



# **Estimación y mediciones de proyectos**

**Sistemas Multimedia**  
Curso 2019-20

A black and white portrait of Galileo Galilei, an elderly man with a full, curly beard and hair, looking slightly to the left. The background is dark and textured.

FLASHBACK

***“Mide lo que se pueda medir,  
y lo que no, hazlo medible”***

Galileo Galilei



## Medir

*“Lo que no se puede definir, no se puede medir  
Lo que no se puede medir, no se puede mejorar  
Lo que no se puede mejorar, finalmente se deteriora”*

Ergo

**Si se puede medir, se puede mejorar**



## Medir

Cada **acción** que realizamos en un tiempo determinado tiene un **coste** y produce unos **beneficios** en concreto.

**Beneficios, tiempo, coste:** la mayoría de las cosas que citamos las podemos **cuantificar**.

Esa cuantificación puede servir para **comparar** con otra acción o simplemente para valorar sobre si la acción en sí misma ha cumplido las **expectativas** con las que las iniciamos.

## FLASHBACK

# El triángulo de hierro

Sólo se pueden fijar dos cosas:

- **Alcance** (funcionalidades)
- **Tiempo**
- **Coste**

La tercera la fija el **desarrollo**





# Estimaciones

Al final, realizar una estimación para un proyecto y mediciones cada cierto periodo de tiempo permite:

- Estimar **coste inicial** (puede servir para realizar un presupuesto)
- Conocer si **se cumplen** las estimaciones **iniciales**
- Realizar propuestas de **mejora** en las estimaciones
- Evaluar la **productividad** de un equipo



## Las horas

La base de la estimación para muchos jefes de proyecto sigue siendo las **horas por persona**, ya que es un aspecto perfectamente **cuantificable** a nivel de costes y de capacidades

Es decir, si mi equipo:

- Está formado por *5 personas*
- Tiene una dedicación diaria de *8 horas*
- Mi capacidad es de **40 horas a la semana**

## Las horas

**Problema:** no todos los miembros del equipo tienen la misma **capacidad** de desarrollo

No nos referimos al número de horas sino de que, en función de su **experiencia** o **conocimiento** sobre una tarea, su capacidad puede ser mayor o menor







## Los puntos o Story Points

Cada tarea debe ser definida en base a unos **puntos** que se consideren necesarios para su finalización

- En equipos **consolidados**, la equivalencia punto-hora es automática
- En equipos **cambiantes** o con miembros con diferentes capacidades puede ser interesante utilizar los puntos como elemento de medida



## Los puntos o Story Points

Fueron introducidos en la **eXtreme Programming**

Cada organización puede decidir la definición de un punto:

- Una jornada completa ideal de un programador senior
- Media jornada

Lo que es importante es que todo el mundo tenga claro **cuál es el valor del story point**

## Los puntos o Story Points

El objetivo del cálculo de la velocidad es determinar la **capacidad** de trabajo del equipo para un Sprint

Para este fin, se basa en la información **recopilada** de los Sprint anteriores





## Capacidad y velocidad de equipo

1. Calcular la **velocidad real** resultante del trabajo realizado por el equipo en el *primer Sprint*
2. **Interpretar** correctamente el significado del valor obtenido
3. **Utilizar** este cálculo de velocidad durante el planning del *segundo Sprint*
4. Conseguir el **compromiso del equipo** con el trabajo planificado para el *segundo Sprint*



## Cálculo de la velocidad

Total de puntos completados:

- Se suman los puntos de historia correspondientes a las historias terminadas

**Suma de puntos finalizados:**  
 $5 + 5 + 4 + 6 + 4 + 8 = 32$  story points



## Cálculo de la velocidad

Total de jornadas reales:

- Se suman las jornadas reales de cada miembro en el Sprint
- Se descuentan las ausencias o circunstancias especiales

**Jornadas previstas:**  $3 \text{ personas} * 15 \text{ días} * 8 \text{ horas} = 360$

**Jornadas reales:** 328 Horas (bajas de 4 días)

Se estima 1 Story Point = 1 Jornada 8 Horas

**$328/8 = 41 \text{ Story Points}$**



## Cálculo de la velocidad

### Cálculo de velocidad

- La velocidad es la división de los puntos completados y de las jornadas reales

**SP Finalizados / SP Utilizados:**  
 $32/41 = 0.78 \rightarrow$  **Velocidad del 78%**



## Cálculo de la velocidad

Una velocidad del 78% significa que:

- El equipo ha podido dedicar el **78%** de su jornada laboral a acciones concretas de desarrollo que se habían **programado** para el Sprint
- Y que un **22%** ha sido invertido en **otras acciones**





## Cálculo de la velocidad

Si un equipo tiene una velocidad del **75%** y la jornada de trabajo es de **8 horas** podemos decir que los miembros del equipo:

- Dedican **6 horas** al día a **trabajo efectivo** (75%)
- Dedican **2 horas** a tareas **no programadas** (25%)



## Cálculo de la velocidad

Y en el próximo Sprint

- 15 días
- 3 Personas
- 1 persona estará de vacaciones 5 días

**¿Cuántos Story Points Podemos Abordar?**



## Cálculo de la velocidad

Y en el próximo Sprint

- 15 días
- 3 Personas
- 1 persona estará de vacaciones 5 días

$$(15 \text{ días} * 2) + (1 * 10 \text{ días}) = 40$$
$$40 * 0.78 = 31.2 \text{ Story Points}$$



## Cálculo de la velocidad

*[Cliente] dice:*

- *¿Cuánto va a costar esto?*
- *¿Y cuánto tiempo tardaremos en tenerlo?*
- *La verdad es que tenemos unas fechas límite muy definidas, y el presupuesto no nos da para muchos cohetes.*



## Cálculo de la velocidad

*[Vendedor/Técnico ágil] Responde:*

- *Pues es que nosotros no estimamos en horas, sino en Story Points.*
- *Pero no se preocupe, porque esto es un marco de trabajo Scrum altamente reconocido.*



## Cálculo de la velocidad

*[Cliente] Señala:*

- *¿Ah, sí? Pues que sepas que mientras tanto, te pagaré en BarPoints (tickets del bar).*
- *Pero no te preocupes, porque esto en nuestra cafetería de empresa es una forma de pago altamente reconocida.*

# Cálculo de la velocidad

Story points es interno, no externo

Los clientes siguen necesitando saber en muchas ocasiones

- **Cuándo** van a estar disponibles sus aplicaciones
- A qué **precio**





## Estimando el backlog

Se deben estimar el coste todos los elementos del Product Backlog, en el que puedes tener

- Tareas a **alto nivel** de detalle.
- Historias de usuario.
- Tareas de **bajo nivel** (en sprint backlog)





## Estimando el backlog

Si no se dispone de un **nivel de detalle adecuado** puede producir **estimaciones poco certeras**

Para todos los elementos que se hallen en el Sprint Backlog, la estimación no es sólo inicial sino que debe **actualizar** en cada Sprint con una estimación del trabajo que queda pendiente para su finalización



## La necesidad de las estimaciones

Si se quiere hacer un correcto **seguimiento** del estado del proyecto

- Hay que cuantificar el **coste previsto** (en horas o puntos) de cada una de las historias de usuario.

Dada la filosofía de las metodologías ágiles, es importante que esta estimación se realice **en equipo**, ya que imponer una estimación por parte de un jefe de proyecto hace que el equipo puede asumir pero no respetar adecuadamente esa estimación.



## La necesidad de las estimaciones

También es exigible cierto **sentido común** por parte del equipo:

- Todo proyecto tiene unos **plazos** que cumplir

El **excesivo optimismo** tampoco es una buena opción

- Hay un cementerio de proyectos inacabados llenos de estimaciones en exceso optimistas

¿Cómo estimar?



## Método Wideband Delphi

En 1981, Barry Boehm, en su libro Software Engineering Economics, definió una variante del método de estimación Delphi utilizado en los años 70, que a su vez derivaba de modelos utilizados en la Guerra Fría



## Método Wideband Delphi

1. El **Scrum Master**, o un responsable de las estimaciones, **presenta** la historia de usuario o tarea a estimar
2. Se abre un **debate** entre los miembros del equipo de desarrollo, donde todos pueden realizar comentarios acerca de la dificultad de la tarea



## Método Wideband Delphi

3. Cada miembro del equipo cuantifica su **estimación** para la tarea y la escribe
4. El responsable **recoge las estimaciones** de todas las tareas de cada uno de los miembros del equipo de desarrollo
5. El responsable prepara un **resumen de las estimaciones** y la reparte a todos los miembros del equipo de desarrollo.

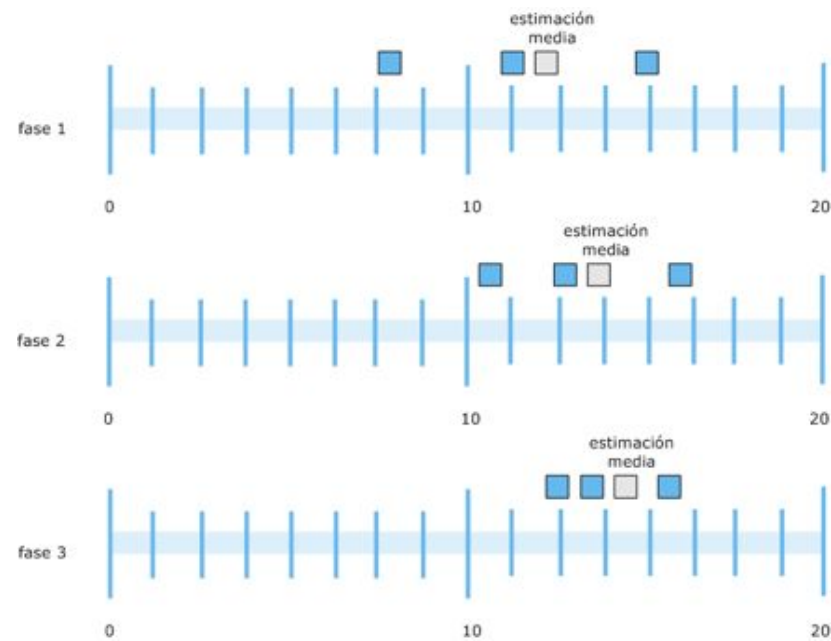


## Método Wideband Delphi

6. El equipo de desarrollo **votan anónimamente** si aceptan la estimación media (si alguien vota que no, se vuelve al paso 3)
7. La estimación final es una **estimación única** (single-point estimate). También podría ser un rango creado durante la estimación en la que la single-point es el caso esperado
8. Se repiten los pasos del 3 al 7 hasta que se llegue a un consenso o resultado homogéneo razonable



# Método Wideband Delphi





## Método Planning Poker

Una variante muy curiosa del método Wideband Delphi, muy aceptada dentro de la comunidad ágil, fue el Planning Poker definido por *James Greenig*

Se basa en el uso de una **baraja de cartas** cuyas cartas tienen los números de la serie de *Fibonacci* (1,2,3,5,8,13,..)



## Método Planning Poker

1. El Scrum Master, o un responsable de las estimaciones, **presenta** una historia de usuario o tarea a estimar.
2. Se abre un debate entre los miembros del equipo de desarrollo, donde todos pueden realizar comentarios acerca de la dificultad de la tarea.
3. Cada miembro del equipo elige la carta que contiene el valor más **cercano a su estimación** de la historia y la pone boca abajo



## Método Planning Poker

4. Se dan las **vuelatas** a las cartas.
5. El responsable puede dar por correcta las estimaciones **homogéneas**, y repite la estimación en aquellos casos donde ha habido **discrepancias** notables.
6. Se repiten los pasos del 3 al 5 hasta que se llegue a un **consenso** o resultado homogéneo razonable.

# Método Planning Poker



# Método Planning Poker



# Método Planning Poker



# Método Planning Poker





# Método Planning Poker



# Método Planning Poker





# Método Planning Poker

Dos problemas **comunes**

- Incertidumbre del **producto**
- Incertidumbre **técnica**

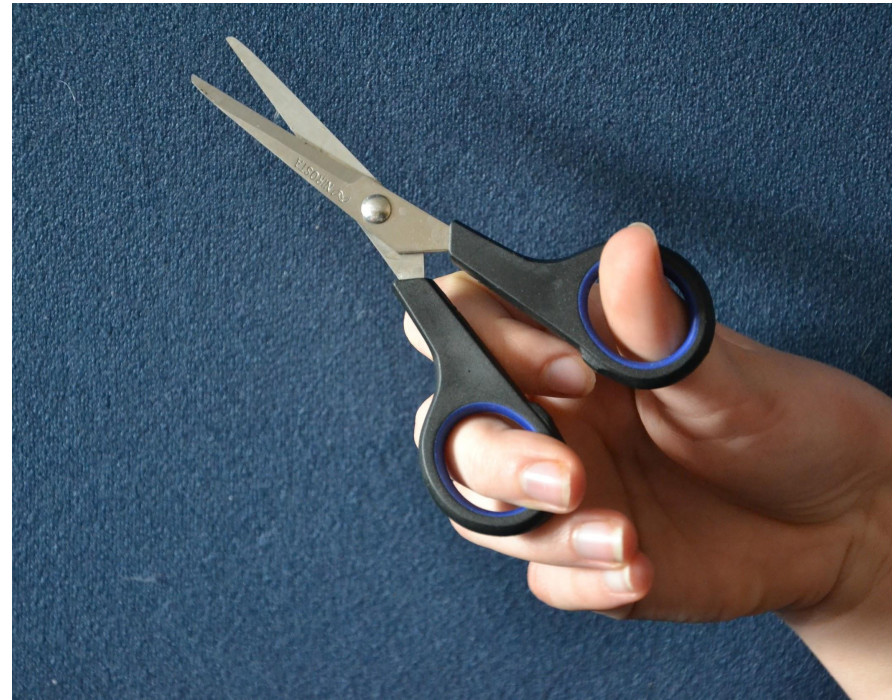
Dos soluciones

- Apartar la historia hasta que se resuelva la incerteza
- Utilizar una estimación en forma de rango

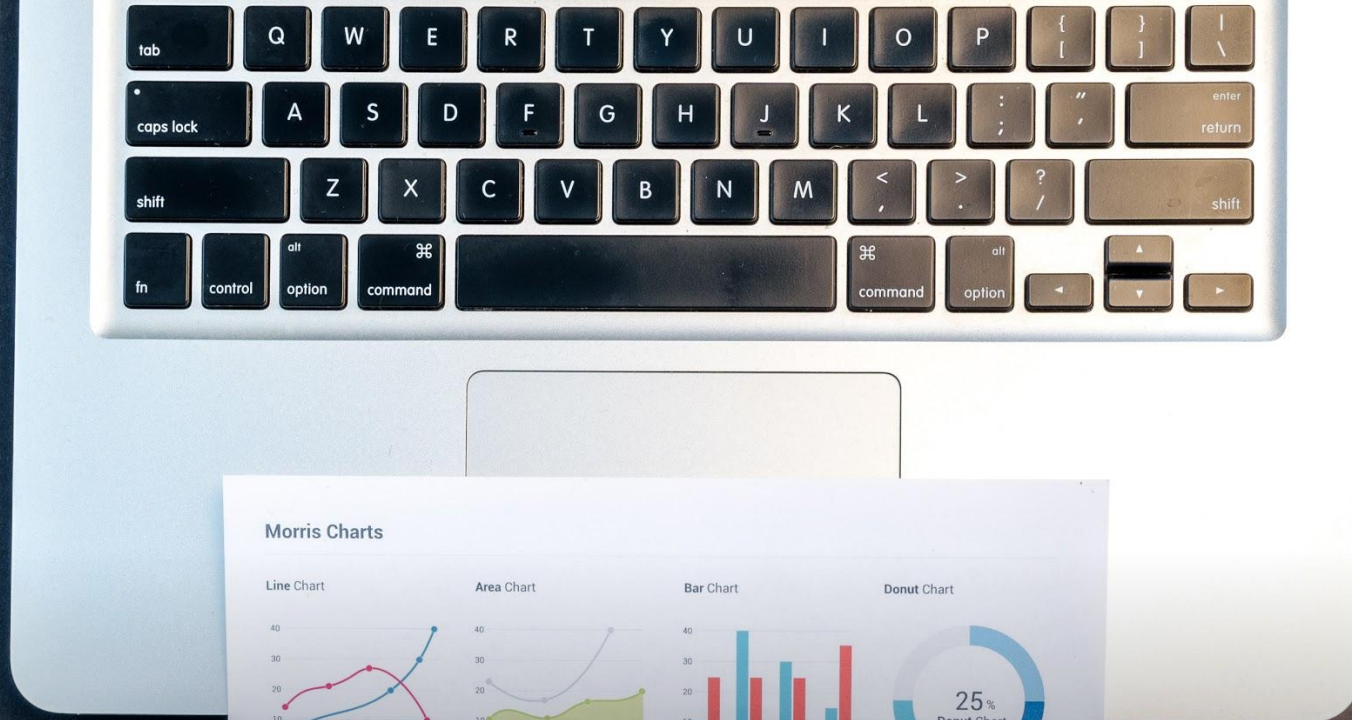
# Método Planning Poker

Ante el total desacuerdo

- Dividir la tarea en subtarear más sencillas





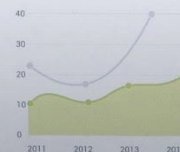


### Morris Charts

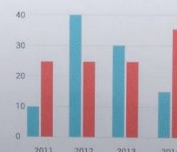
Line Chart



Area Chart



Bar Chart

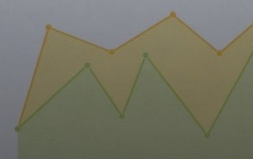


Donut Chart



### Sparkline Charts

Line Chart



Bar Chart



Pie Chart



### Easy Pie Charts



50%

75%

100%

# Gráficos de seguimiento

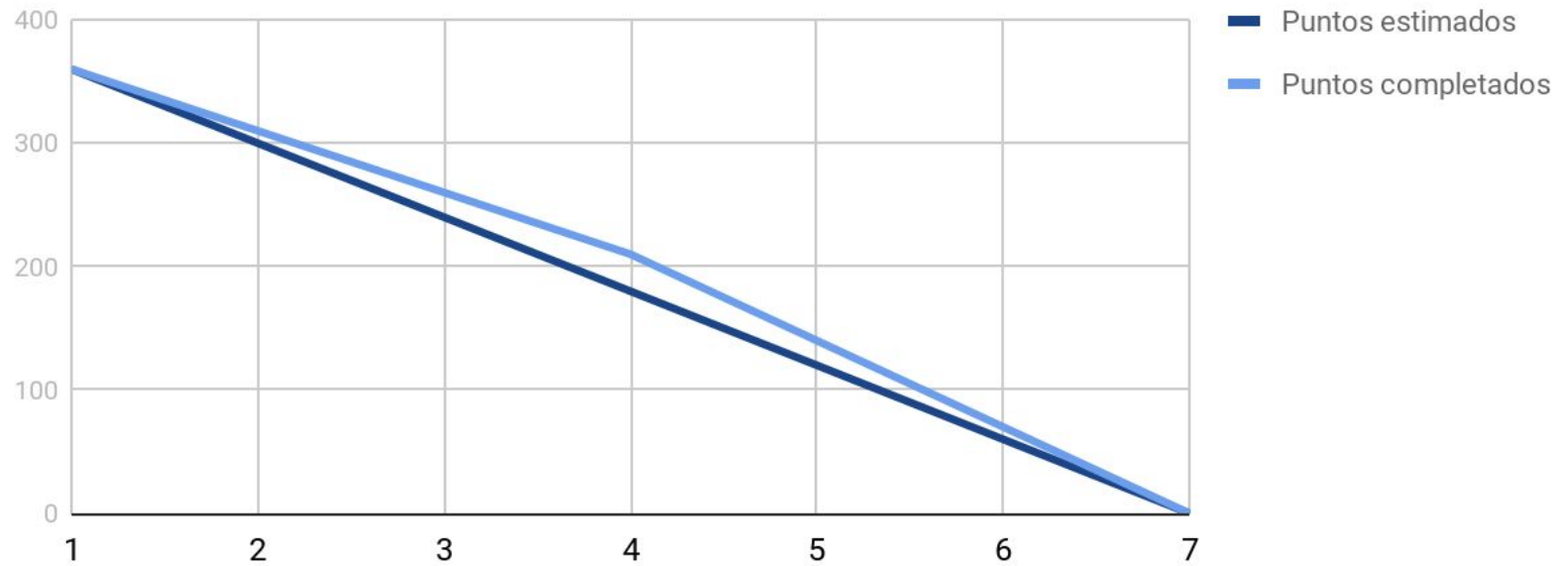


# Burn down chart

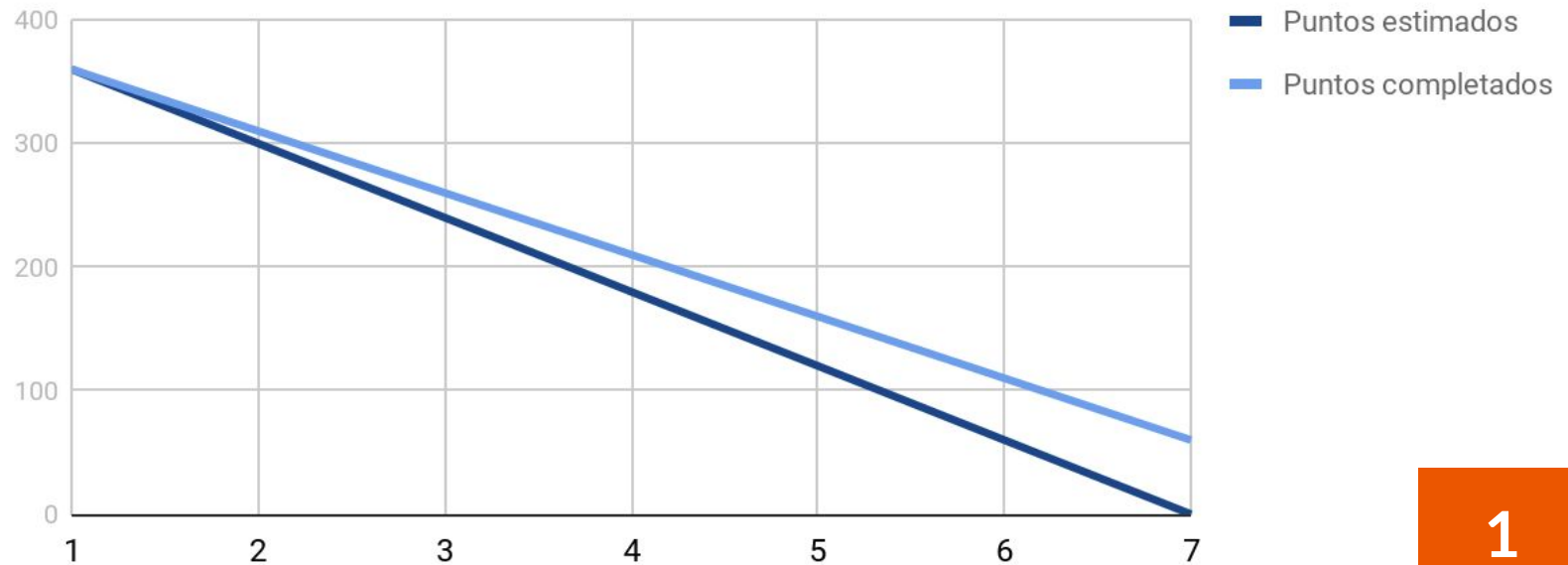
## Burn down chart o gráfico de progreso

- Este gráfico tiene como objetivo comprobar el **seguimiento** del progreso de un proyecto Scrum
- En el eje horizontal se muestran los **sprints previstos** para el proyecto y en el eje vertical los **puntos de historia pendientes**
- Se suelen dibujar dos líneas, las **estimadas al inicio** del proyecto y las que se van **cumpliendo**.
- Los puntos de historia se rellenan al **principio** de cada Sprint

# Burn down chart

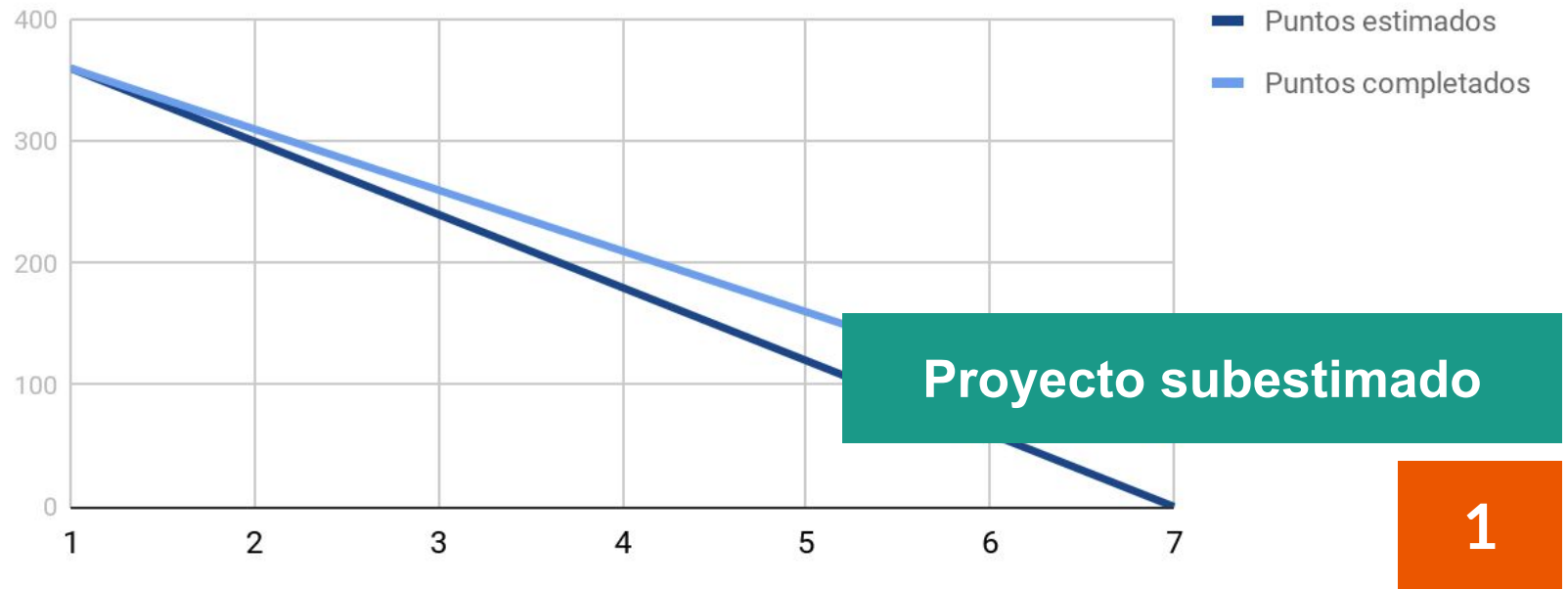


# Burn down chart

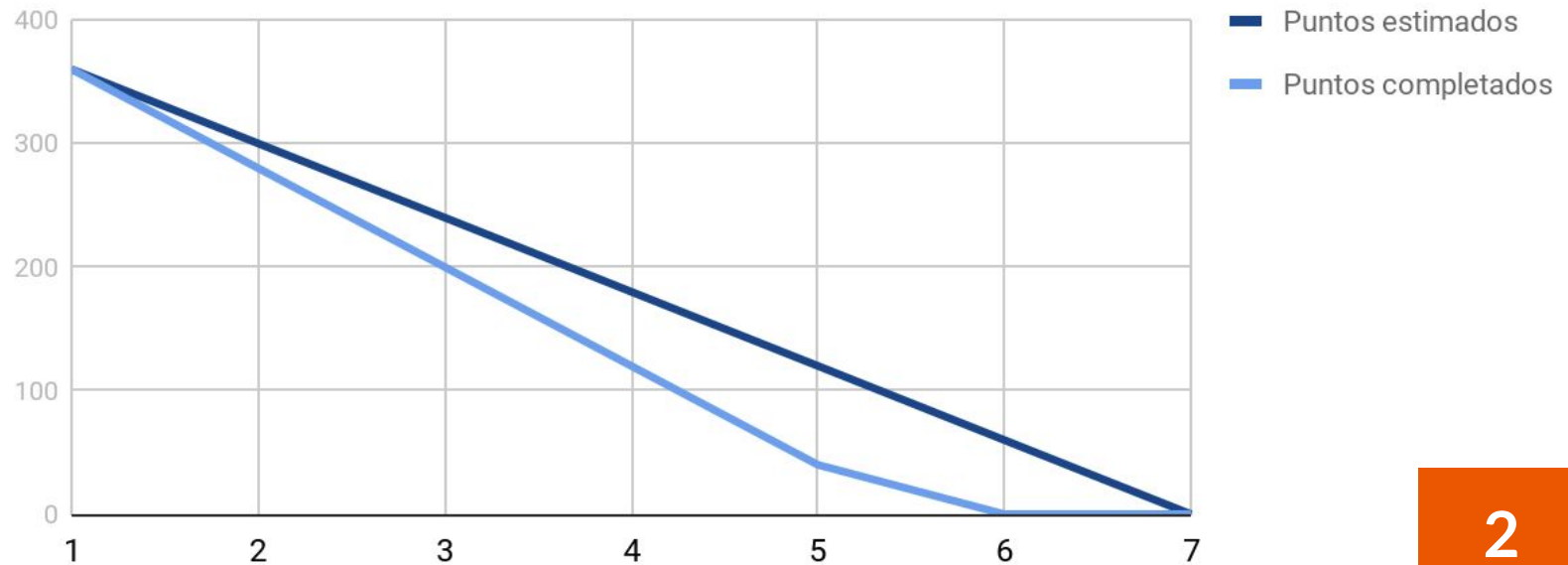




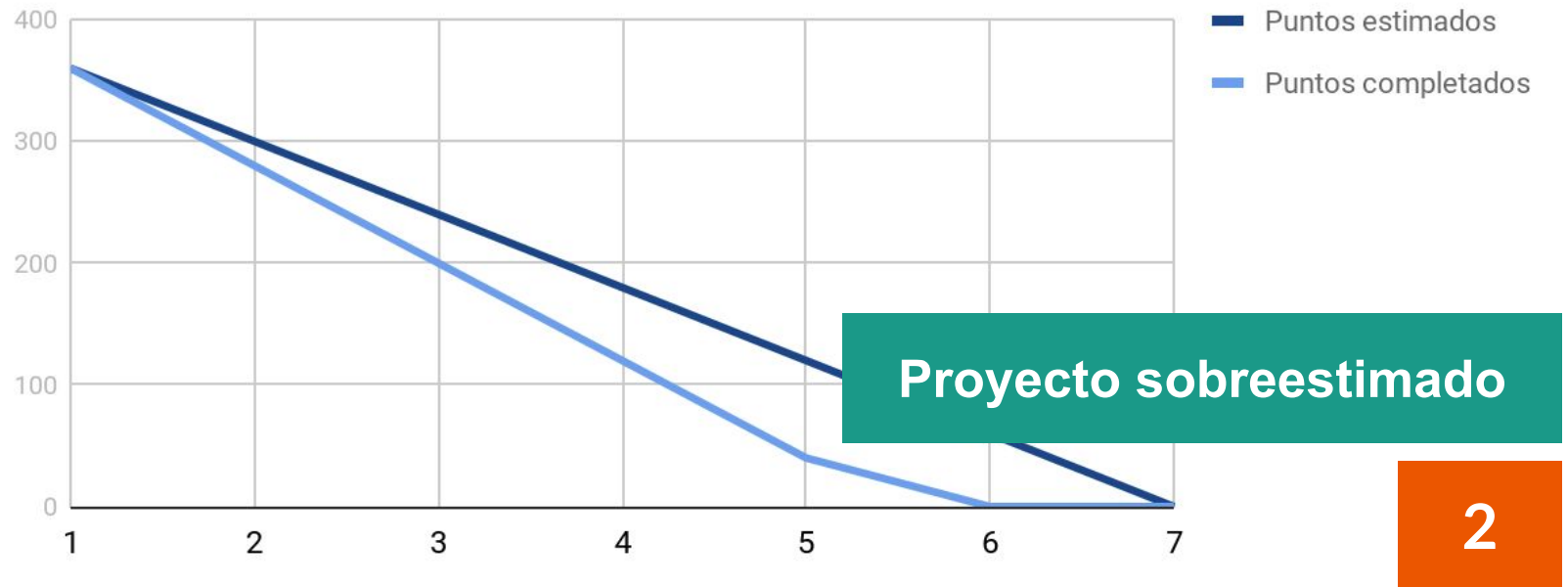
# Burn down chart



# Burn down chart



# Burn down chart



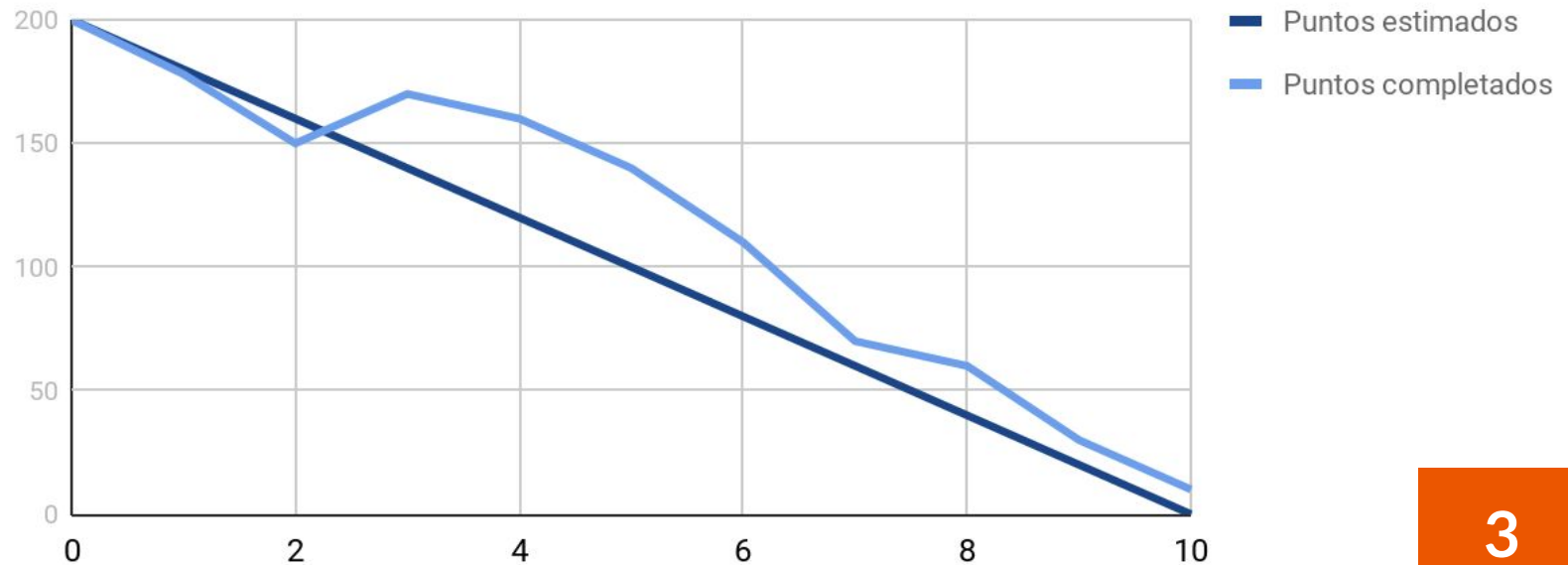


## Burn down chart

Este gráfico también puede ser calculado para **cada día** de proyecto en vez de para cada Sprint.

Si se utilizan herramientas de cálculo se puede obtener de forma automática información más actualizada aunque disponer de ella a nivel de sprint suele ser suficiente

# Burn down chart





## Burn down chart

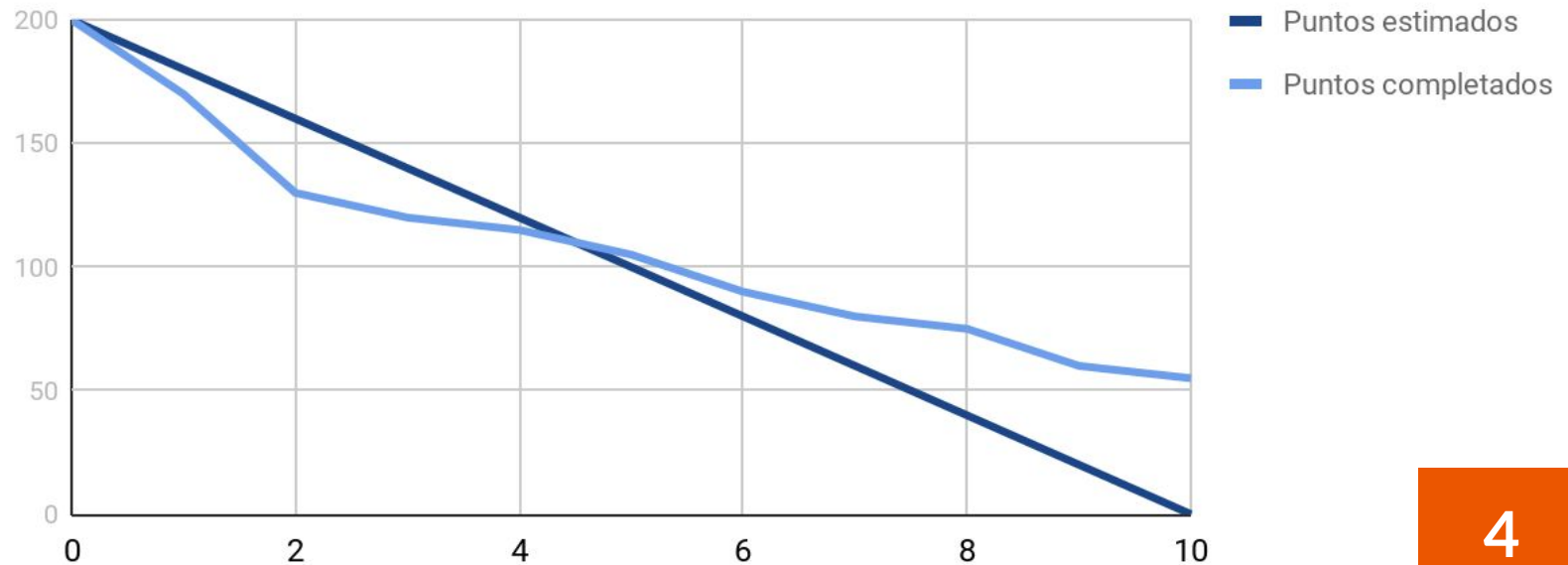
Los Sprint 1 y 2 iban bien.

En los Sprint 3 y 4 o se han reestimado o se han añadido nuevas tareas y se ha producido desfases

En los Sprint 5 y 6 se recuperan tiempo perdido

Al final no hay un retraso importante

# Burn down chart





## Burn down chart

Los Sprint 1 y 2 iban bien.

En los Sprint 3 y 4 se ajustó la velocidad del equipo.

Es posible que en las últimas tareas se cometieran errores de estimación o en las que más incertidumbre había.

No se acabará a tiempo





## Burn down chart

Duda razonable

- Si al principio estimo 2500 horas de trabajo pendiente
- Si acabo el primer sprint que dura 250 horas,
- ¿Cuánto trabajo queda pendiente?



## Burn down chart

Duda razonable

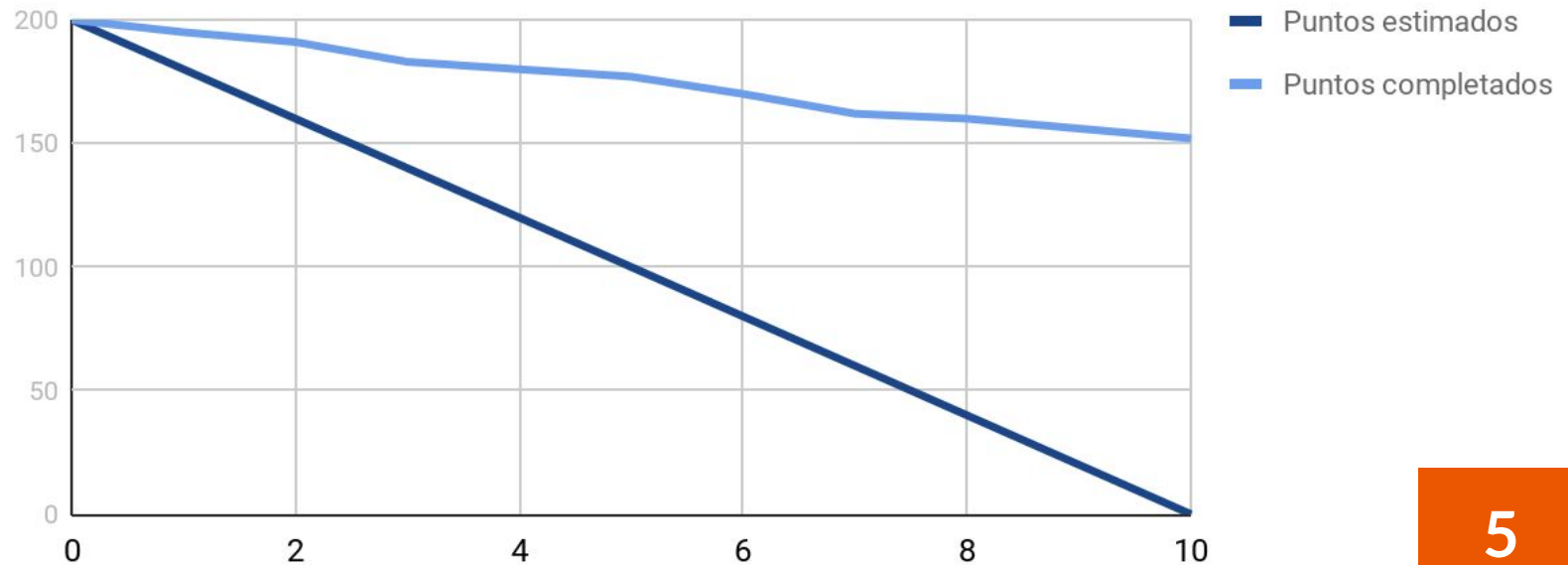
- Si al principio estimo 2500 horas de trabajo pendiente
- Si acabo el primer sprint que dura 250 horas,
- ¿Cuánto trabajo queda pendiente?

Quedan las horas que estime para las tareas que faltan

A full-body image of Jason Voorhees from the Friday the 13th movie series. He is standing in a dark, misty forest at night. He wears his signature yellow hockey mask with black eye holes and a red mouth. He is dressed in a dark, tattered jacket over a black shirt. In his right hand, he holds a large, curved machete. The background shows dark trees and a foggy atmosphere, creating a scary and ominous mood.

**La gráfica más temible**

# Burn down chart





## Burn down chart

Es casi horizontal y calcular cuando puede tocar suelo puede provocar pánico.

Estimaciones incumplidas

Errores en el Done

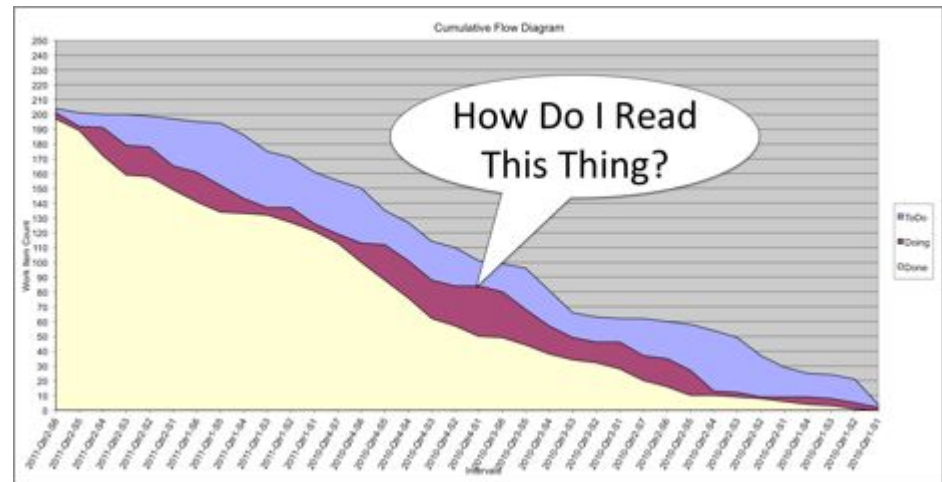
Falta de Acuerdo con el cliente

# Cumulative flow diagram

Se cuentan las semanas que tiene para el proyecto

Se diferencian:

- Hecho
- Haciendo
- Por hacer



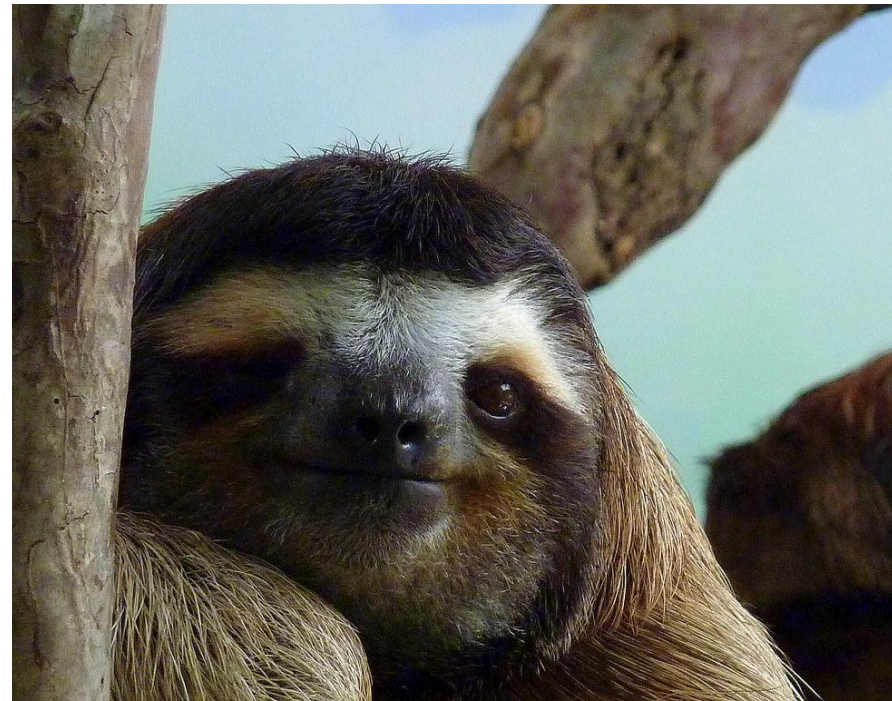


## Cumulative flow diagram

- Inicialmente **todas** las funcionalidades están **por hacer**
- Al principio la velocidad suele ser **baja**
- Posteriormente se **incrementa** la velocidad
- Al final la velocidad también suele **reducirse**

## Cumulative flow diagram

¿Porqué suele reducirse la velocidad al final?





# Cumulative flow diagram

¿Porqué suele reducirse la velocidad al final?

- Porque al final muchas tareas que se daban por terminadas no lo estaban del todo





# Estimación y mediciones de proyectos

Dudas, cuestiones...