÓXIMOS PASOS

BACKLOG:

- 1 ↑
- 2 ↑
- 3 ↓

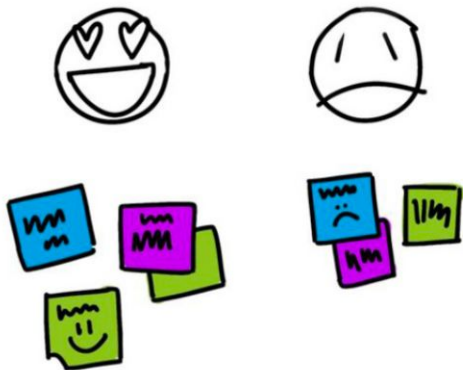
SOMOS TRANSPARENTES ENTRE EL EQUIPO Y CON LOS STAKEHOLDERS :)

# SCRUM PROCESS



# Sprint Retrospective

## 1 Inspeccionamos el Sprint



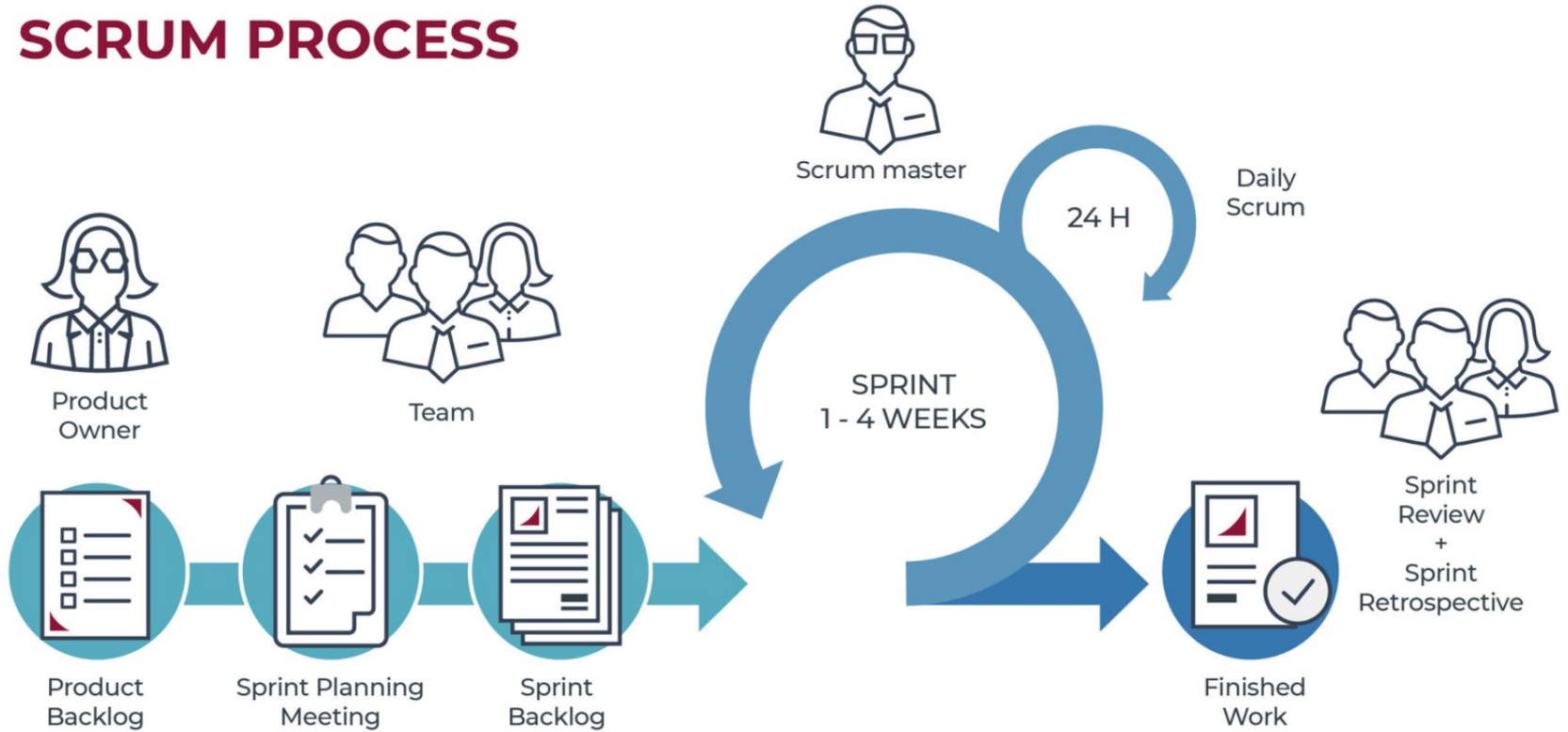
- Cómo nos hemos comunicado y relacionado
- Si nuestros procesos han sido eficientes
- Si nuestras herramientas son útiles

## 2 Elaboramos un plan de acción que nos ayude a ser mejores



Lo llevaremos a cabo en el siguiente Sprint ;)

# SCRUM PROCESS



# Siempre en evolución

Algunos cambios de la última guía oficial de SCRUM (2020)

- Introduce el objetivo del producto
- Aumenta la independencia del equipo: el equipo es independiente en decidir quién, cómo y ahora en qué realizar el trabajo
- Responsabilidad en vez de roles

<https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Spanish-Latin-South-American.pdf>



# Responsabilidades (Roles)



## PRODUCT OWNER

- gestiona el Product Backlog
- está disponible para el equipo
- colabora con los Stakeholders



## DESARROLLADORES

- construyen la solución
- velan por la calidad
- se autoorganizan



## SCRUM MASTER

- ayuda a eliminar impedimentos
- forma y acompaña en prácticas ágiles
- es facilitador y líder servicial

# Responsabilidades



## PRODUCT OWNER

- gestiona el Product Backlog
- está disponible para el equipo
- colabora con los Stakeholders

- Define las funcionalidades del producto.
- Define el alcance y cronograma.
- Responsable de alcanzar los objetivos financieros. Prioriza el product backlog.
- Ajusta las características y prioriza para cada sprint. Acepta o rechaza los resultados del trabajo.

# Responsabilidades



## DESARROLLADORES

- construyen la solución
- velan por la calidad
- se autoorganizan

- 4 a 8 personas.
- Programadores, testers, UI/UX Designers, entre otros. El equipo se auto-gestiona, auto-organiza.
- Idealmente no hay títulos.

# Responsabilidades



## SCRUM MASTER

- ayuda a eliminar impedimentos
- forma y acompaña en prácticas ágiles
- es facilitador y líder servicial

- Responsable por velar por los valores y prácticas SCRUM. Elimina posibles impedimentos.
- Entrena al equipo para conseguir la mejor performance. Ayuda a mejorar la productividad.
- Habilita la relación cercana entre todos los roles y funciones.
- Escudo del equipo ante las interferencias externas.

# Scrum Planning

- Story Points
- Planificar un release
- Planificar una iteración
- Necesidad de ajustar

# Story Points

Son una unidad relativa de medida que indican el esfuerzo necesario para realizar una determinada historia de usuario.

- Los story points son estimados por todos los miembros del equipo.
- Representan un valor relativo, que puede variar de equipo a equipo.



1 point



3 points



5 points



8 points



11 points

# Story Points

## Valor relativo

Los puntos de historia dan una visión del tamaño de la historia de usuario.

Una historia de 3 puntos es más grande en **complejidad** que una de 1 punto.

Una historia de 2 puntos no es necesariamente el doble que una historia de 1 punto.

Una historia de 11 puntos puede ser dividida en dos o más historias

# Story Points

## Determinar los puntos

**Dificultad:** ¿Cuánto esfuerzo requiere completar la historia del usuario, según la definición de “listo”?

**Complejidad:** ¿Qué tan complejo es crear la historia de usuario? ¿Es sencillo o un desafío?

**Incertidumbre:** ¿Cuáles son las posibilidades de encontrar sorpresas que no habíamos previsto de antemano?



# Story Points

## Determinar los puntos

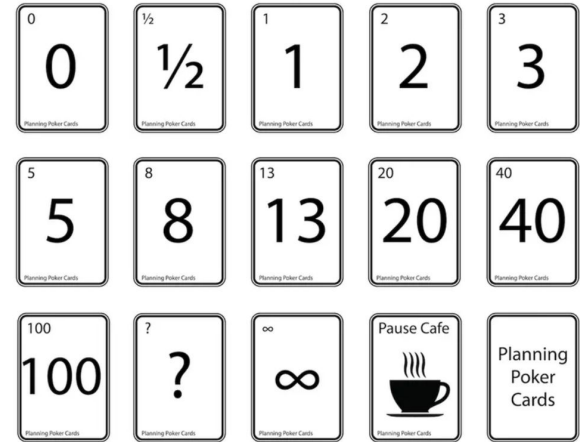
1. Seleccionamos una historia como punto de referencia y le asignamos puntos de historia.
2. Las próximas historias usarán de referencia a esta para ver si es más difícil o es más fácil.

1	2	3	5	8	13
A user can...	A user can...	A user can...	A user can...	A user can...	A user can...
A user can...	A user can...	A user can...	A user can...		A user can...
A user can...		A user can...	A user can...		
A user can...			A user can...		

# Story Points

## Determinar los puntos: Scrum Poker

1. Se selecciona una historia y los participantes seleccionan una carta para representar el valor en puntos de historia que creen que tiene la historia.
2. Todos muestran las cartas a la vez para evitar sesgo.
3. Si son muy distintas, se discuten las diferencias y se vuelve al paso 1.
4. Se acaba cuando hay un consenso entre los puntos a asignar.



# Velocidad de desarrollo

## ¿Cuántos puntos puede manejar un sprint?

- La velocidad de desarrollo es el número de story points que un equipo puede manejar en un sprint.
- Se calcula con el promedio de story points realizados en sprints anteriores.

# Velocidad de desarrollo

## Paso 1

Se suman todos los puntos de historia realizados en el sprint.

No incluir las historias realizadas a la mitad.

**Table 11.1** *Stories completed during an iteration.*

Story	Story Points
A user can...	4
A user can...	3
A user can...	5
A user can...	3
A user can...	2
A user can...	4
A user can...	2
<b>Velocity</b>	<b>23</b>

# Velocidad de desarrollo

## Paso 2

Se calcula el promedio con los sprints anteriores.

En el ejemplo serían aproximadamente 100 puntos la velocidad de desarrollo.

Sprint 1	Sprint 2	Sprint 3	Sprint 4	Sprint 5	Sprint 6	Sprint 7
80	70	95	105	130	110	120

# Velocidad de desarrollo

## ¿Y en el primer sprint?

Una estrategia es decidir en equipo qué historias intentarán acabar en el sprint 1, pero sin el compromiso de acabar todas.

Para el sprint 2 y siguientes ya se pueden utilizar las historias terminadas en el sprint 1 para calcular la velocidad.

# Planificación de un Release

## 1. Priorizar historias de usuario

- Cuál es el riesgo de que una historia no se complete como se espera.
- Las dependencias que una historia puede tener con otra. Deseo de la historia por muchos usuarios y clientes. Deseo de la historia por un pequeño grupo pero importante.
- Cohesión de la historia en relación a otras historias.

# Planificación de un Release

## 2. Periodo de iteración y release

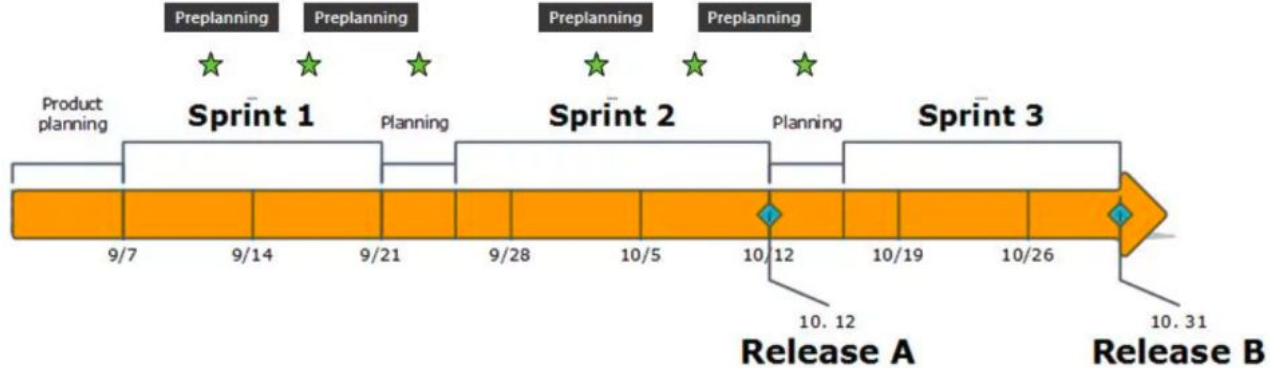
¿Cuánto demorará mi sprint?

- Normalmente pueden ser dos semanas, tres semanas, o hasta cuatro semanas.



# Planificación de un Release

## 3. Agrupar historias en Sprints y Sprints en Release



# Planificación de un Release

## Release Goals:

Release every 6 weeks

	ID	Story	Estimation	Priority
	6	As a general user I want to register a new account.	3	1
	2	As a member I want to sign-in my account.	2	2
	5	As a member I want to sign-out my account.	1	3
	10	As an administrator I want to disallow suspicious registration attempts.	2	4
Release	1	As a member I want to add new items into shopping cart.	5	5
	3	As a member I want to checkout shopping cart.	7	6
Release	4	As a member I want to track the delivery.	2	7
	9	As a member I want to record delivery addresses.	1	8
	7	As a member I want to cancel order.	1	9
Release	8	As an administrator I want to see the list of accounts logged in.	2	10
Total			26	

Velocity: 3 points / sprint

Sprint length: 2 weeks

## Release Goals:

- (1) First release after user about to register, sign-in and sign-out.
- (2) Second release after user can successfully purchase items.
- (3) Final release when all the remaining stories are done.

## First Release

Stories: 6, 2, 5, 10 -> Estimation = 8 Points  
Estimation / Velocity = 8 / 3 = 3 Sprints

## Second Release

Stories: 1, 3, 4, 9 -> Estimation = 15 Points  
Estimation / Velocity = 15 / 3 = 5 Sprints

## Third Release

Remaining stories -> Estimation = 3 Points  
Estimation / Velocity = 3 / 3 = 1 Sprint

# Planificación de una Iteración

Discutir las historias a considerar.

Desagregar las historias en tareas.

Un desarrollador acepta la responsabilidad de una tarea.

Todas las historias fueron discutidas y las tareas aceptadas.

**Table 10.1** *It's easy to track tasks, the developer doing each task, and estimates on a white board.*

Task	Who	Estimate
Code basic search screen	Susan	6
Code advanced search screen	Susan	8
Code results screen	Jay	6
Write and tune SQL to query the database for basic searches	Susan	4
Write and tune SQL to query the database for advanced searches	Susan	8
Document new functionality in help system and user's guide	Shannon	2

# Necesidad de Ajustar

La agilidad implica abrazar el cambio:

- Las estimaciones tempranas suelen no ser muy certeras, pero son necesarias.
- A medida que el proyecto avanza tenemos más elementos para mejorar las estimaciones necesarias.
- El plan del Sprint **no se ajusta**. El ajuste puede ocurrir una vez **terminado** el sprint, y sobre todo luego del sprint review y el sprint retrospectivo.

# Necesidad de Ajustar

## Tipos de Ajuste

- **Ajuste por tiempo:** Ir a negociar más tiempo para realizar lo acordado.
- **Ajuste por alcance:** Se entrega en el deadline pactado pero se negocia cuántas funcionalidades salen en el release (alcance del producto)
- **Calidad no se negocia:** Un producto de mala calidad en realidad termina costando más caro a la larga – NO es recomendable.