Estimaciones

Variables de estimación

- Tiempo/Duración → Puede ser que no se estime ya que sale de estimar los recursos y el costo (hs/pers) que se necesita
- Costo/Esfuerzo → Parecido a lo que pasa con el tiempo
- Recursos
- **Tamaño** → Primera pregunta que tenemos que hacernos
    - Cual es el tamaño de lo que tenemos que construir?
    - Es la única variable que no cambia
    - A partir de esta derivan el resto, las demas no se estiman

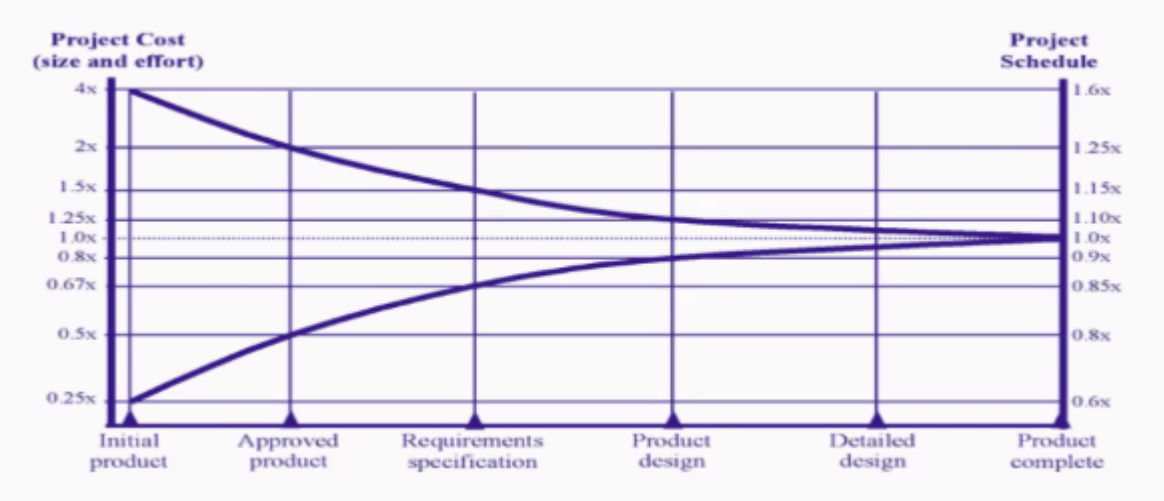
> > Requerimientos → Tamaño (+ Complejidad)→ Esfuerzo → Costo y Duración

Por qué fallan las estimaciones?

- Optimismo
- Estimaciones informales
- No hay historia
- Mala definición del alcance
- Falta de experiencia
- Complejidad del problema
- No se estima el tamaño
- Porqeu la estimacion fue buena pero cuando arranca el proyecto
    - Mala admin de los requerimientos
    - No hay seguimiento y control
    - Se confunde progreso con esfuerzo

Nivel de incertidumbre

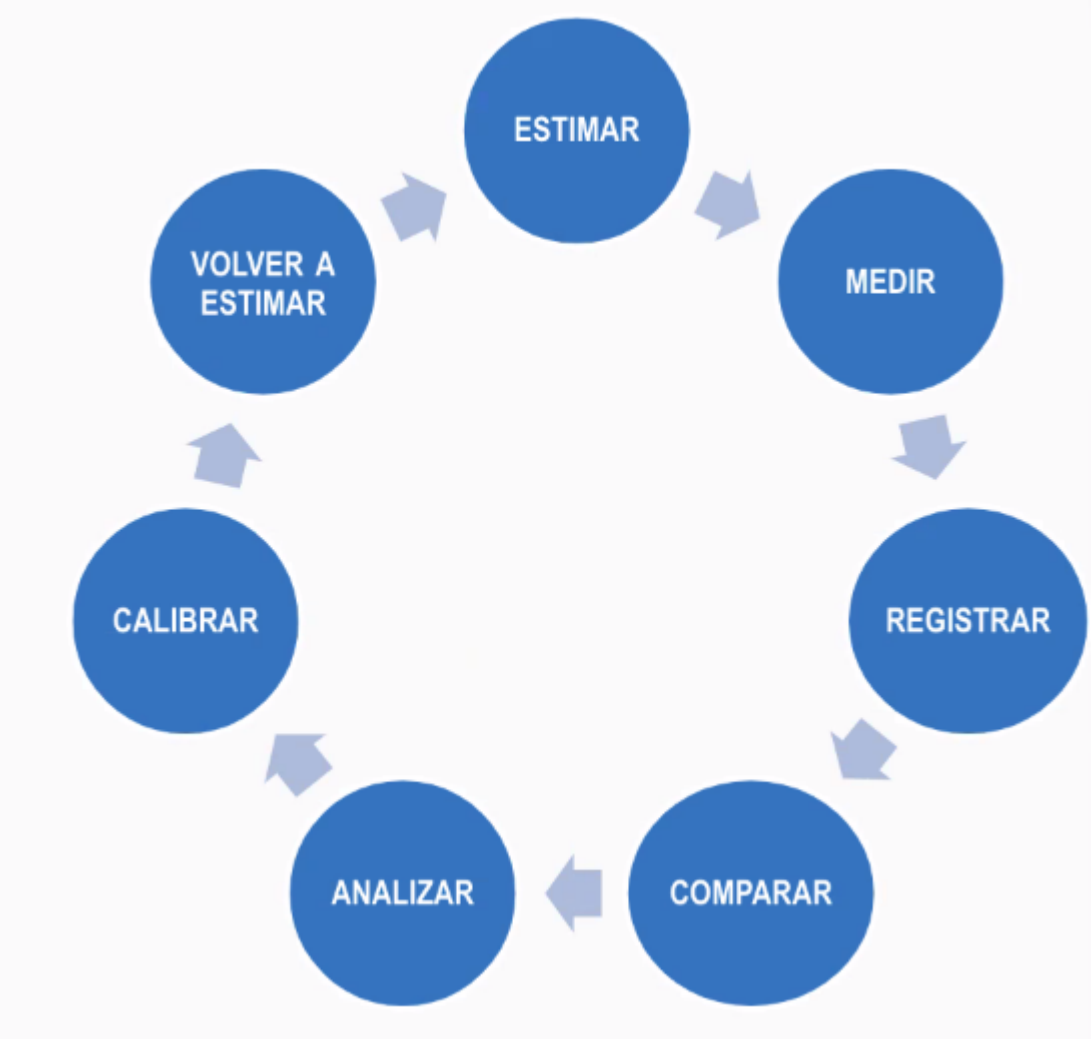
El dia del fin del proyecto es cuando realmente sabemos cuanto lleva el desarrollo del producto.



Tips

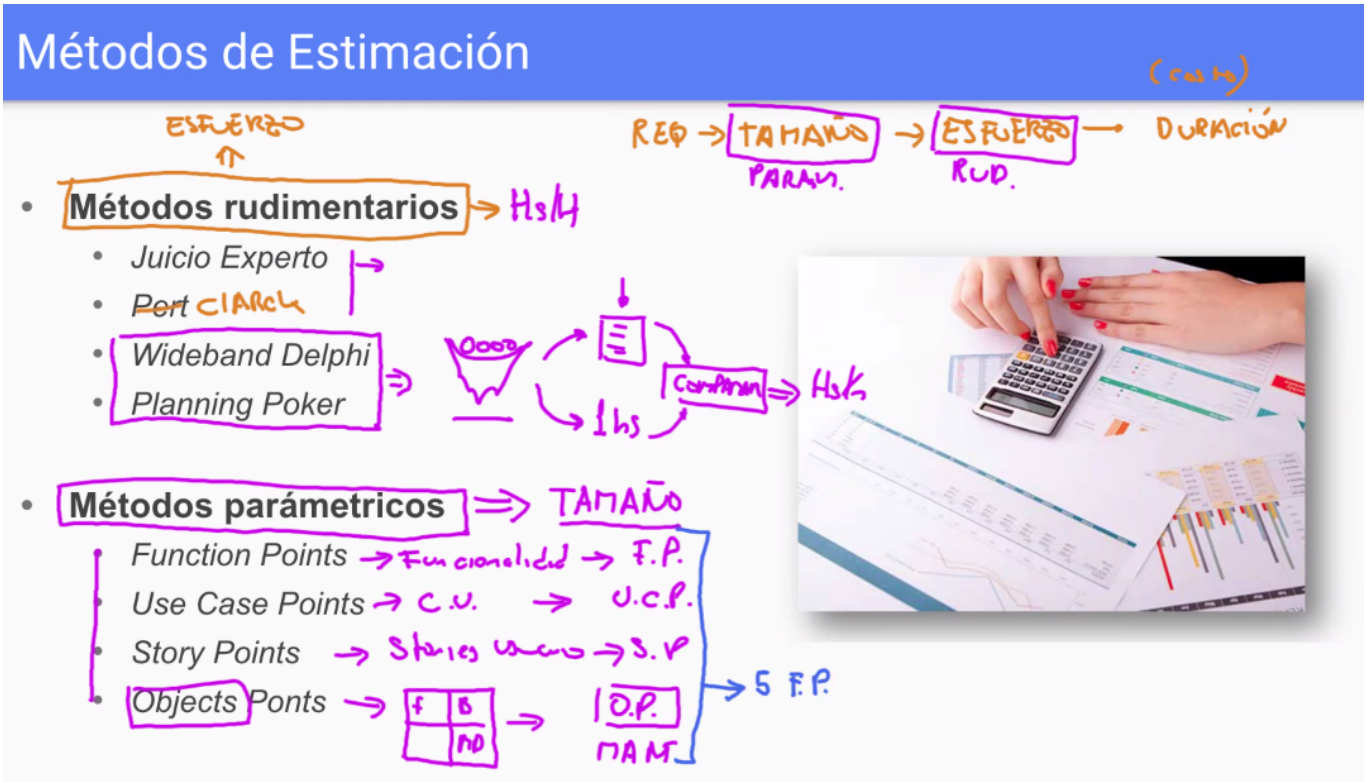
- Diferenciar entre estimaciones, objetivos y compromisos
- Asociar las estimaciones a un % de confiabilidad
- Es recomendable presentar las estimaciones como un rango en vez de como un único valor
- Siempre presentar los supuestos en conjunto con la estimación
- *Ley de Parkinson* → "Toda tarea se expande hasta ocupar el tiempo que tiene asignado"
- Considerar todas las actividades relacionadas al desarrollo de sw, no solamente codificación y testing (análisis, diseño, etc)
- No asumir que solo por el paso del tiempo y de las fases de un proyecto se avanza con menor incertidumbre en las estimaciones
- Recolectar datos históricos para tener como referencia

Ciclo de la Estimación



- Estimar: Comenzar a estimar el tamaño para derivar el esfuerzo y el costo.
- Medir: Mientras evoluciona el proyecto, ir midiendo las variables tamaño, esfuerzo y costo.
- Registrar: Dejar claras las mediciones
- Analizar: Razones de desvíos. Delta de medicion. Ver cuales fueron las causas de esos desvíos
- Calibrar: Ajustar c/u de las variables y parámetros que intervienen en el proceso de estimación.
- Volver a estimar:
    - Mismo proyecto pero con mas información que al comienzo del mismo
    - Nuevos proyectos con el proceso ajustado por la "calibración"

Métodos de estimación



Métodos rudimentarios

Lo que estiman es **esfuerzo** (en realidad no, se contradicen, pero todos son con el estómago), son técnicas que resultan en cantidad hs/hombre que lleva hacer la tarea de forma ininterrumpida. A partir de experiencias se sacan conclusiones.

- Juicio Experto: alguien con experiencia define una estimación (totalmente estomacal) Rápida de hacer.
- Pert/Clarck: conjunto de 4/6 personas que estiman como el juicio experto pero se ponderan esas estimaciones. Rápida.
- Wideband Delphi: El equipo define cuanto es el esfuerzo que nos va a llevar hacer la actividad. Conjunto de personas que tiene que estimar se reunen, analizan los requerimientos, y en forma individual estiman cada tarea. Al final comparan y eligen que tanto les va a llevar.
- Planning Poker: El equipo define cuanto es el esfuerzo que nos va a llevar hacer la actividad. Conjunto de personas que tiene que estimar se reunen, analizan los requerimientos, y en forma individual estiman cada tarea. Al final comparan y eligen que tanto les va a llevar.

Planning Poker

Se estima tamaño. Variación del método Wideband Delphi, bueno para proyectos on-going o a largo plazo. Método estomacal. Todo el equipo participa, se fija el tiempo en el que se va a estimar (1h aprox). Todas las user stories tienen que estar priorizadas y leidas

previamente por el equipo. Se miden las user stories en story points siguiendo una escala o serie (generalmente la de fibonacci). Definir tarea pivot (el estándar para poder comparar). Definir la velocidad del equipo (cant de user stories por sprint). Dar la definición de terminado de una user story.

La dinámica es:

- El moderador plantea la user story (sería el product owner si fuera scrum, no el scrum master bobina)
- Cada integrante propone una estimación sin saber la de los demas
- Dar vuelta las cartas al mismo tiempo
    - El mayor y menor número dan explicaciones
    - Se vuelve a estimar y repetir esto hasta que haya un consenso
- Se anota el ID de la User story en el pizarrón en la columna de terminados

No deja de ser un estómago consensuado

Métodos paramétricos

Lo que estiman es **tamaño** (mmmm dudoso), que luego puede derivar en duración/esfuerzo.

- Function Points: evalúa la funcionalidad del sistema, aplica una serie de reglas (está en el paper), obtiene una unidad de tamaño que se llama **function point** (FP)
- Use Case Points: Similar a Function points, pero mide los casos de uso y obtiene la unidad **Use Case Point** (leer paper)
- Story Points: mide en función de las user stories y termina siendo **story points**. En una user storie uno captura un requerimiento basado en lo que el usuario quiere hacer sobre el sistema
- Object Points: define que el sistema es un conjunto de objetos o componentes y cuenta cuantos objetos son modificados por el requerimiento que se esta estimando (ver final del ppt) (orientado mas al mantenimiento, segun chelo)

Function Points

Un sistema se puede dividir en 5 clases:

- External Inputs (EI): Data del pasa desde afuera del sistema hacia adentro, la data puede ser usada para mantener uno o mas *Internal logical files* (ILF). (Agregar al carrito)
- External Outputs (EO): Data pasa del interior del sistema hacia afuera. Una EO puede o no actualizar un ILF. La data crea reportes o output files para enviar a otras

aplicaciones, eso se representa como *derived information*. ()
- External Inquiry (EQ): Proceso donde se realiza input y output pero el input no actualiza ningún ILF y el output no contiene derived information. (Consultar estado de pedido)
- Internal Logical Files (ILFs): Grupo de información correlacionada que reside en el sistema y se mantiene con inputs externos.
- External Interface Files (EIFs): Grupo de data correlacionada que reside fuera del sistema y se usa para referencia solamente, la mantiene otra aplicación. Es el ILF de otro sistema.

Se categoriza el sistema y en base a la siguiente tabla se calculan los *Unadjasted Function Points*

Luego se ajusta en base a una tabla y un cálculo predefinido.

Use Case Points

Basado en el *function points method*, el objetivo de su creación fue desarrollar un método mas sencillo para proyectos orientados a objetos. El método requiere que se pueda contabilizar el número de transacciones en cada caso de uso. Una **transacción** es un evento que ocurre entre un actor y el sistema.

Los actores se dividen en:

- Simple: Una API definida. Factor de peso 1
- Average: Otro sistema interactuando por medio de algún protocolo (TCP/IP). Factor de peso 2
- Complex: Una persona interactuando con una interfaz gráfica. Factor de peso 3

Sumando los actores que intervienen se consigue el *Unadjasted actor weight* (UAW)

Los casos de uso se clasifican en:

- Simple: 3 o menos transacciones. Factor de peso 5
- Average: entre 4 y 7 transacciones. Factor de peso 10
- Complex: mas de 7 transacciones. Factor de peso 15

Sumando los casos de uso se consigue el *Unadjasted use case weight* (UUCW)

El *Unadjasted use case point* (UUCP) se consigue sumando el UAW y UUCW

Usando unas tablas de factores técnicos y ambientales se calculan:

$$TCF = 0.6 + (.01 * TFactor)$$

$$EF = 1.4 + (-0.03 * EFactor)$$

Finalmente:

$$UCP = UUCP * TCF * EF$$

Karner propone usar 20hs/h por cada use case point. Schneider y Winters recomiendan que los factores ambientales determinen la cantidad de horas hombre por caso de uso (dependiendo de la tabla antes descripta). Otra posibilidad es estimar 36hs/h por cada use case point

Extra

Si tengo un proyecto donde el tiempo es una constraint, puedo utilizar timebox development para encararlo, reduciendo el alcance de esos timebox y lo que va a aumentar es el esfuerzo. El costo deriva de ese esfuerzo que requiero.

Método para estimar (Matriz)

- Saber que voy a contar
- Contar cuantos tengo de cada cosa
- Como voy a categorizar cada uno de esos elementos que identifique, para ver que peso tienen (no todo se puede contar igual, no hay misma complejidad)
- Identificar cuanto pesa cada una de esas categorías
- Identificar valores de ajuste y ver con cuanto lo ajusto