



# Ingeniería de Software

## 5 - SCRUM Planning

IIC2143-3

**Josefa España**

jpespana@uc.cl



# Contenidos

- 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 un 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.





# Story Points

## Determinar los puntos

Para determinar los puntos de historia nos hacemos las siguientes preguntas:

- **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.

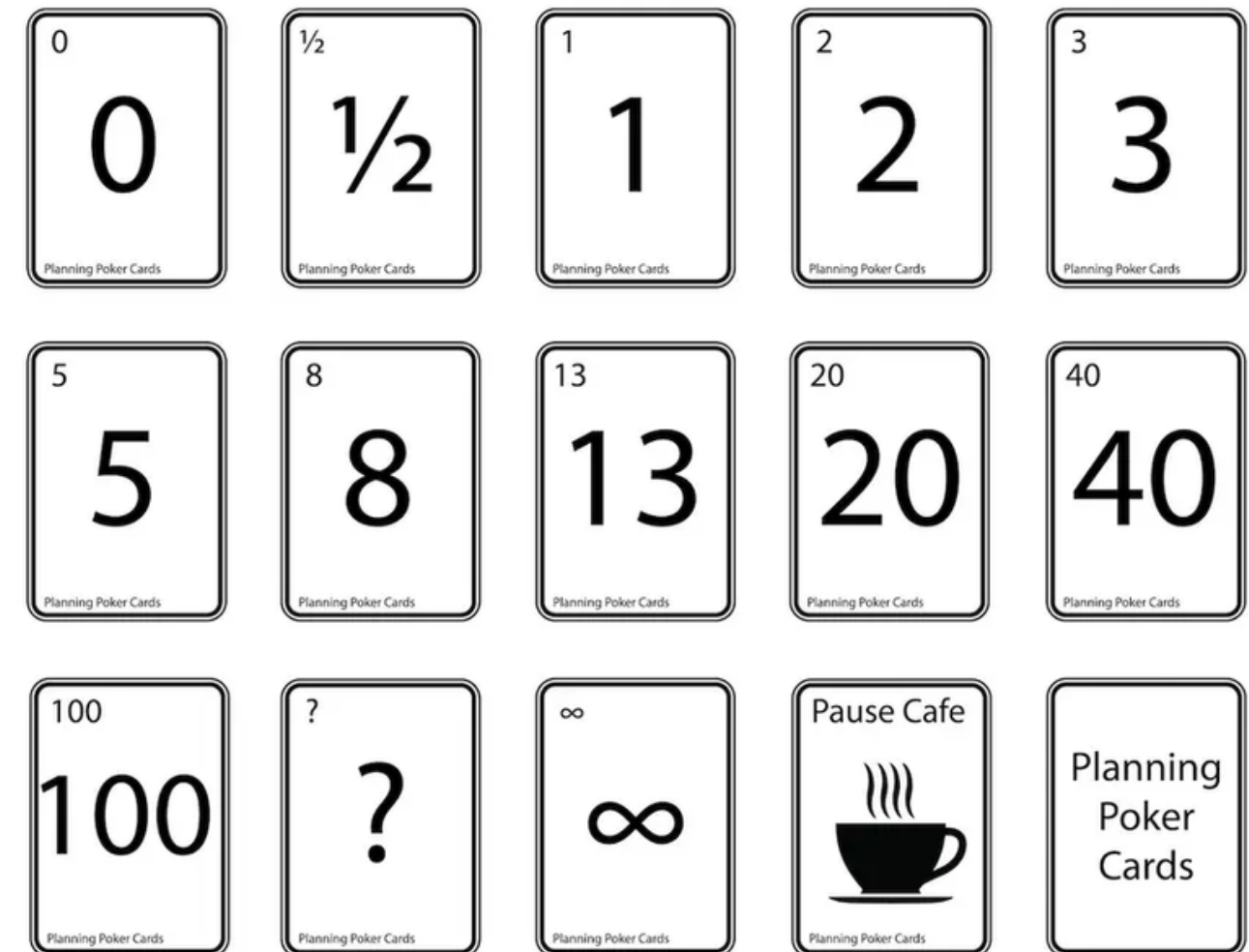
1	2	3	5	8	13
A user can...	A user can...	A user can...	A user can...	A user can...	A user can...
A user can...	A user can...	A user can...	A user can...		A user can...
A user can...		A user can...	A user can...		
A user can...			A user can...		



# 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

## Ejemplo

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
Velocity	23



# Velocidad de Desarrollo

## Ejemplo

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?

En el primer sprint es difícil de adivinar cuántos puntos de historia podemos manejar.

- 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.

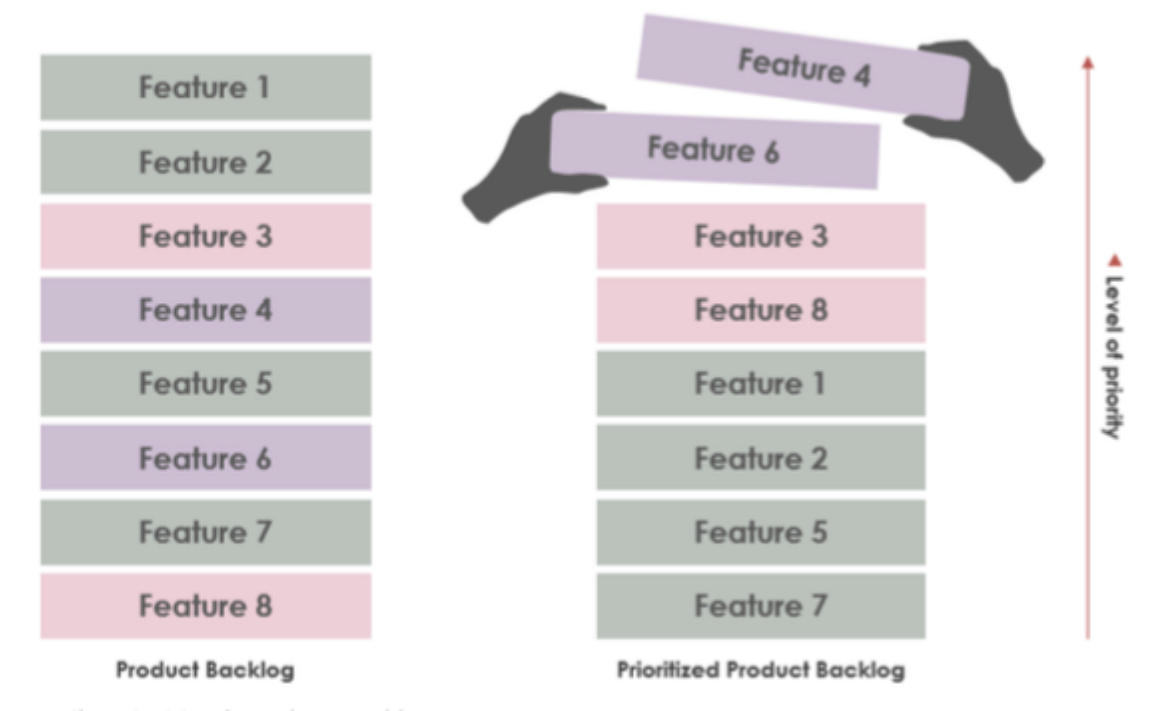


# Planificación de un Release

## 1. Priorizar historias de usuario

### Consideraciones:

- 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?

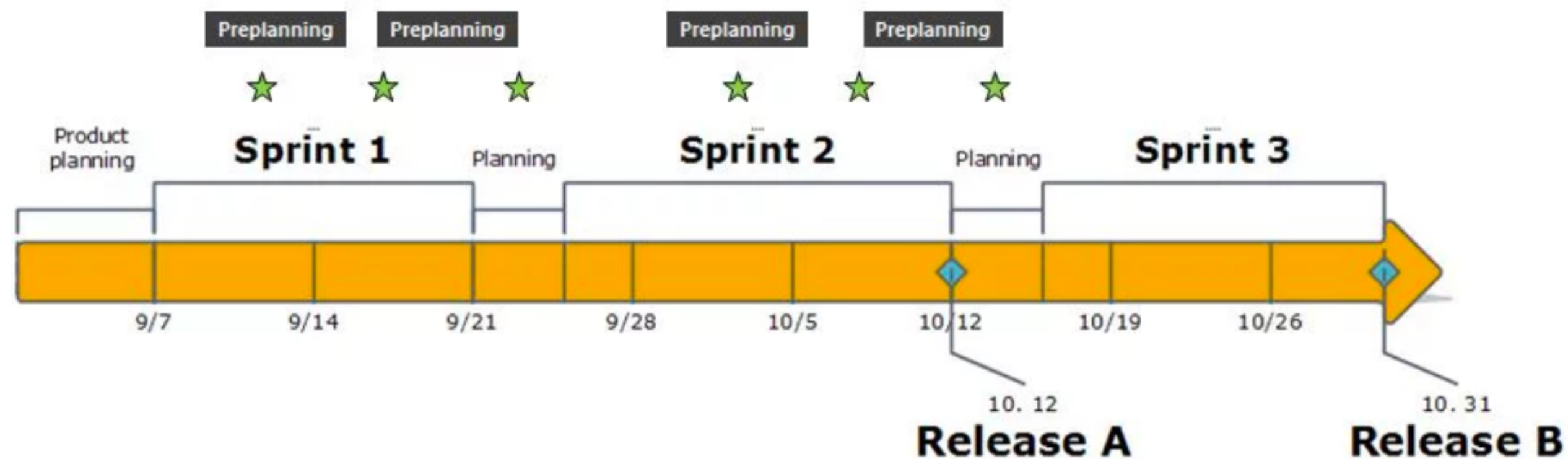
- Normalmente pueden ser dos semanas, tres semanas, o hasta cuatro semanas.





# Planificación de un Release

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





# Planificación de un Release

## Ejemplo Release Plan

<b>Release Goals:</b> Release every 6 weeks				
	<b>ID</b>	<b>Story</b>	<b>Estimation</b>	<b>Priority</b>
	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
	<b>Release</b> 10	As an administrator I want to disallow suspicious registration attempts.	2	4
	<b>Release</b> 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
	<b>Release</b> 4	As a member I want to track the delivery.	2	7
	9	As a member I want to record delivery addresses.	1	8
<b>Release Goals:</b> (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.	7	As a member I want to cancel order.	1	9
	<b>Release</b> 8	As an administrator I want to see the list of accounts logged in.	2	10
<b>Total</b>			26	
<b>First Release</b>				
Stories: 6, 2, 5, 10 -> Estimation = 8 Points				
Estimation / Velocity = 8 / 3 = 3 Sprints				
<b>Second Release</b>				
Stories: 1, 3, 4, 9 -> Estimation = 15 Points				
Estimation / Velocity = 15 / 3 = 5 Sprints				
<b>Third Release</b>				
Remaining stories -> Estimation = 3 Points				
Estimation / Velocity = 3 / 3 = 1 Sprint				

Velocity: 3 points / sprint

Sprint length: 2 weeks



# Planificar 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.



¿Consultas?



# Ingeniería de Software

## 5 - SCRUM Planning

IIC2143-3

**Josefa España**

jpespana@uc.cl