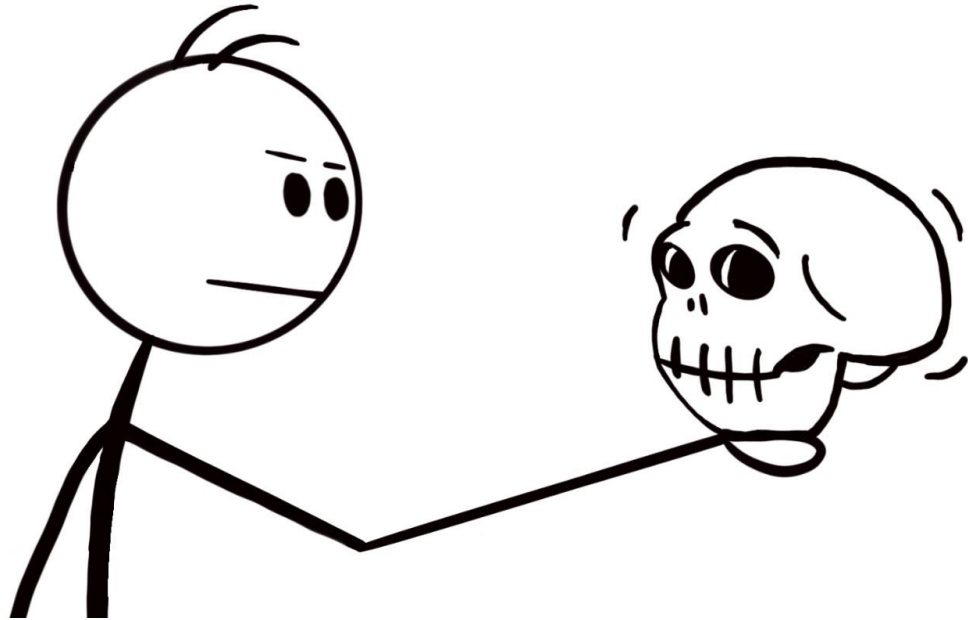
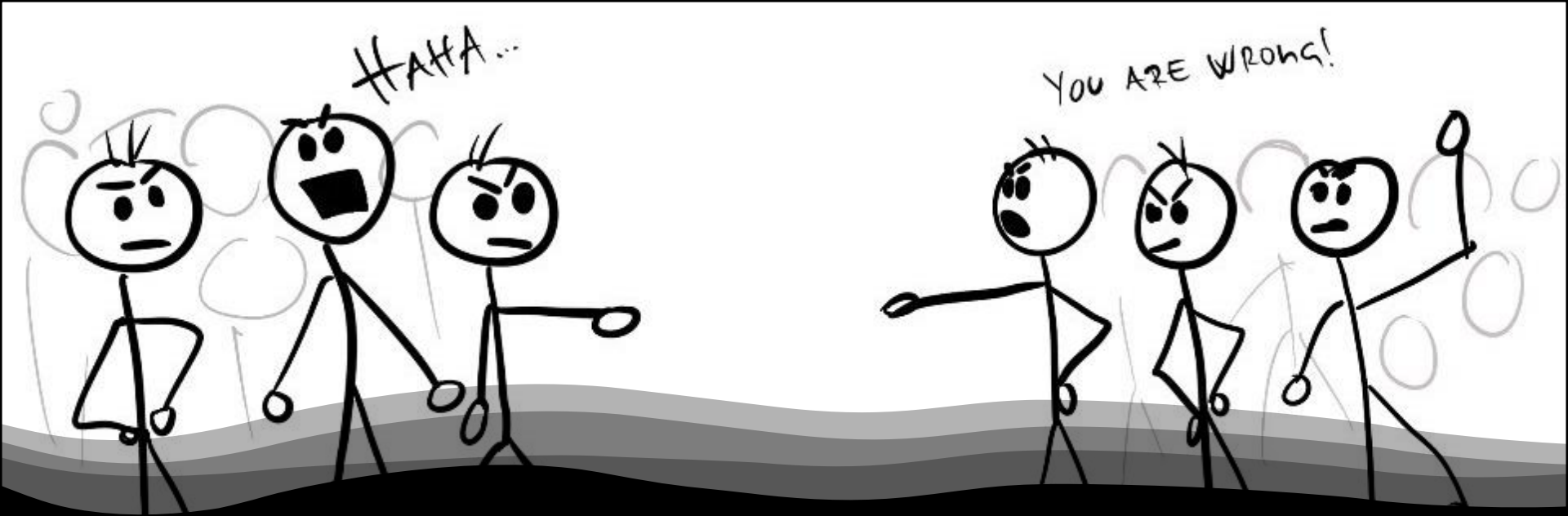


Estimate or #noEstimate?





No ser dogmático sobre nada.  
Ser pragmático es la clave del éxito en casi todo.

# Tips respecto de las estimaciones...



Si las estimaciones se utilizan como compromisos son muy peligrosas y perjudiciales para cualquier organización.



Lo más beneficioso en las estimaciones es el “proceso de hacerlas”.



La estimación podría servir como una gran respuesta temprana sobre si el trabajo planificado es factible o no.



La estimación puede servir como una gran protección para el equipo.

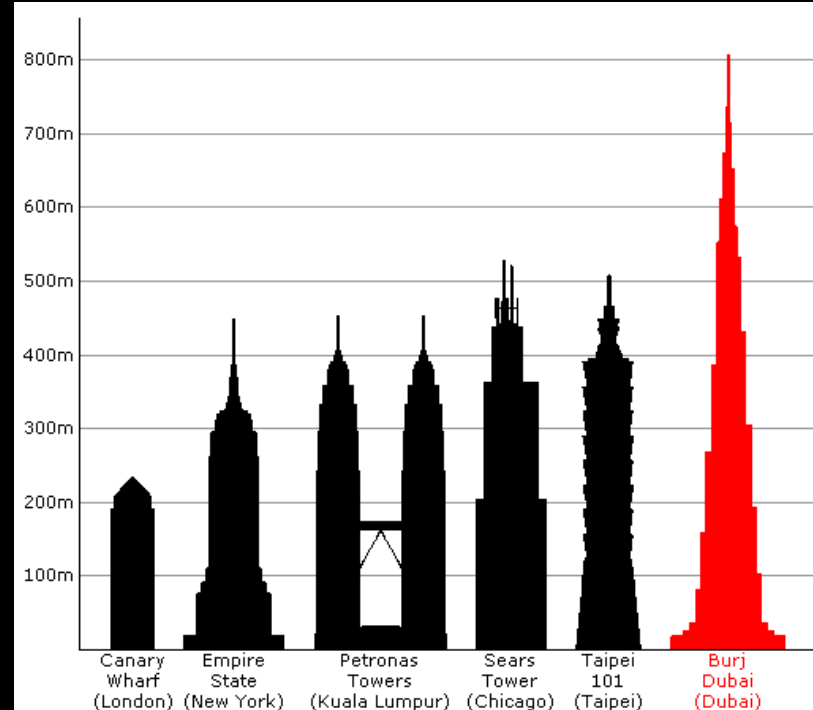
## Estimaciones Ágiles

- Las features/stories son estimadas usando una medida de tamaño relativo conocida como story points (SP)
- Las medidas relativas no son absolutas.
- Story Points no es una medida basada en tiempo



# Estimación Relativa

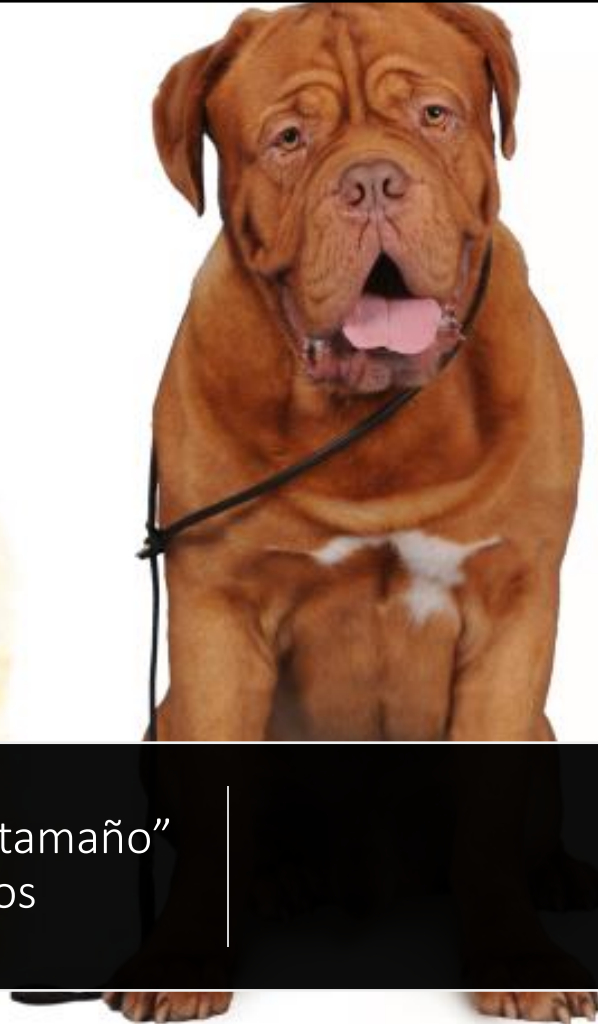
- Las personas no saben estimar en términos absolutos
- Somos buenos comparando cosas
- Comparar es generalmente más rápido.
- Se obtiene una mejor dinámica grupal y pensamiento de equipo más que individual
- Se emplea mejor el tiempo de análisis de las storys



# y...pero, el tamaño?

- “La palabra tamaño refiere a cuan grande o pequeño es algo”
- El tamaño es una medida de la cantidad de trabajo necesaria para producir una feature/story.
- El tamaño indica:
  - Cuán compleja es una feature/story
  - Cuánto trabajo es requerido para hacer o completar una feature/story
  - Y cuán grande es una feature/story





Por favor, “dé tamaño”  
de estos perros

# Descripción del Trabajo vs. Esfuerzo



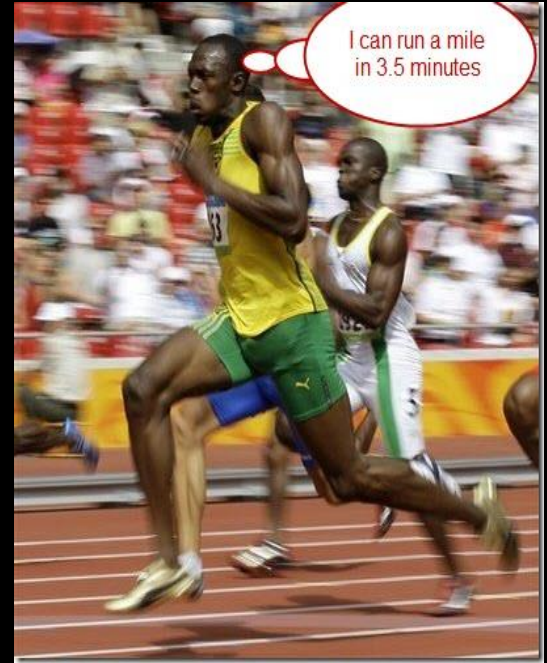


# Tamaño Vs. Esfuerzo



Las estimaciones basadas en tiempo son más propensas a errores debido a varias razones.

- Habilidades
- Conocimiento
- Asunciones
- Experiencia
- Familiaridad con los dominios de aplicación/negocio



## Tamaño NO ES esfuerzo

# Y qué hacemos con el tamaño!?????

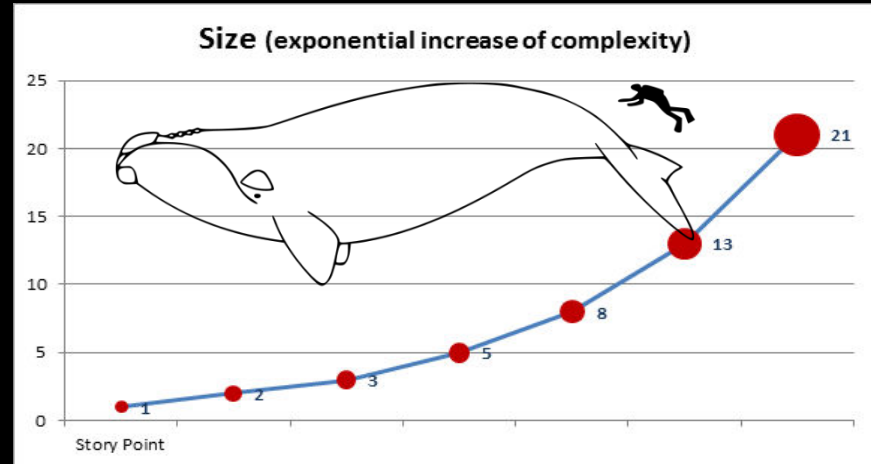
- Tamaño por números: 1 a 10
- Talles de remeras: S, M, L, XL, XXL
- Serie  $2^n$  : 1, 2, 4, 8, 16, 32, 64, etc.
- Fibonacci: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, etc.

**Una vez elegida la escala no se cambia! Si se cambia cambiamos el metro patrón**



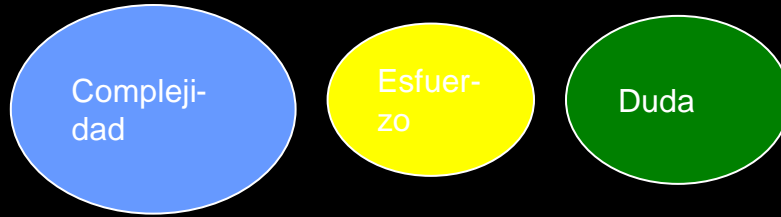
# Y qué es un Story Point? (una de muchas definiciones)

- Es una unidad de medida específica (del equipo) de, complejidad, riesgo y esfuerzo, es lo que “el kilo” a la unidad de nuestro sistema de medición de peso
- Story point da idea del “peso” de cada story y decide cuan grande (compleja) es
- La complejidad de una
  - feature/story tiende a
  - incrementarse exponencialmente.



# 3 Stories que queremos estimar

Story 1



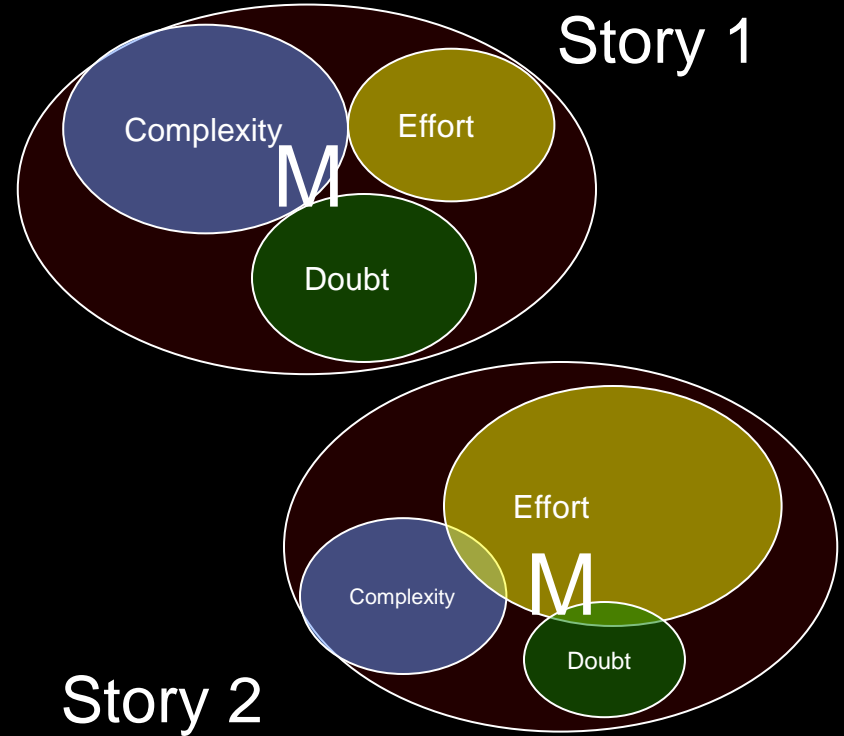
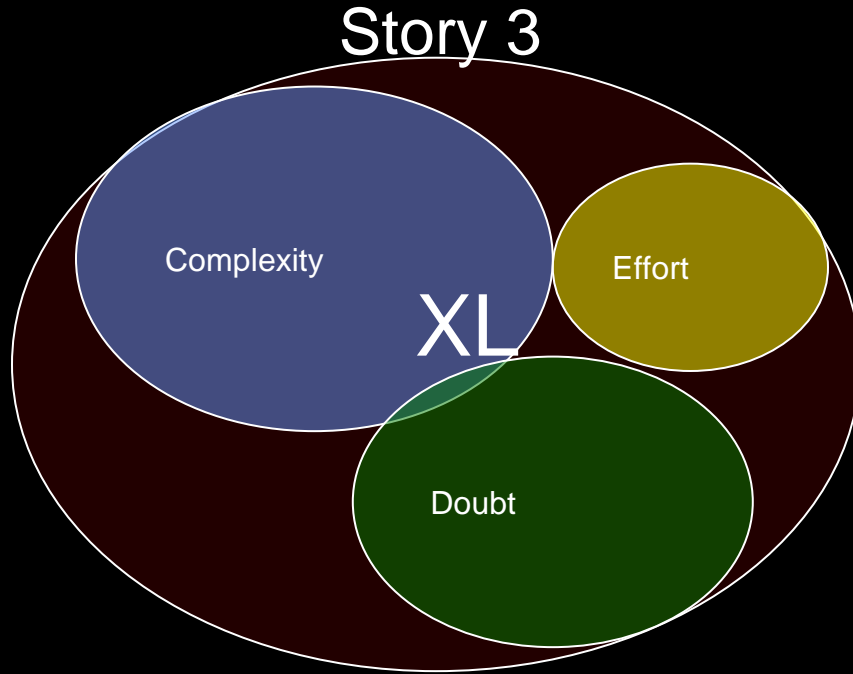
Story 2



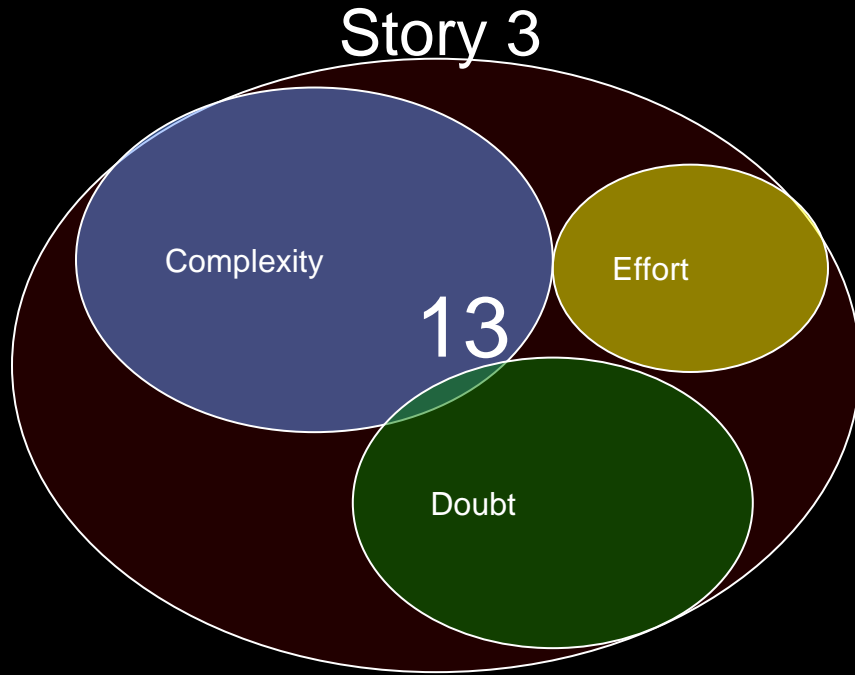
Story 3



# “tamaño” de las Stories

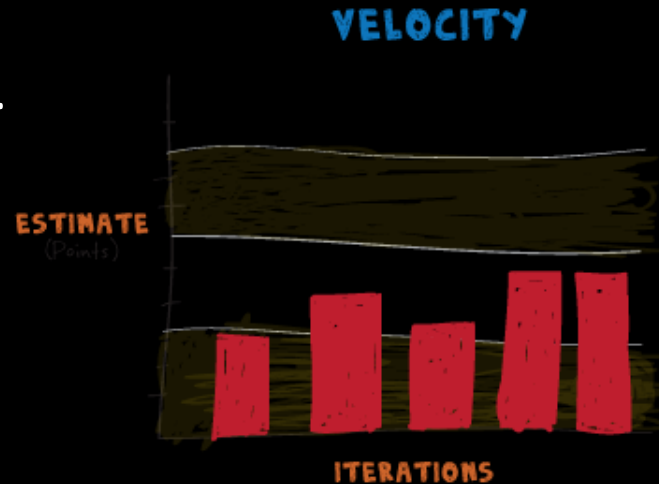


# Y si usamos números?



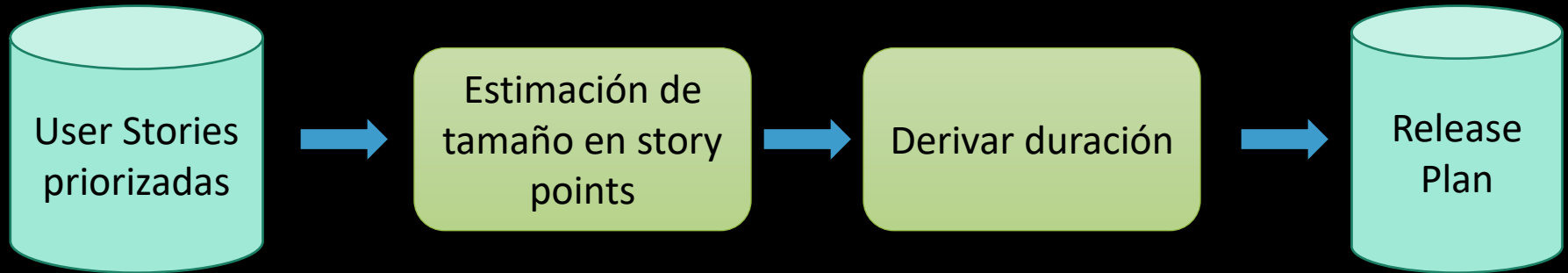
# Velocidad (Velocity)

- **Velocidad/ Velocity** es una medida (métrica) del progreso de un equipo. Se calcula sumando el número de story points (asignados a cada user story) que el equipo **completa** durante la **iteración**.
- Se cuentan los **story points** de las **Users Storys** que están **completas**, no parcialmente completas.
- La Velocidad corrige los errores de estimación.



# ¿Y cómo hago con un proyecto?

- Si estimo User Storys cómo hago para estimar un proyecto?
  - La duración de un proyecto no se “estima”, se deriva.... tomando el número total de story points de sus user storys y dividiéndolo por la velocidad del equipo.



- La velocidad nos ayuda a determinar un horizonte de planificación apropiando
- La estimación en story points separa completamente la estimación de esfuerzo de la estimación de la duración.



# Una Propuesta de método de estimación






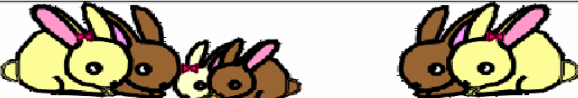


# Poker estimation

- Popular entre los Agile practitioners, publicado por Mike Cohn
- Combina opinión de experto, analogía y desegregación.
- Participantes en “planning poker” son desarrolladores
  - “Las personas más competentes en resolver una tarea deben ser quienes las estiman”



# Fibonacci (se acuerdan?)

- La secuencia empieza en 1 y cada numero subsecuente es la suma de los dos precedentes. (1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144....)

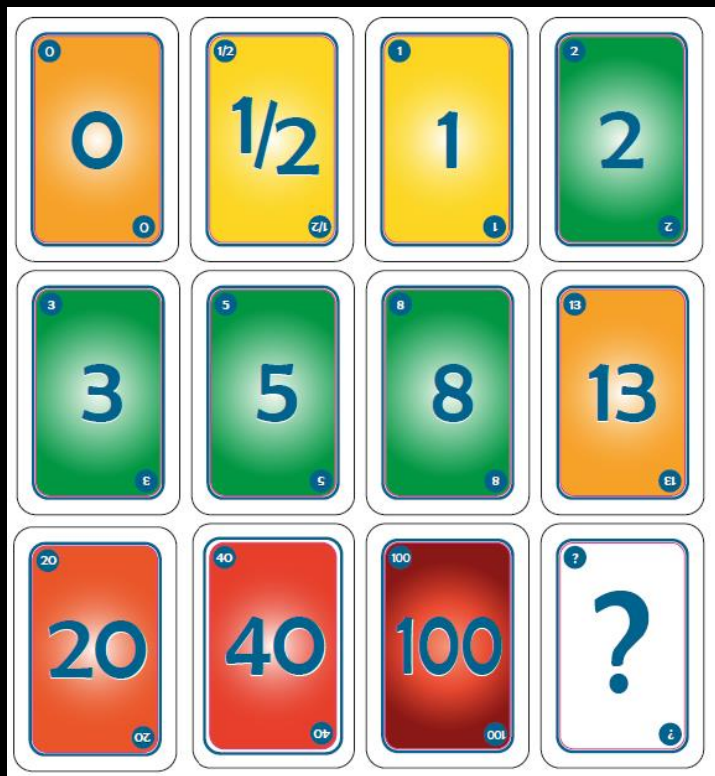
		Pares de conejos 1
1 mes		1
2° mes		2
3° mes		3
4° mes		5
5° mes		8

# Poker Planning



<http://www.planningpoker.com/detail.html>

# ¿Cómo “decodificar” las estimaciones?



- 0: Quizás ud. no tenga idea de su producto o funcionalidad en este punto.
- 1/2, 1: funcionalidad pequeña (usualmente cosmética).
- 2-3: funcionalidad pequeña a mediana. Es lo que queremos. 😊
- 5: Funcionalidad media. Es lo que queremos 😊
- 8: Funcionalidad grande, de todas formas lo podemos hacer, pero hay que preguntarse sino se puede partir o dividir en algo más pequeño. No es lo mejor, pero todavía 😊
- 13: Alguien puede explicar por que no lo podemos dividir?
- 20:Cuál es la razón de negocio que justifica semejante story y más fuerte aún, por qué no se puede dividir?.
- 40: no hay forma de hacer esto en un sprint.
- 100: confirmación de que está algo muy mal. Mejor ni arrancar.

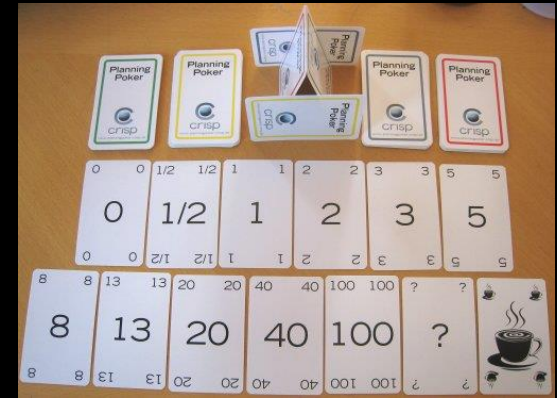
# Poker Planning

## Prerrequisitos:

- Lista de features/stories a ser estimadas
- Cada estimador tiene un mazo de cartas.

## Pasos:

1. Determine la **base story** ( la canónica) que será usada para comprar con las otras stories. Digamos, Story Z.
  - 1.1 La story a ser estimada se lee a todo el equipo.
  - 1.2 Los estimadores discuten la story, haciendo preguntas al product owner (las que se necesiten).
  - 1.3 Cada estimador selecciona una carta y pone la carta boca abajo en la mesa.
  - 1.4 Cuando todos pusieron las cartas, las mismas se exponen al mismo tiempo.
  - 1.5 Si todos los estimadores selecciona el mismo valor, ese es el estimado. Sino, los estimadores discuten sus resultados, poniendo especial atención en los más altos y los más bajos. Después de la charla, GOTO to 1.3.
2. Se toma la próxima story, se discute con el product owner.
3. Cada estimador asigna a la story un valor por comparación contra la base story. “Cuan grande/pequeña, compleja, riesgosa es esta story comparada con Story Z?”. GOTO to 1.3



Agile es empírico,  
Inspeccionar y adaptar  
es mandatorio!





# Bibliografía

- Software Estimation: Demystifying the Black Art by Steve McConnell  
Microsoft Press 2006 ISBN:0735605351
- Agile Estimating and Planning, Mike Cohn,  
Pearson Education, ISBN: 9780137126347
- <http://www.planningpoker.com/detail.html>

