CC51A – Ingeniería de Software

Estimación de Costos y Tiempos

Sergio Ochoa D.

Parte de este material ha sido cedido por el Prof. Jaime Navón (www.ing.puc.cl/~jnavón).

Estructura de la Presentación

- Medidas del Software
- Predicciones
- Métodos de Estimación
- Conclusiones

Medidas del Software

Medidas del Software

Planeación => estimar tiempos y recursos => estimar tamaño del software => medida del software

Medida

Proceso por el cual se asignan números o símbolos a entidades de un set determinado, para describir algún atributo de ellas.

- Altura: (Juan, 180) (Julia, 165) (Michael 205)
- Peso: [Juan, 75] [Julia, 60) [Michael 95]

Lo importante es que si Michael es más alto que Juan, la medida de la altura debe ser también mayor

Medidas del Software

Medida directa, indirecta o predicción

Medida directa: se observa la entidad y se aplica el instrumento de medida directamente (peso, estatura)

*Medida indirecta:*se mide en forma directa otra entidad (o más de una) y luego se usan relaciones conocidas entre ellas (velocidad= distancia/tiempo)

Predicción: queremos una estimación de tamaño de una entidad que puede no existir en ese momento

Predicción

Similar a medición indirecta !

- Establecer una conexión entre las entidades medidas directamente y las entidades cuya medida será predicha
- Refinar la conexión hasta llegar a una o más fórmulas que permitan traducir las medidas directas en medidas aproximadas

La diferencia clave está en que en la predicción no es posible establecer una conexión completa => sólo se puede aproximar la medida correcta.

Predicción

Similar a medición indirecta !

- Sólo es posible predecir si tenemos buenas medidas en qué basarnos!
- A pesar de lo obvio de esta afirmación muchas veces los administradores muestran gran interés en estimación pero no en mediciones de software.
- El esfuerzo de desarrollar un software, siempre es una actividad de **PREDICCION**
- La predicción precisa requiere medición cuidadosa de atributos claves en proyectos ya completados.

Mediciones de Software

- *Razones de baja aceptación en la industria:*
  - administradores no saben qué hacer con las mediciones
  - falta de la validación necesaria => falta de confianza en las medidas

  Ejemplo 1

  *El modelo **COCOMO** simple relaciona el esfuerzo E (meses-hombre) con el tamaño S (MLOCS) de acuerdo a:*

  $$E = a*S^b$$

  Donde a y b son parámetros determinados por el tipo de software a ser desarrollado.

  Para usar este modelo para predecir el esfuerzo en la etapa de captura de requisitos, necesitamos primero determinar (predecir) los parámetros y luego el tamaño del eventual sistema.

Mediciones de Software

- El modelo por si solo no es suficiente para efectuar una predicción
- Un sistema de predicción consiste en un modelo matemático, más un conjunto de procedimientos orientados a obtener variables y parámetros
- Estrictamente, COCOMO es más que un método de estimación de costos, es un sistema de predicción de costos

Sistemas de Predicción

Sistemas de Predicción - *Ejemplo 2*

Albrecht desarrolló la idea de Puntos de Función (FPs) para medir el atributo de funcionalidad percibida por el usuario en los documentos de especificación.

Validación => demostrar que S1 > S2 => FP(S1) > FP(S2)

En la práctica nadie ha validado esta medida en esta forma, y los FPs se usan no como medidas sino como parte de un sistema de predicción de costos (alternativo a COCOMO)

Sistemas de Predicción

Validación de un Sistema de Predicción

- Se debe establecer empíricamente la precisión de la predicción (comparando el modelo con puntos conocidos)
- Involucra experimentación y testeo de hipótesis

Validación de una Medida

- Se debe asegurar que la medida es una caracterización numérica adecuada del atributo en cuestión
- No es necesario que sirva para predecir algo o que tenga o no utilidad

Estimación **Wideband-Delphi**

El Método *Wideband-Delphi:* Originado en Rand Corporation, perfeccionado por B.Boehm.

La idea fundamental es usar varios expertos que hacen estimaciones independientes y luego hacerlos converger hacia una estimación única.

1º. Cada experto recibe las especificaciones del programa y un formulario de estimación.

2º. Se reúnen a conversar sobre suposiciones, dudas, etc.

3º. Cada uno lista las tareas y produce una estimación.

4º. Las estimaciones son recogidas por un moderador quien tabula los resultados y los devuelve a los expertos (estimaciones, promedio, mediana, etc).

Estimación **Wideband-Delphi**

5º. Se reúnen nuevamente y discuten las tareas.

6º. Se vuelve a la tercera etapa (nueva estimación).

La discusión entre los expertos a menudo clarifican aspectos y producen cambios en las estimaciones para la etapa siguiente.

El método produce estimaciones bastante precisas.

Estimación con Lógica Difusa

Método de Lógica Difusa (Fuzzy-Logic): Desarrollado por Putman, se basa en comparar (en líneas gruesas) con información histórica de productos anteriores

Se construye una tabla (escala de tipo logarítmica) con rangos y subrangos de tamaño de proyectos previos. Por ejemplo si el programa más pequeño es de 104 LOC y el más grande de 17.243 LOC ...

	bajo	*interm*	*alto*
muy pequeño	*104*	*173*	*288*
pequeño	*288*	*481*	*802*
mediano	*802*	*1338*	*2230*
grande	*2230*	*3719*	*6202*
muy grande	*6202*	*10341*	*17243*

Estimación con Lógica Difusa

- Las filas indican el tamaños (en LOC), y las columnas la complejidad de los programas (o sistemas).
- El estimador selecciona una de las categorías y subcategorías, comparando el nuevo proyecto con los proyectos anteriores dentro del rango considerado.
- Luego toma como referencia (principalmente) aquellos proyectos previos que están en esa categoría.

Estimación a través de Componentes Estándares

El método de *Componentes Estándares:* se basa en mantener una base de datos histórica con información de componentes usados en proyectos previos, en varios niveles de abstracción: subsistemas completos, módulos, interfaces de usuario, etc.

- Se estima cuántas de cada una de ellas habrá en el nuevo proyecto (estimado, máximo y mínimo).
- Se combina esto, ponderando 4 veces el más probable y una vez los máximos y mínimos (4*est + máx + min)/6.

Estimación a través de Componentes Estándares

Ejemplo:

Componente	*LOC*	*min*	*est*	*max*	*prob*	*LOC*
Módulo A	*932*	*11*	*18*	*22*	*17.5*	*16310*
Módulo B	*543*	*35*	*40*	*44*	*39.8*	*21611*
	…	…	…	…	…	…
	…	…	…	…	…	…

Total: 546359

Estimación a través de Puntos de Función

Puntos de Función: *Probablemente el más popular, inventado por Albrecht en IBM en 1979*

Se identifican 5 funciones básicas:

- ***Inputs*** (pantallas o formularios usados para captura)
- ***Outputs*** (pantallas o reportes que la aplicación produce)
- ***Consultas*** (usuario interroga o pide información)
- ***Archivos***
- ***Interfaz*** (archivos compartidos de entrada o salida, parámetros, etc).

Estimación a través de Puntos de Función

Número	*Tipo*	*Peso*	*Total*
8	*Inputs*	*x4*	*32*
12	*Outputs*	*x5*	*60*
4	*Consultas*	*x4*	*16*
2	*Archivos*	*x10*	*20*
1	*Interfaz*	*x7*	*7*

Total Sin Ajustar: 135

Factor de Complejidad: 1.06

Puntos de Función: 143

Estimación a través del Factor de Complejidad

Para calcular el factor de complejidad se consideran 14 aspectos, otorgándosele a cada uno, una influencia de 0 a 5, generándose así un ***puntaje*** que va de 0 a 70.

El factor de complejidad estará dado por:

FC = 0.65 + 0.01 * Puntaje

FC se mueve en el rango de 0.65 hasta 1.35.

Estimación a través del Factor de Complejidad

Los factores son:

- *comunicaciones*
- *objetivos de desempeño*
- *tasa de transacciones*
- *eficiencia para usuario*
- *proceso complejo*
- *facilidad de instalación*
- *varios sitios*
- *funciones distribuidas*
- *configuración sobrecargada*
- *entrada de datos on-line*
- *actualización en línea*
- *reuso*
- *facilidad de operación*
- *facilidad de mantención*

Estimación Probe

El Método *PROBE*: es aplicable en ambientes de OOP (por lo menos el diseño)

diseño conceptual

↓

identificación de principales objetos (tipo, tamaño)

↓

calcular proyección de LOC

↓

estimar tamaño

↓

calcular intervalo de predicción

Estimación Probe

Diseño Conceptual

- Objetos y funcionalidad. No necesariamente diseño definitivo, sólo para estimación.
- Se clasifican objetos (tipo y tamaño). Se busca en base de datos histórica información sobre lo más parecido que hemos desarrollado.

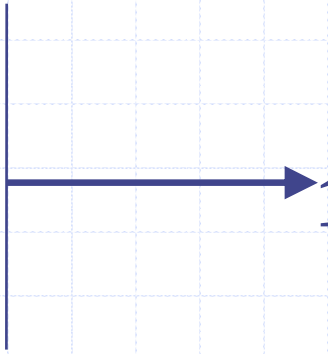
Ejemplo:

Objeto: Matrix

Tipo: Data

Métodos: 13

Tamaño: mediano



8.84 LOC/Method

114.9 LOCs

Conclusiones

- Los modelos teóricos sirven, pero tienen muchas limitaciones.
- A pesar que las técnicas de medición, estimación y predicción se conocen desde hace mucho tiempo, es muy común que se acepten deadlines imposibles para el proyecto, desde el principio.
- Muchas veces el plazo de desarrollo es fijado por gente del área comercial o por ejecutivos sin efectuar ningún tipo de cálculo serio.
- Aunque muchas veces es difícil, el Ingeniero Jefe del Proyecto debe ser muy claro desde un principio y defender enérgicamente los tiempos reales obtenidos científicamente.

Conclusiones

- En muchas ocasiones el encargado del proyecto se pone la soga al cuello sólo sin que nadie realmente lo esté forzando a un schedule tan ajustado.

"Cualquier comandante que acepta llevar a cabo un plan que el considera defectuoso es culpable; el debe exponer sus razones, insistir en los cambios necesarios y finalmente ofrecer su dimisión antes de ser el instrumento de la derrota de su ejército"

Napoleón

- Es mejor desarrollar un modelo propio, que use la información histórica de la empresa.