

LA PLANIFICACIÓN EN UN PROCESO ÁGIL

FAIRLEY

ACTIVIDADES DE PLANIFICACIÓN EN UN PROYECTO ÁGIL

- Trabajar con el cliente para desarrollar una visión del producto.
- Determinar la duración del proyecto y el nivel de esfuerzo a aplicar.
- Obtener el compromiso de un representante del cliente con conocimiento, de que estará involucrado permanentemente en el proyecto.
- Establecer el ambiente de desarrollo.
- Planificar la frecuencia de las iteraciones.
- Planificar la frecuencia de entrega de funcionalidad a los usuarios.

ACTIVIDADES DE PLANIFICACIÓN EN UN PROYECTO ÁGIL

- Establecer una metáfora de diseño:
 - descripción evocativa simple de cómo funciona el programa.
 - P. ej. «este programa funciona como una colmena de abejas, que salen a buscar polen y lo traen de vuelta a la colmena» como descripción de un sistema de recuperación de información basado en agentes.
- Adoptar una versión de desarrollo ágil.
- Planificar revisiones continuas con los clientes, desarrolladores y otros *stakeholders*.
- Planificar revisiones periódicas del estado del proyecto.
- Llevar adelante una evaluación y mitigación de riesgos inicial.
- Planificar evaluaciones de riesgo continuas y actividades de mitigación.

ESTIMACIONES Y PLANIFICACIÓN ÁGIL

PLANIFICACIÓN ÁGIL

- Foco en la actividad de planificación y no en el plan.
- Inicial (*realease*) y por iteración.
- Planificación tradicional: basada en actividades (WBS).
- Planificación ágil: basada en *features*.

CAUSAS DEL FRACASO DE LA PLANIFICACIÓN TRADICIONAL

1. Me paso del plazo. Causas de retrasos:
 - Las actividades no terminan antes (ley de Parkinson)
 - Los retrasos se acumulan
 - Las actividades no son independientes
2. La multitarea afecta la productividad:
 - Imposible asignar efectivamente el trabajo con antelación
 - Planificación tradicional se enfoca en mantener un alto nivel de utilización de los recursos humanos, y no en dejar suficiente holgura para la variabilidad de las tareas. Como cargar una autopista al 100 % de capacidad.
3. *Los features a desarrollar no se priorizan por valor para el cliente (no es verdad)*
4. Ignoramos la incertidumbre en requisitos y en estimaciones de esfuerzo.
 - Para reducir la 1.a: iterar y mostrarle al cliente frecuentemente.
 - Para reducir la 2.a: iterar.
5. Cada estimación tiene un margen de seguridad. No equiparar estimación = compromiso.

ENFOQUE ÁGIL

- Trabajar como un solo equipo
- Iteraciones cortas
- Entregar algo en cada iteración
- Enfocarse en las prioridades para el negocio
- Inspeccionar y adaptarse

ROLES

- *Product owner* (establece prioridades)
- Cliente (financia)
- Desarrollador (cualquier miembro del equipo)
- Gerente de proyecto (lidera). Puede ser tmb desarrollador, y ocasionalmente *product owner*

ITERACIONES CORTAS

- No hay fases:
 - Dependiendo del proyecto sí puede haber fase corta al inicio.
 - Pero una vez esté andando, todo el trabajo (requisitos, diseño, codificación, prueba, etc. se hace concurrentemente dentro de cada iteración).
- Se termina a tiempo, aunque se deba dejar de lado funcionalidad.
- Son muy cortas. Generalmente de 1 a 4 semanas, a veces hasta 3 meses.
- Iteraciones pueden ser:
 - De largo fijo
 - De largo variable, que se determina al comienzo de cada iteración.

INCREMENTO

- Es el producto de una iteración (sprint).
- Si es puesto en producción o no, lo decide el *product owner*.
- Al inicio el *product owner* plantea la meta de negocio del *sprint*. Puede que sea una liberación del *sprint*.

ENTREGAR ALGO EN CADA ITERACIÓN

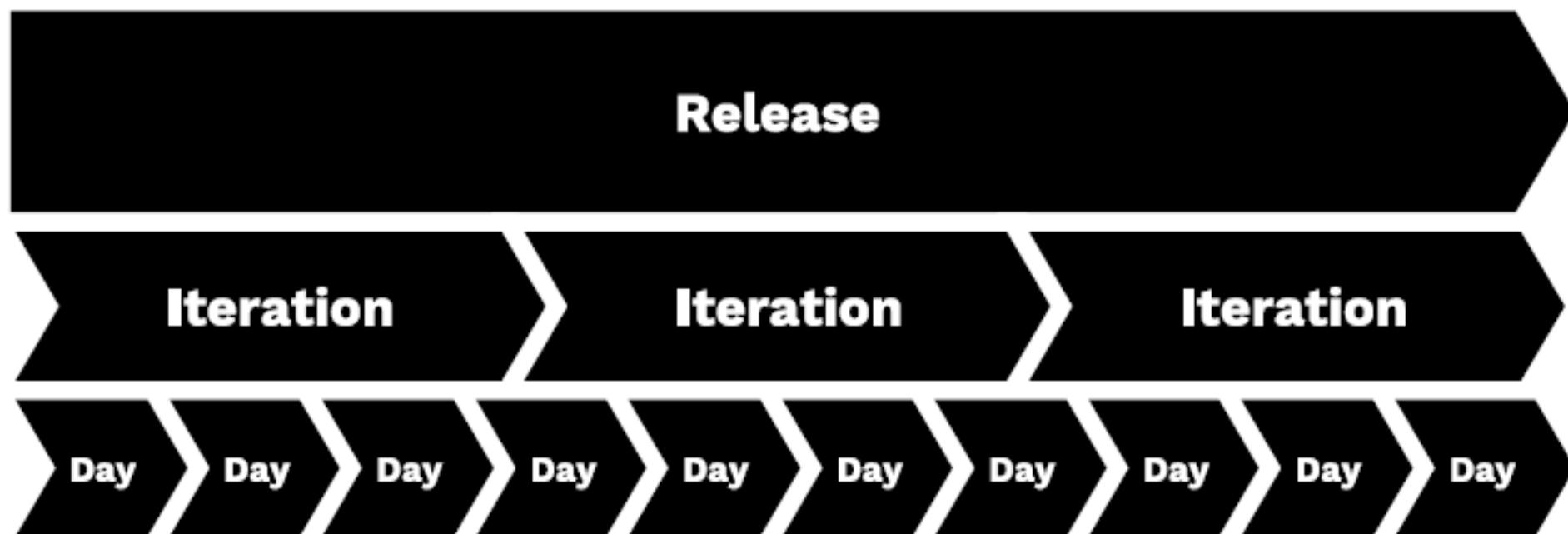
- Toda iteración debe generar valor de negocio con software funcionando.
- Incremento
 - Es el producto de una iteración.
 - En cada iteración (product potentially shippable)
- Si es puesto en producción o no, lo decide el *product owner*.
- Al inicio el *product owner* plantea la meta de negocio de la iteración. Puede que sea una liberación.
- No se puede generar suficiente funcionalidad para satisfacer al cliente → RELEASE (2 a 6 meses).
- Release sale a producción.

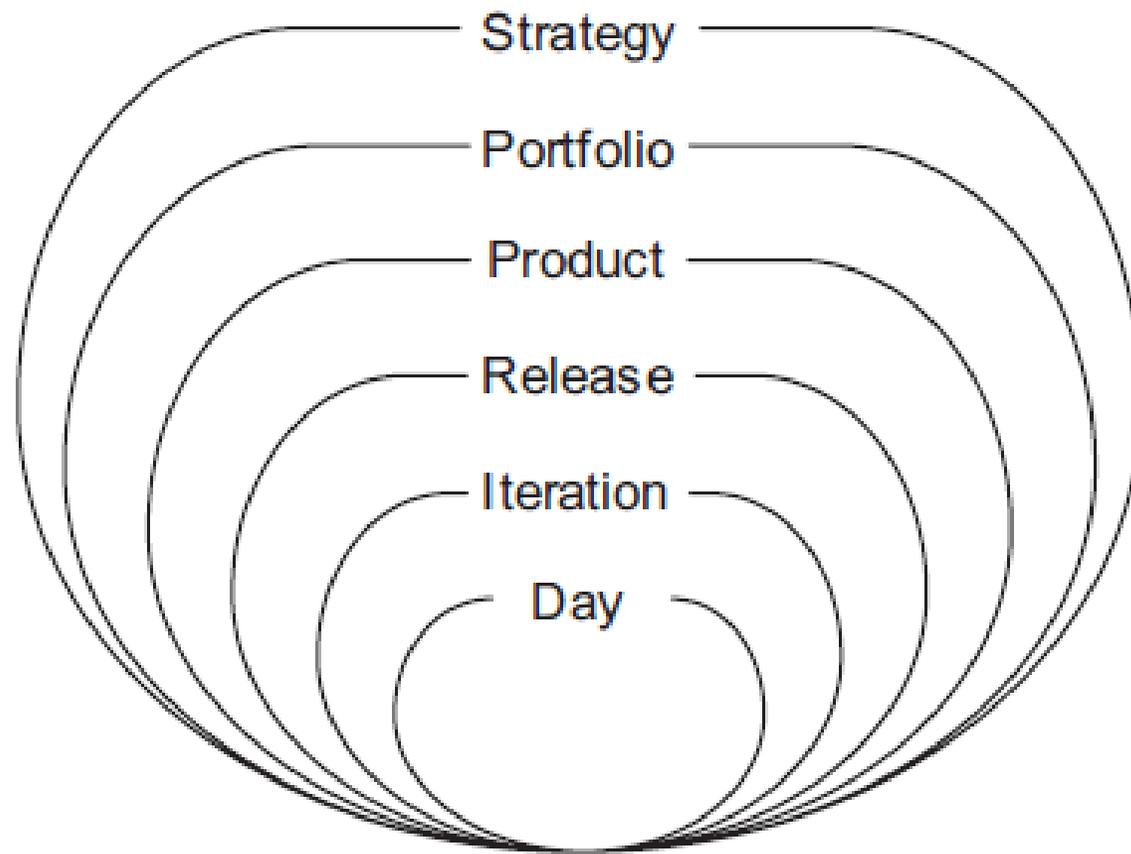
ENFOCARSE EN LAS PRIORIDADES DEL NEGOCIO

- 1. Entregar funcionalidad en el orden especificado por el *product owner* --> *release plan* (3 a 6 meses).
- 2. *Features* valoradas por el usuario -- > se pueden usar historias de usuario.
- Historia de usuario:
 - Es una descripción breve de funcionalidad como la ve el usuario o el cliente.
 - Forma libre. No hay sintaxis prescripta.
 - Puede ser útil escribirlas con este formato:

«Como <tipo de usuario>, quiero <capacidad>
de modo que <valor para el negocio>.»
 - (Puedo mantener documentación, pero el foco es oral).

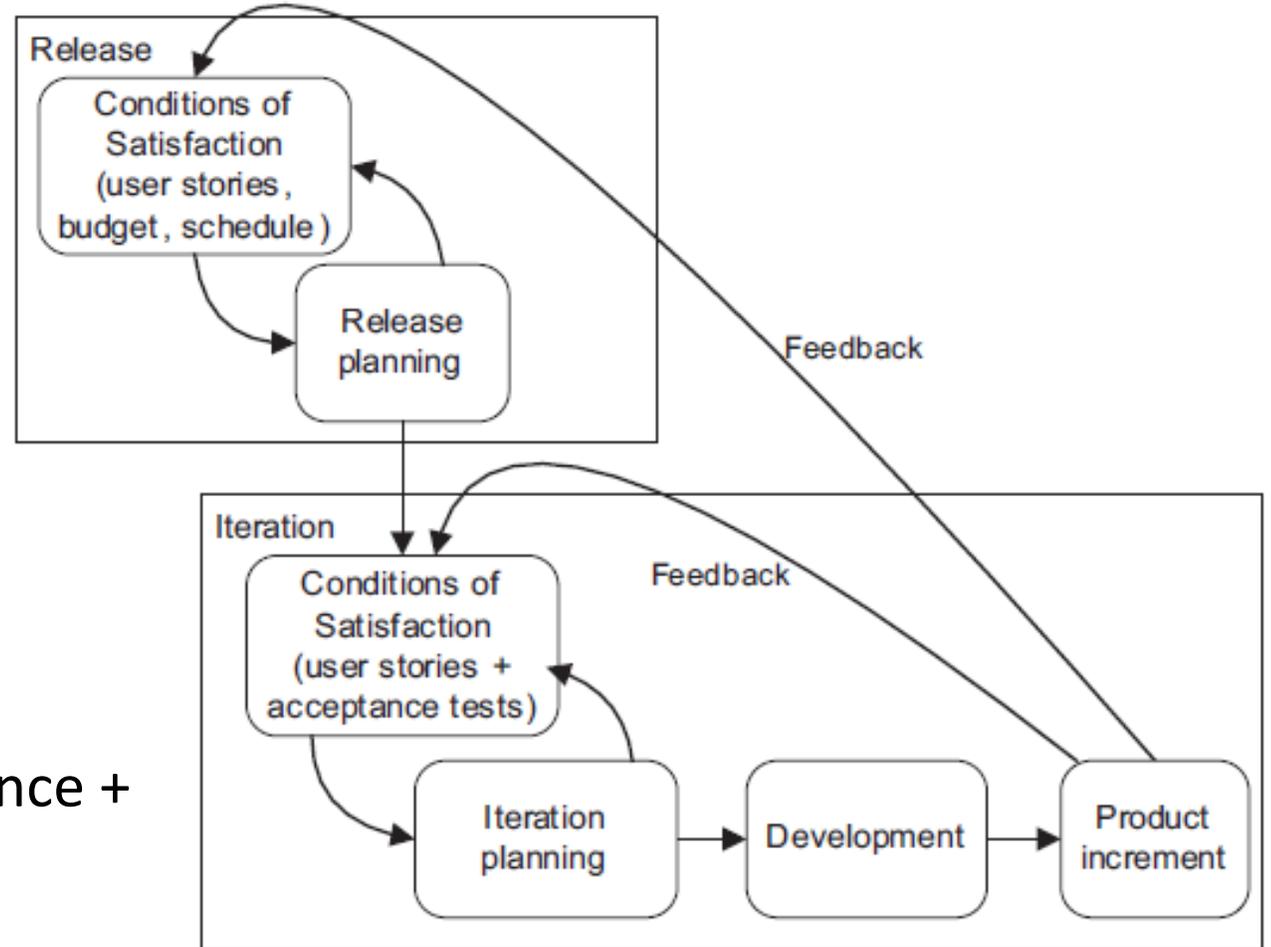
3 Niveles de planificación





PLANIFICACIÓN DE LA *RELEASE*

- Cuándo:
 - al inicio del proyecto
 - antes de cada iteración
- Qué:
 - Seleccionar *features* que va a contener la *release*.
 - establecer las condiciones de satisfacción del proyecto: alcance + plazo + presupuesto + calidad.



PLANIFICACIÓN DE LA ITERACIÓN

- Se establecen las condiciones de satisfacción del *product owner* para la iteración:

features que le gustaría que se desarrollaran a continuación

+

algunas pruebas de alto nivel para cada *feature*.

- Se define el criterio de calidad: «*definition of done*».

PLANIFICACIÓN DIARIA

- Reunión diaria para coordinar el trabajo, sincronizar esfuerzos diarios.
- Se hacen, evalúan y revisan los planes.
- Horizonte de planificación: un día.
- Se enfoca en planificar las tareas y coordinar las actividades individuales que llevarán a completar la tarea.

OTROS NIVELES DE PLANIFICACIÓN

- Planificación del producto:
 - el *product owner* mira más allá de la release actual y planifica la evolución del producto o sistema.
- Planificación del portafolio:
 - selección de los productos que mejor implementarán la visión.
- Planificación estratégica:
 - establece la visión.

SPRINT

- ~~Sprint 0~~

- Es necesario prepararse antes del comienzo de los *sprints*.

INSTANCIAS DE PREPARACIÓN



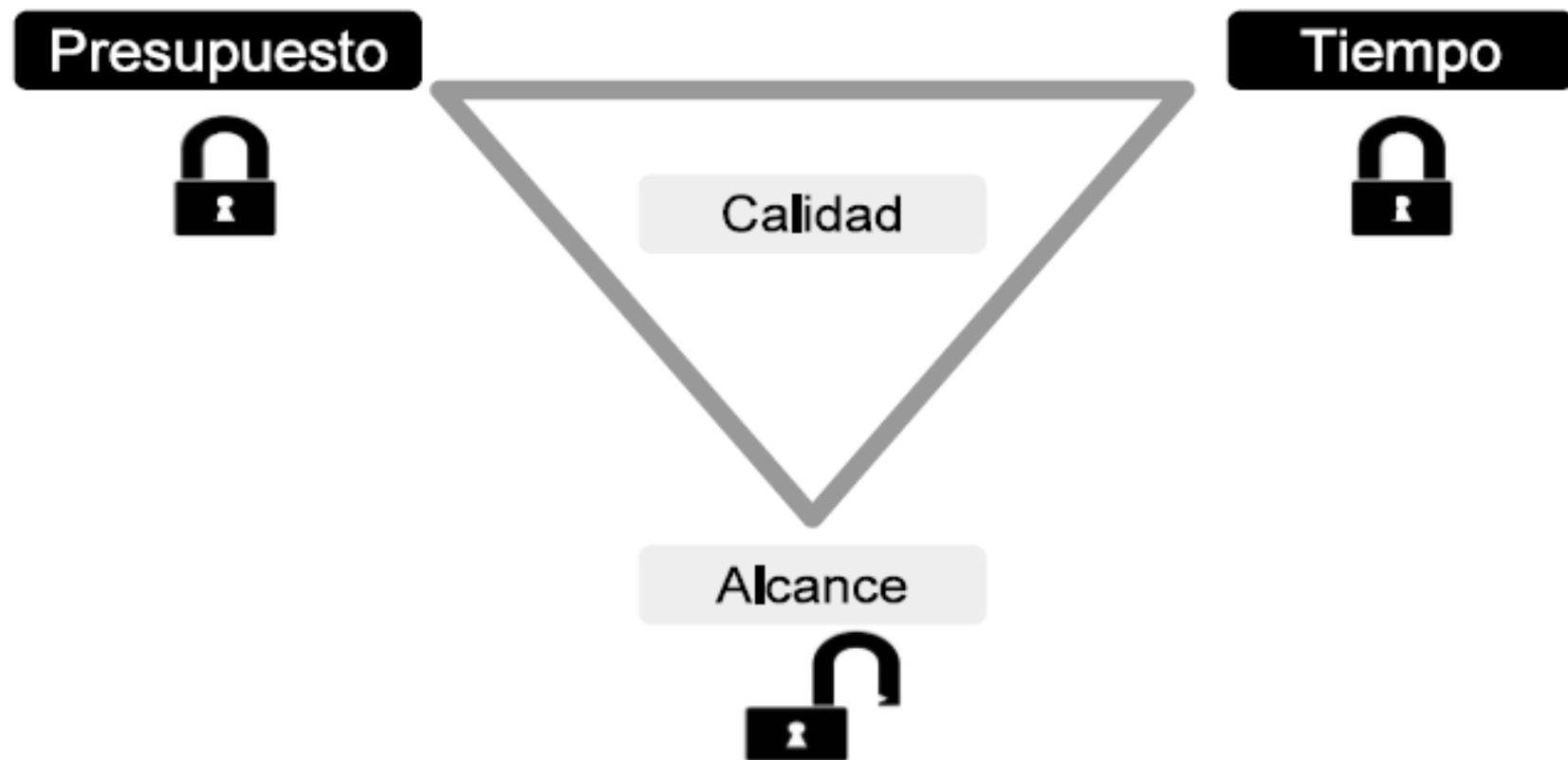
INCEPTION DECK

- Taller de 2 días.
- Objetivos:
 - Descripción del producto.
 - Visión clara del PO o del sponsor.
 - Definir la viabilidad del proyecto:
 - Involucrados del producto
 - Involucrados del hacer (alguien que sabe)
- Si la respuesta es sí → *Kick-off*

STORY MAP

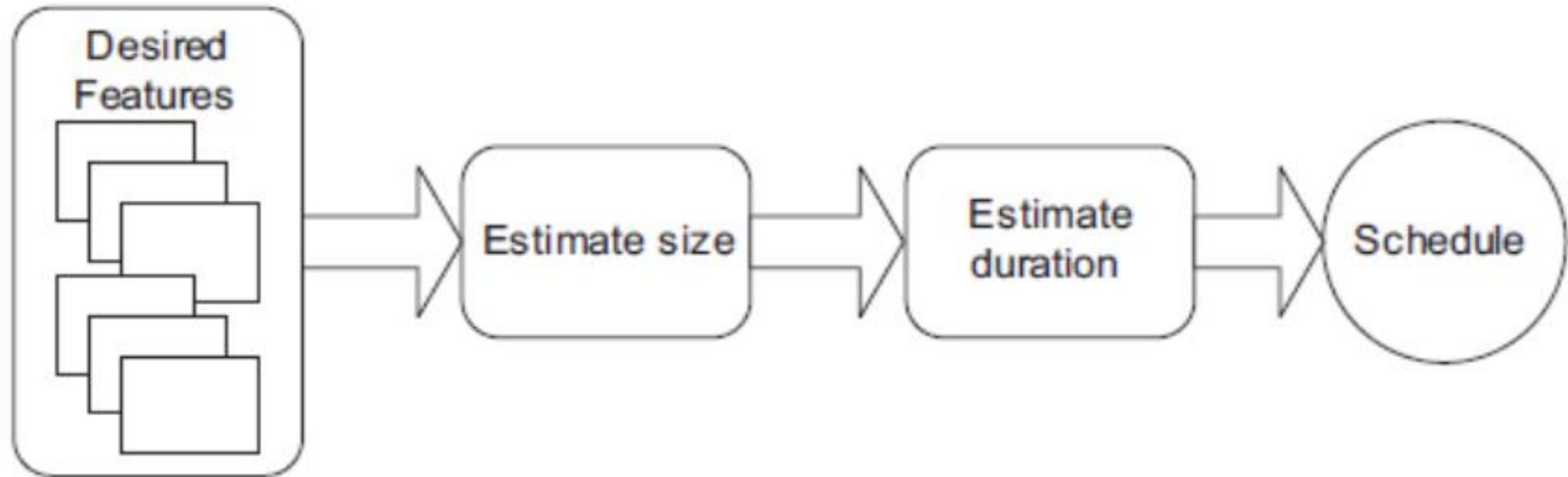
- Relevamiento de alto nivel de objetivos de uso del usuario.
- Historias (u otros artefactos).

Limitantes en la gestión ágil



ESTIMACIÓN

ESTIMACIÓN



UNIDADES USADAS EN ÁGIL

- Ideal time



- Story points



PUNTOS DE HISTORIA

PUNTOS DE HISTORIA



Definición

Representan el esfuerzo de construcción que deberá emplear el equipo y que está influenciado por la complejidad, incertidumbre, riesgos, etc.

Establecer magnitud

Usar un ancla en la escala de puntos según los diferentes ítems del backlog que puedan demandar diferentes grados de esfuerzo del equipo

Escala relativa

Este método es fundamentalmente de estimación relativa

1,2,3,5,8,13,21

- No pensar esto en horas. La hora solo funciona con preasignación.

PUNTOS DE HISTORIA

- Son relativos. P. ej. porciones en un restaurante.
- Representan el tamaño en conjunto de la historia. Comprenden:
 - la cantidad de esfuerzo de desarrollo,
 - la complejidad el desarrollo,
 - incertidumbre,
 - el riesgo, etc.
- Dos formas de empezar:
 - asignarle 1 a la más chica
 - asignarle un valor medio a una de medio tamaño.
- Todas en mismo orden de magnitud.
 - Entre 1 y 10
 - Entre 10 y 100. (No asignar 66 y 67).

DEFINITION OF «DONE»

En el mundo ideal cuando se estima un Story point la estimación debe estar basada en un Definition Of Done

- Mike Cohn



Story A

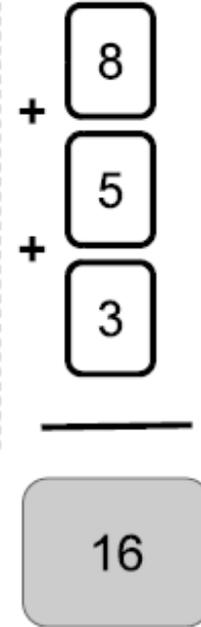
Como [usuario] quiero
[funcionalidad] para
obtener [beneficio]

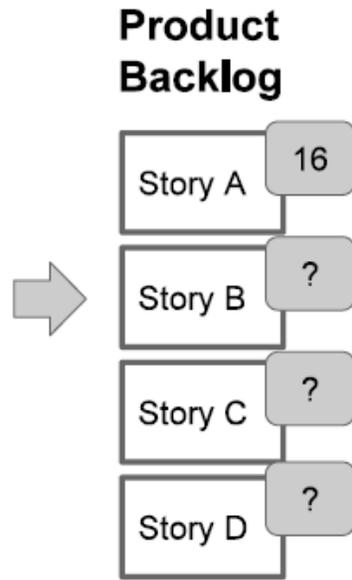
¿sp?



Definition of Done:

- Code
- Review
- Dev QA

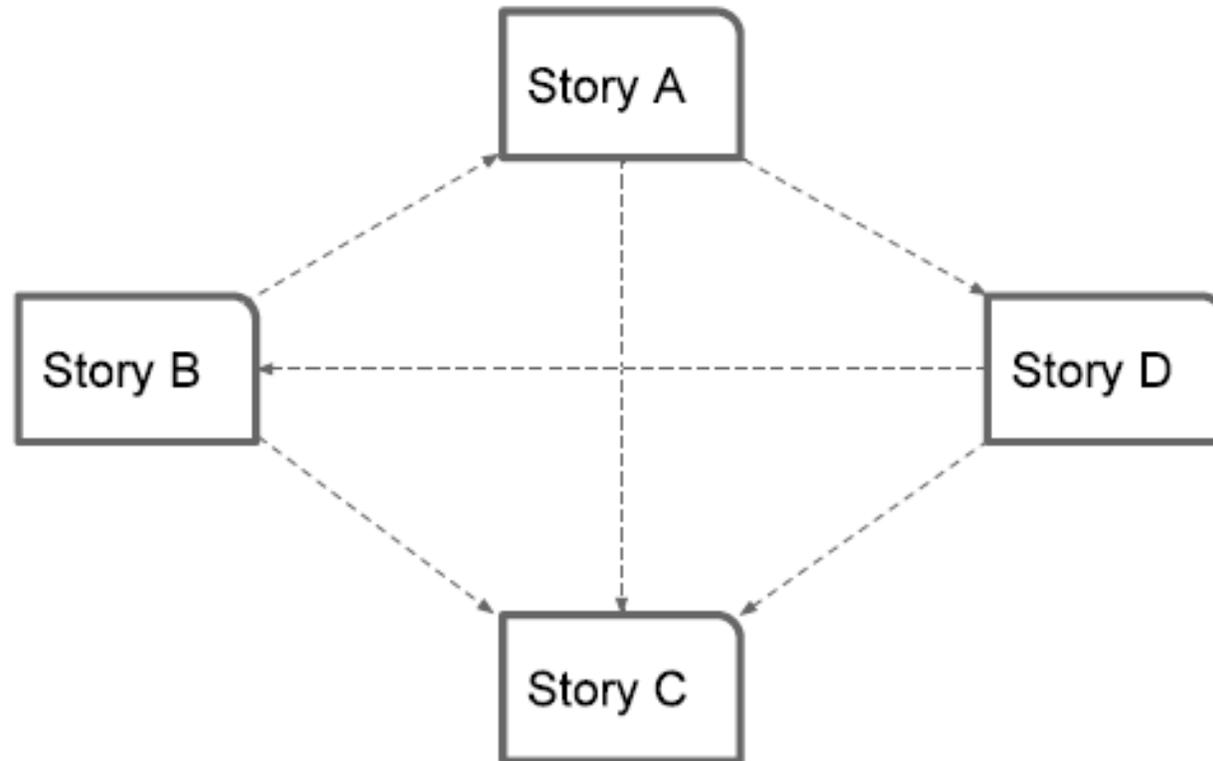




Estimado el tamaño de A, ¿Qué tamaño tienen **B**?

La estimación de cada story debe ser **relativa**, por lo tanto se debe comparar el esfuerzo que llevará construir una con otra

TRIANGULAR



IDEAL TIME

IDEAL TIME



Definición

Es el tiempo de trabajo sin interrupciones y suponiendo que todo lo que necesitas siempre está disponible.

Factor de foco (FF)

El **ideal time** tiene un cálculo de corrección a través del FF.

El FF al iniciar un proyecto se asume que es **~75%**.

Ajustes de Velocidad

Con este método la unidad de **velocidad** es el **ideal time**.

Cuando avanzan los sprint hay que ajustar la velocidad por un nuevo FF.

IDEAL TIME VS. ELAPSED TIME

- Ideal time != elapsed time (P. ej. partido de fútbol o basketball).

- En

Table 5.1 Factors affecting ideal time

Supporting the current release	Training
Sick time	Email
Meetings	Reviews and walk-throughs
Demonstrations	Interviewing candidates
Personnel issues	Task switching
Phone calls	Bug fixing in current releases
Special projects	Management reviews

- Overhead
- Interrupciones (gerentes cada 5 min, desarrolladores cada 15 min)
- Multitarea – pérdida de eficiencia

ESTIMACIÓN EN DÍAS IDEALES

- Asumimos que:
 - La historia que estamos estimando es lo único en lo que estaremos trabajando
 - Todo lo que necesitemos va a estar a la mano al comenzar
 - No habrá interrupciones
- Estimamos la cantidad de días ideales que llevará desarrollar, probar y aceptar la historia.
- No será necesario considerar el impacto del ambiente.

IDEAL TIME

- *Ideal Time* (IT) = cantidad de horas de trabajo sin ningún tipo de distracciones.
- P. ej. 3 personas x 15 días = 45 días
- IT = 45 días ideales

DURACIÓN

- Puedo usar días ideales como medida de tamaño de las historias y convertirlas en duración con la velocidad.

VELOCIDAD

- En este método es la cantidad de desarrollo de software aceptado que puede hacer todo un equipo en una iteración.
- Medido en cantidad de **días-hombre** de rendimiento.
(Pueden ser en días-hombre u horas-hombre).



- $FF(1) = 75\%$ (valor inicial estadístico)
- Velocidad estimada (VE) = $IT \times FF(1)$
- P. ej. $VE = 45 \times 0,75 = 33,75$

$$VE = 34$$

Quiere decir que en una iteración de este equipo tengo un rendimiento de trabajo equivalente a 34 días «estimados».

VELOCIDAD REAL

La **VR** va a estar dada luego que hayamos completado todas las tareas que tomamos en la planificación.

- Ej. VR (velocidad real) = 30
- En el ejemplo, el equipo completó las tareas planificadas en un tiempo menor a 34 días estimados. Sobró tiempo. Incluye todo.





AJUSTE DE FF

- Cada **dos sprints** ajusto el FF según la velocidad real (VR)
- $VR / VE = FF (n)$
- $30 / 34 = 0,8882$
- $FF (n) = 88 \%$

- Calculo la VE con el nuevo FF (n):
- P. ej. $VE = 45 \times 0,88 = 39,6$

VE = 40

La velocidad estimada para el siguiente sprint mejoró, el sprint pasó a equivaler **40** días.

En el siguiente *sprint* voy a tomar *backlog* por este nuevo valor. Cada persona estima si lo que tienen que hacer entra o no en esos días estimados.

PREASIGNACIÓN

En la práctica del método Ideal Time es común que los equipos decidan pre-asignar en la instancia de planificación.

No calcular días ideales para ningún rol específico.

TÉCNICAS DE ESTIMACIÓN

ESTIMACIONES ENTRE TODOS

- No sabemos quién va a realizar la tarea.
- Otros pueden tener algo que decir.

ESCALA DE ESTIMACIÓN

- Un orden de magnitud:
 - 1 a 10
 - 10-100. Preidentificar los números. No una de 66 y otra de 67 (falso nivel de precisión).
- Dos escalas de estimación:
 - 1, 2, 3, 5, y 8 (Fibonacci)
 - 1, 2, 4, y 8
- Incluir el 0 para features que no llevan trabajo:
 - Para mantener todas las features en un rango de 10. Si le asigno algo distinto de 0 a las chicas, las grandes se van de rango.
 - Si el trabajo está más cerca de 0 que de uno, no quiero que esa tarea sume a la velocidad, porque en la próxima iteración ese punto me va a costar más.

HISTORIAS, TEMAS Y ÉPICAS

- Épica:
 - Una historia de usuario grande
 - Para features que no estamos seguros de querer o que pueden no quedar.
 - Queremos una estimación preliminar sin poner mucho esfuerzo.
- Tema:
 - Un conjunto de historias de usuario relacionadas.
 - Se combinan y tratan como una sola entidad al estimar o planificar la release.
 - Una épica, solo por su tamaño, es un tema.

HISTORIAS, TEMAS Y ÉPICAS

- Al usar temas y épicas, se reduce el esfuerzo de estimación.
- Su estimación será menos precisa que la de historias de usuarios.
- Las historias a desarrollar en las próximas iteraciones deben ser suficientemente pequeñas para ser completadas en una sola iteración y estimadas en un solo orden de magnitud (1, 2, 3, 5 y 8).
- Temas y épicas (de ejecución más distante) pueden ser estimadas en otro orden de magnitud (13, 20, 40 y 100).

TÉCNICAS

- Juicio de expertos
- Estimación por analogía
- Desagregación
- Solas o combinadas (p. ej. *Planning Poker*)

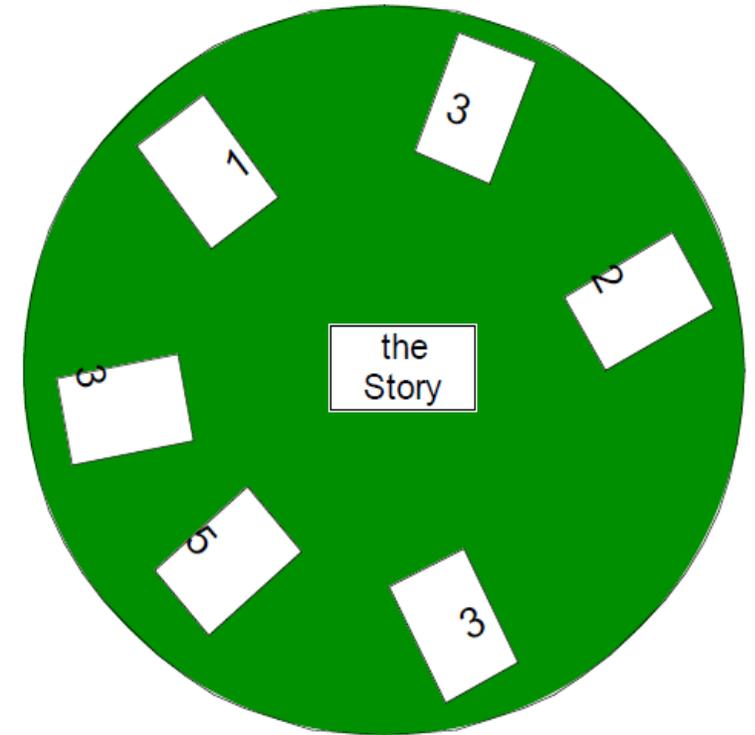
PLANNING POKER

PLANNING POKER

- Es usada con varias metodologías
- El equipo de trabajo completo participa en la estimación de las historias
 - Cliente lee la historia
 - Si es necesario, hace aclaraciones
 - Cada integrante elige una carta con valor de estimación (1,2,3,5,7,10, infinito)
 - Todos dan vuelta la carta a la vez

PLANNING POKER (2)

- Si todos estiman lo mismo, ya está.
- Si no, los que estimaron en los extremos dan sus argumentos.
- Se repite el juego hasta que converge.
- Si no, se posterga, divide o se toma la estimación más baja.



PLANNING POKER (3)

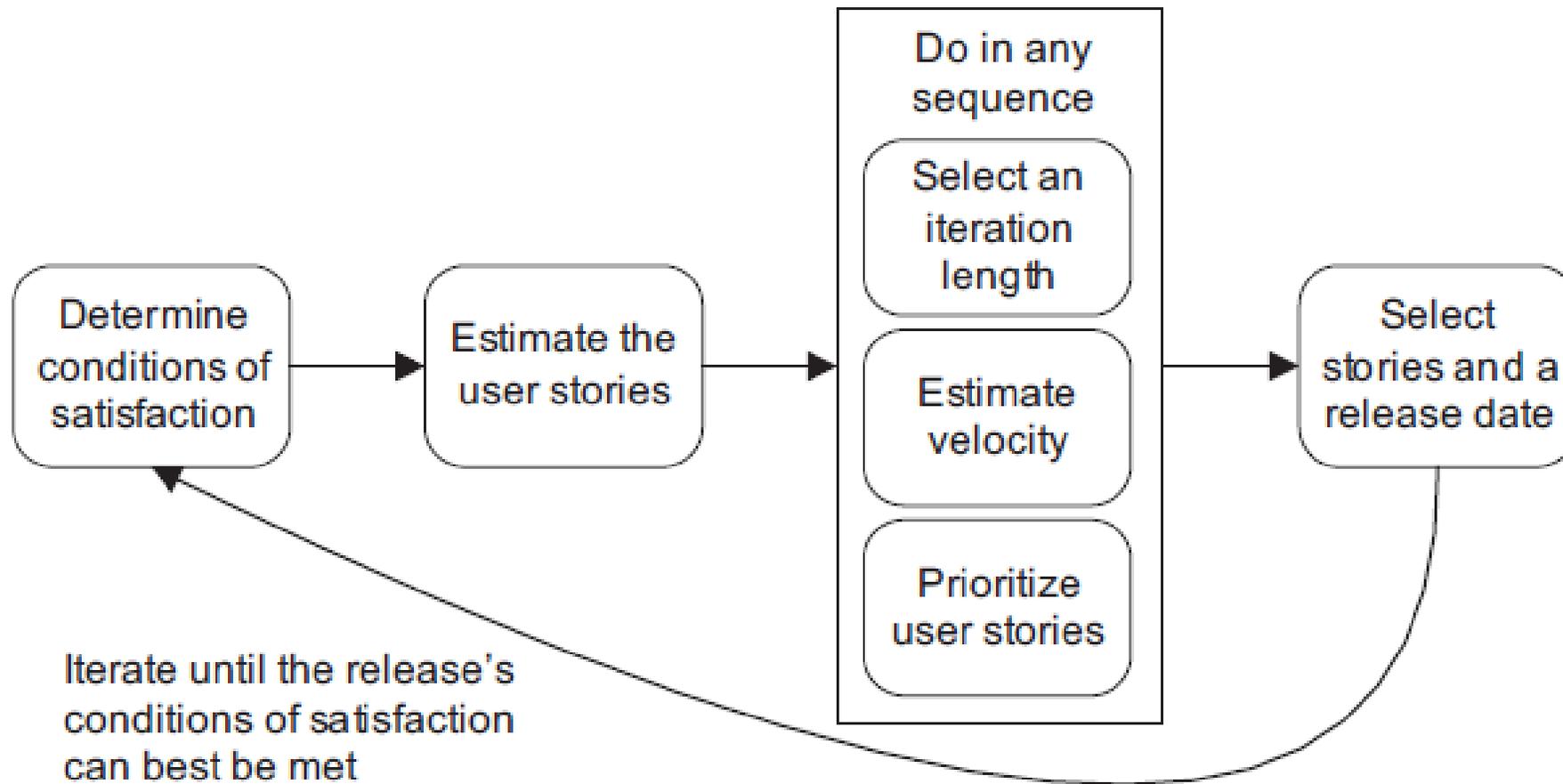
- Rápidamente se logra estimación
 - Cuando hay acuerdo es un minuto
 - Si no hay acuerdo requiere discusión y esta tiene lugar
- Todo el equipo participa
- Implementación eficiente (y divertida) del método Delphi

ESTABLECER EL CRONOGRAMA

ESTABLECER EL CRONOGRAMA

- Planificar la release
- Planificar la iteración
- Seleccionar un largo de iteración apropiado.
- Estimar la velocidad.

PLANIFICAR LA *RELEASE*



PRIORIZACIÓN DE TEMAS

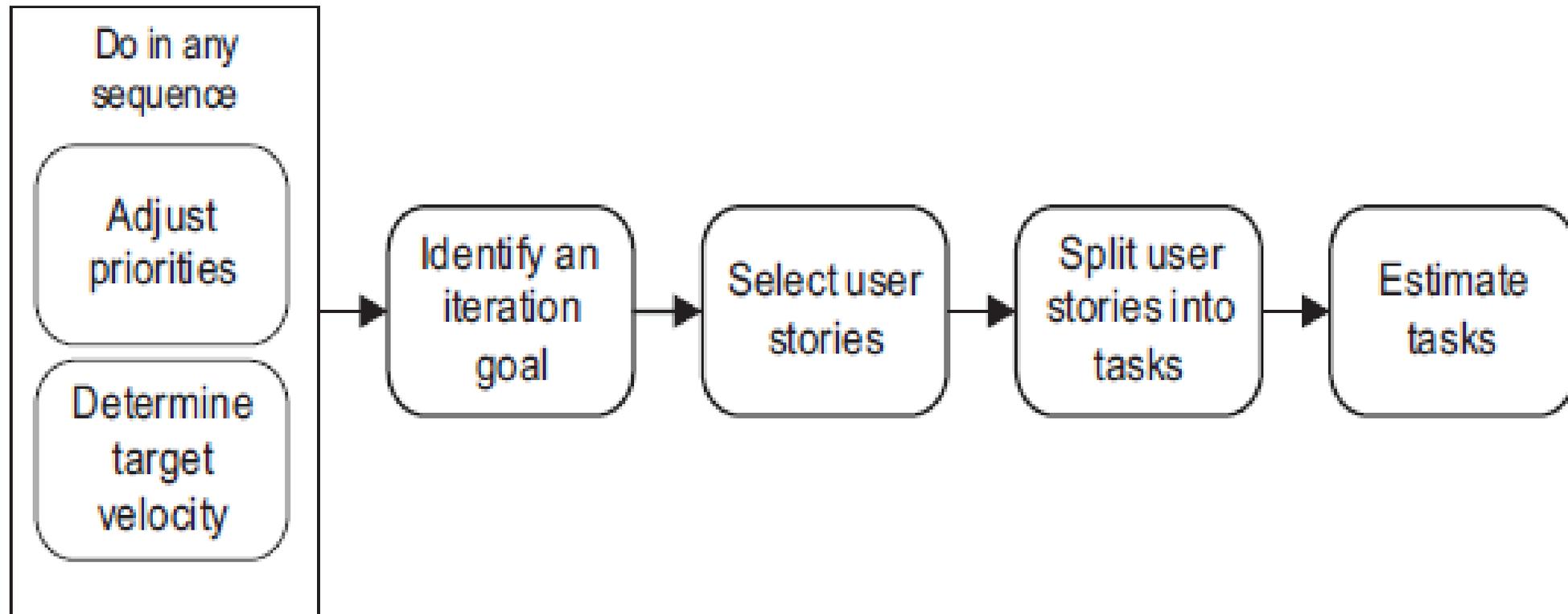
- Factores a considerar
 1. El valor financiero de tener esas features.
 2. El costo de desarrollo
 3. La cantidad e importancia del aprendizaje al desarrollarlas.
 4. La cantidad de riesgo que se elimina.

REFINAMIENTO DE HISTORIAS

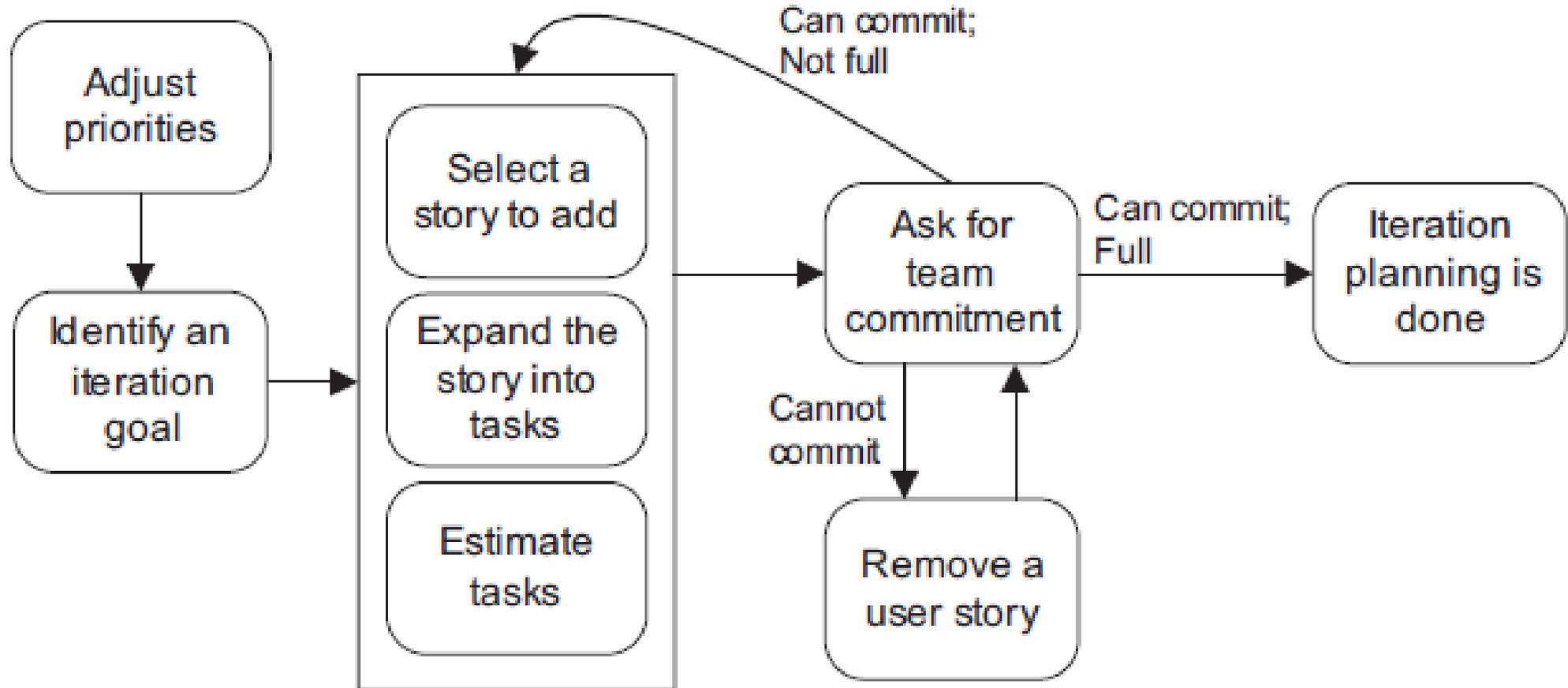
- Partir historias cuando
 - Es más larga que la iteración
 - Solo puedo implementar una versión parcial
 - Es una épica y necesito una estimación más precisa

PLANIFICAR LA ITERACIÓN

PLANIFICACIÓN DE LA ITERACIÓN EN BASE A LA VELOCIDAD



PLANIFICACIÓN DE LA ITERACIÓN EN BASE A COMPROMISOS



SELECCIONAR UN LARGO DE ITERACIÓN APROPIADO

ESTIMAR LA VELOCIDAD

VELOCIDAD

Es la suma de story points de las Stories (features) que fueron aprobadas por el Product Owner al finalizar el Sprint.

Es una medida que indica **cuánto incremento de producto podemos construir en cada sprint.**

Medida de tasa de avance del equipo

- En el primer *sprint* el equipo arriesga, y después, en el segundo, va a intentar identificar el tamaño de lo que el equipo quiere hacer.
- Los que usan esta técnica se basan en estimación de expertos.
- En la retrospectiva: cuando planificaron ¿en qué pensaron? ¿cómo abstraieron el tamaño para decir que estas *x story points* iban a entrar en el *sprint*?
- Para estimar tamaño, estimar historias de igual tamaño (*T-shirt sizing* o rompecabezas).
- En el siguiente *sprint* puedo meter más cosas.

CALCULAR ITERACIONES Y DURACIÓN TOTAL

- Puedo
 - sumar el total de puntos de historia de la *release*,
 - dividir por velocidad (anterior o estimada) en una iteración de duración fija y calcular cantidad de iteraciones.
 - Y la duración del proyecto.



EJERCICIO

- Total: 100 HP
- Velocidad: 11 HP
- Iteraciones de 2 semanas
- ¿Cuántas iteraciones?
 - 9,1 iteraciones
- ¿Cuánto dura?
 - 19 semanas



EJEMPLO

Sprint Backlog

Story A	16
Story B	8
Story C	26
Story D	11

Llegamos a
una Velocidad
Estimada:
VE = 61 sp

EJEMPLO



Sprint Backlog

Story A	16	✓
Story B	8	✓
Story C	26	✓
Story D	11	✗

El Sprint terminó,
calculamos la Velocidad
del sprint:

$$\Sigma (\text{sp planificados}) - \Sigma (\text{sp rechazados})$$

$$V = 61 - 11 = \mathbf{50}$$

¡Puedo corregir estimación de duración sin reestimar, solo ajustando la velocidad!

¿Cómo contemplamos el trabajo que no está directamente relacionado con features pero sí son parte del producto?

Todo trabajo realizado para el producto requiere **esfuerzo** y el tamaño de lo que construimos lo medimos en función del esfuerzo que nos lleva

CAPACIDAD

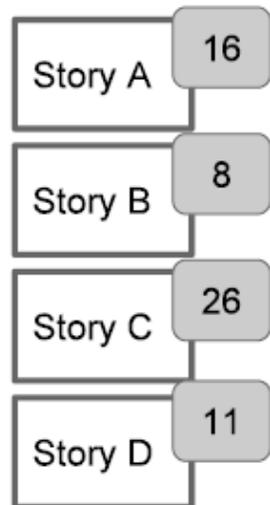
Representa el **esfuerzo** que puede invertir el equipo en un sprint.

Se puede calcular sumando la Velocity más los puntos de otras tareas “no planeadas” que se ejecutaron dentro de la iteración.

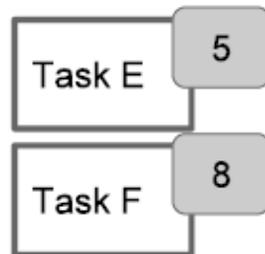
$$CP = V + \text{tareas-no-planeadas}$$



Sprint Backlog



Tareas no planeadas



Datos

$$\text{tnp} = 13$$

$$\text{VE} = 61$$

$$\text{VR} = 50$$

Fórmula capacidad:

$$\text{CP} = \text{VE} + \text{tnp}$$

Cálculos:

$$\text{CP} = 61 + 13$$

$$\text{CP} = 74 \text{ story points}$$

VE : Velocidad Estimada = Stories planificadas

VR : Velocidad Real = Stories aceptadas en el Review

tnp : Tareas no planeadas = Puntuadas al finalizar el Sprint

74

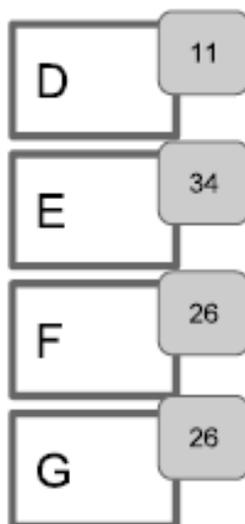
Representa el **esfuerzo** de las tareas planificadas y las no planificadas del sprint



Planning Sprint N+1



Product Backlog



Datos sprint N:

VR = 50 story points

VE = 61 story points

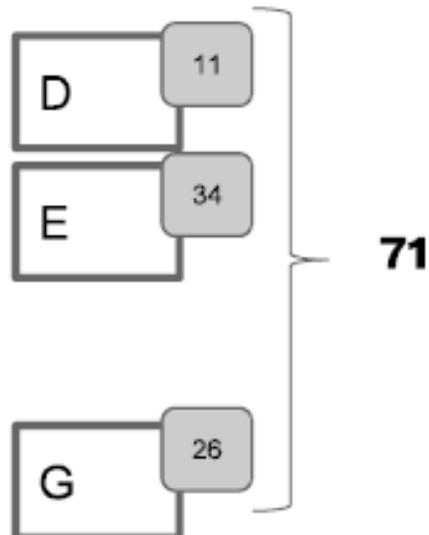
CP = 74 story points

$$D + E + F = 71 \quad G + D + E = 71$$

$$E + F + G = 86 \quad \dots$$

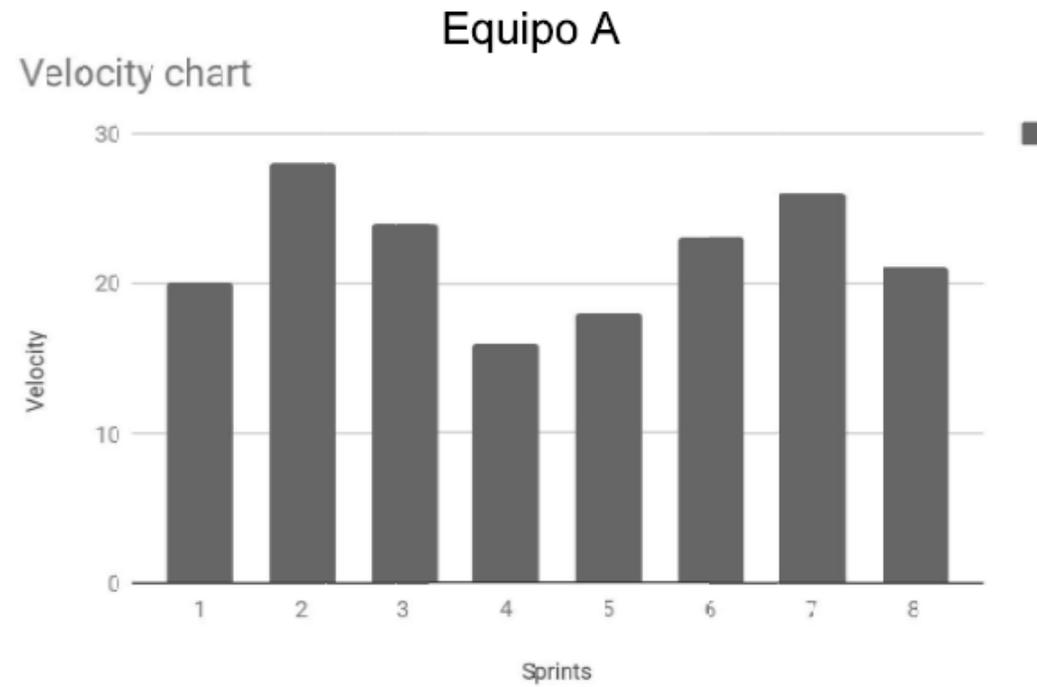
$$F + G + D = 63$$

Sprint Backlog



AJUSTE DE VELOCIDAD

EJEMPLO



Totales:

176 story points

8 sprints



Calculamos la desviación estándar (S) de velocidad:

$$S = \sqrt{\frac{\sum (x_i - \bar{x})^2}{N-1}}$$

x = Velocidad de cada sprint

N = Cantidad de sprints

$$S = \sqrt{\frac{82}{7}}$$

$$S = \sqrt{11,7}$$

$$\mathbf{S = 3,4}$$



- Calculamos la velocidad promedio:
- $Avg (V) = 176 / 8 = 22$
- Dividimos la desviación estándar 3,4 entre el promedio de la velocidad:
- $3,4 / 22 = 0,15 \rightarrow 15 \%$
- 15 % es el ajuste de velocidad que hay que considerar para cualquier velocidad estimada.



15%

Variación estándar relativa de velocidad sobre
1 proyecto y 1 equipo

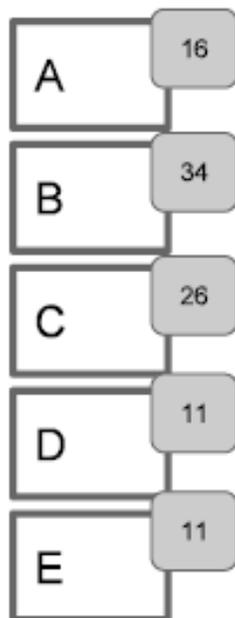
**¿Cómo y cuándo
uso este dato?**



Release planning



Product Backlog



Datos:

VE = **61 story points** = 100%

VE mínima = 100% - **19%** = 81%

VE máxima = 100% + **19%** = 119%

VE(min) = $61 \times 0,81 = 49,41 \Rightarrow$ **49 story points**

VE(max) = $61 \times 1,19 = 72,59 \Rightarrow$ **73 story points**

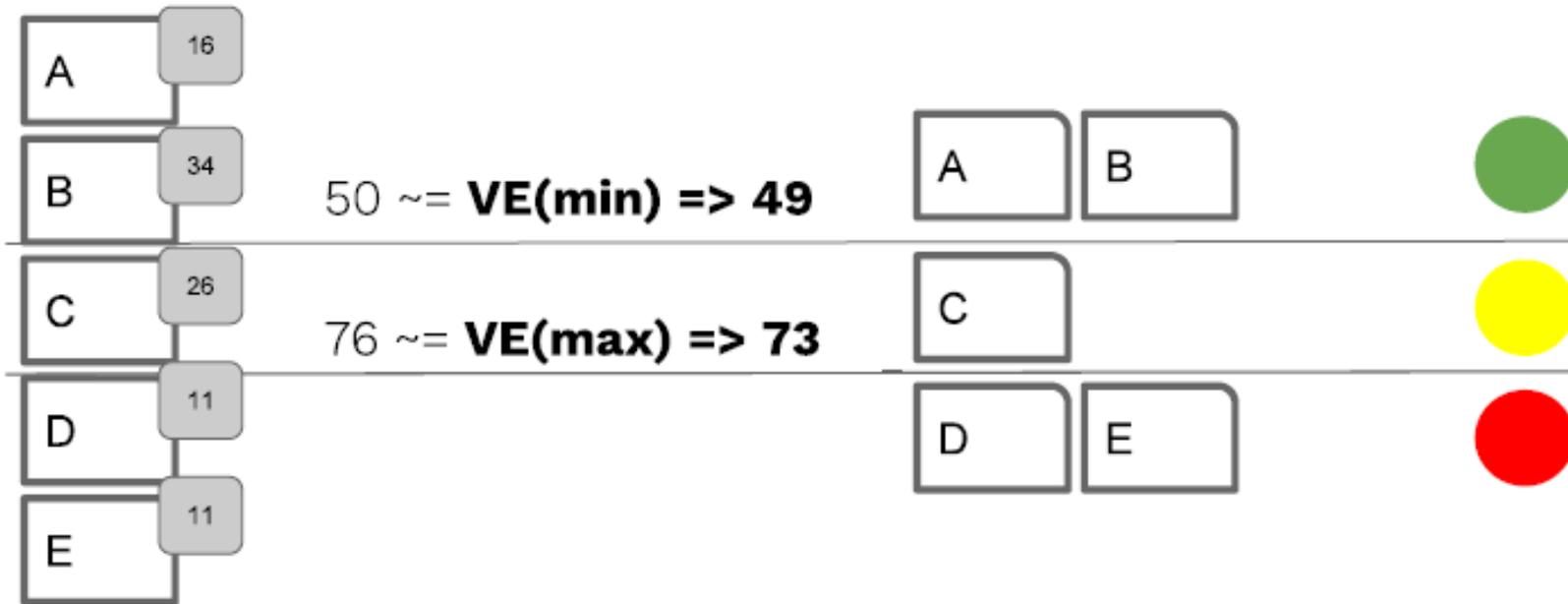
A,B,C,D,E ya priorizados



Release planning



Product Backlog





**Buscar ser
predecible en
Velocidad**

REUNIÓN DE RETROSPECTIVA

REUNIÓN DE RETROSPECTIVA

Es el momento indicado para reflexionar acerca de cómo trabajamos y proponer nuevas hipótesis de trabajo que pueden ser puestas a prueba en el siguiente sprint

REFERENCIAS



- Libro: *“Agile Estimating and Planning” (First edition)* - Autor: Mike Cohn - Editorial: Prentice Hall.
- Paper: *“Scrum Metrics for Hyperproductive Teams: How They Fly like Fighter Aircraft”* - Autores: Jeff Sutherland, PhD y Scott Downey - Link: <https://www.agilealliance.org/wp-content/uploads/2016/01/ScrumMetricsAgile2012.pdf>
- Libro: *“Software Engineering Economics”* - Autor: Barry Boehm - Editorial: Prentice-Hall.
- Libro: *“Requirements Engineering for Software and Systems”, Second Edition* - Autor: Philip A. Laplante - Editorial: CRC Press.

Gabriel Ledesma
Senior Agile Coach at WyeWorks



@gafaled



<https://www.linkedin.com/in/gabrielledesma/>

CRÉDITOS

SELECCIÓN DEL LARGO DE LA ITERACIÓN

- Factores que influyen:
 - La duración de la *release*
 - La cantidad de incertidumbre
 - La facilidad con que se obtiene retroalimentación
 - Cuánto tiempo pueden quedar las prioridades sin cambio
 - Cuán dispuestos estamos a seguir sin retroalimentación
 - El *overhead* de iterar
 - Si se mantiene un sentido de urgencia

ESTIMAR VELOCIDAD

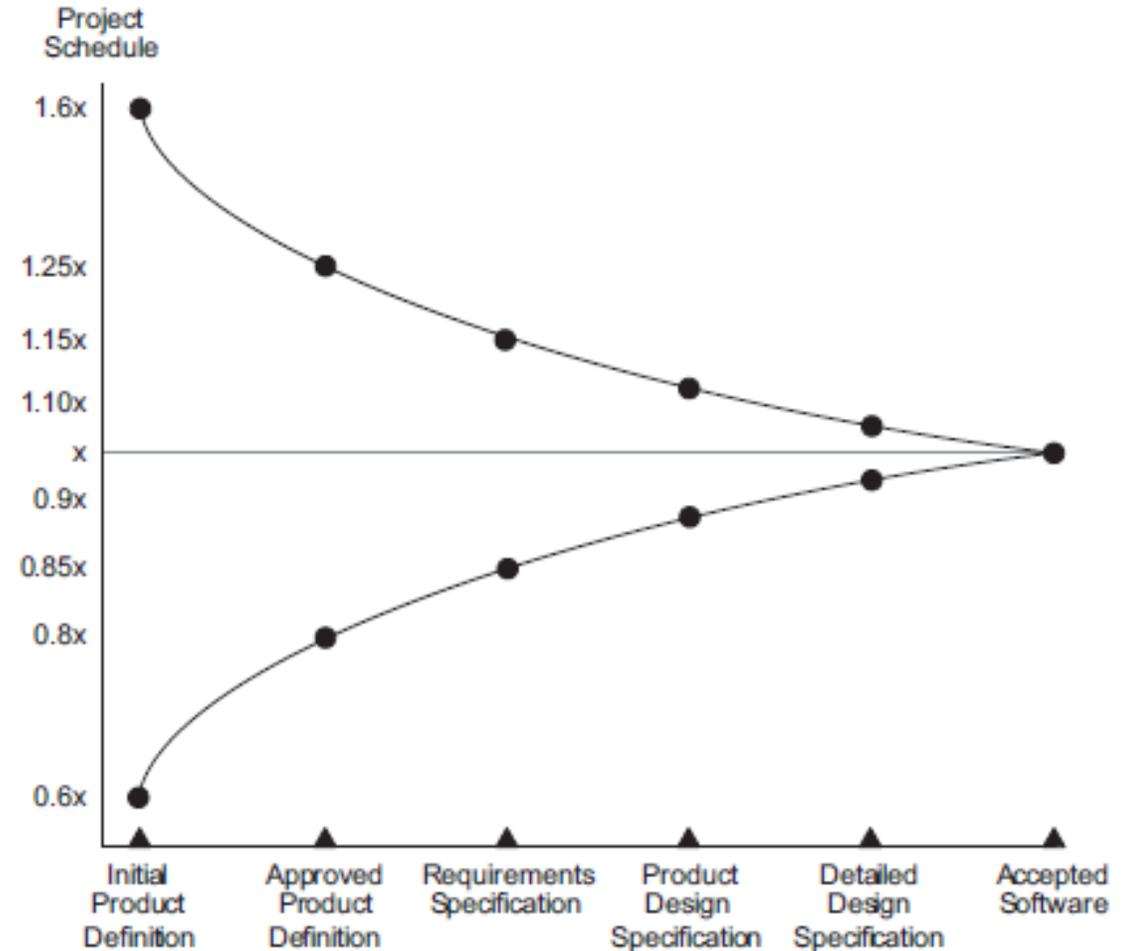
- Tres opciones:
 - Usar valores históricos
 - Ejecutar una iteración
 - Realizar una predicción
- Expresar la estimación en un rango

USAR VALORES HISTÓRICOS

- Genial,
 - si los tengo.
 - Si hay pocos cambios (personal, tecnología) (p. ej. entre una release y otra)
- Preguntarse:
 - ¿La tecnología es la misma?
 - ¿El dominio es el mismo?
 - ¿El equipo es el mismo?
 - ¿El product owner es el mismo?
 - ¿Son las herramientas las mismas?
 - ¿Es el mismo ambiente operativo?
 - ¿Las estimaciones las hicieron las mismas personas?

USAR VALORES HISTÓRICOS

- Si sí:
 - Tomar el promedio y sumar y restar algunos puntos
 - Tomar los valores de la mejor y la peor iteración en los últimos 2 o 3 meses
- Si no,
 - No usar valores históricos
 - Poner un rango más grande: calcular el promedio de la release anterior y multiplicar por el 60 % y 160 % (cono de incertidumbre).



EJECUTAR UNA ITERACIÓN

- Enfoque por defecto
- Ejecutar 1 y observar velocidad. Multiplicarla por 0,6 y 1,6.
- Ejecutar 2 o 3 iteraciones:
 - Usar el rango de velocidades observado. P. ej. 12, 15 y 16 → 12-16
 - Calcular el promedio. Moverse un punto a la derecha en el cono de incertidumbre por cada iteración (no más de 3 o 4 iteraciones):

Iterations Completed	Low Multiplier	High Multiplier
1	0.6	1.60
2	0.8	1.25
3	0.85	1.15
4 or more	0.90	1.10

HACER UNA PREDICCIÓN

- Cuando no tengo datos históricos y no puedo ejecutar iteraciones:
 - Comienzo más adelante (muy lejos en el futuro)
 - Proyecto no puede empezar hasta firmar el contrato (puede que nunca suceda)
- Pasos:
 1. Estimar la cantidad de horas diarias disponibles de cada persona para trabajar en el proyecto (55 % - 80 %).
 2. Determinar la cantidad total de horas que se dedicarán al proyecto en la iteración.
 3. Arbitrariamente y al azar seleccionar historias y expandirlas en tareas (en horas), hasta que tenga suficientes tareas para completar las horas de la iteración. Calcular la velocidad (p. de historia de las historias expandidas).
 4. Convertir la velocidad determinada en el paso 3 en un rango.