Facultad de
INFORMÁTICA

UNIVERSIDAD
NACIONAL
DE LA PLATA

ADMINISTRACIóN DE PROYECTOS

ELSA ESTEVEZ
ecestevez@gmail.com

CONTENIDO

1) Estimación de costos

2) Técnicas de estimación

3) Modelo COCOMO

ESTIMACIÓN DE COSTOS

Estimación de Costos: predicciones de cuanto tiempo, esfuerzo y perfiles de RRHH son requeridos para construir un sistema de software

Muchas veces se intercambia *estimación de esfuerzo* con *estimación de costos.*

Las estimaciones preliminares son las más difíciles y las menos exactas

En Ingeniería de Software somos notoriamente inexactos para calcular tiempo y costo

ESTIMACIÓN DE COSTOS

A diferencia de otras profesiones donde se puede tomar ventaja de las tareas repetitivas, esto no ocurre en ISW.

Difieren:

1) dominio de aplicación,
2) hardware,
3) herramientas,
4) técnicas,
5) personal

En ISW somos mas *creadores* que *constructores*

Problemas de Estimación:

1) Problemas Políticos: cuando las estimaciones se convierten en objetivos, cuando se ajusta el precio por conveniencia
2) Problemas Técnicos: No existen datos históricos para estimar

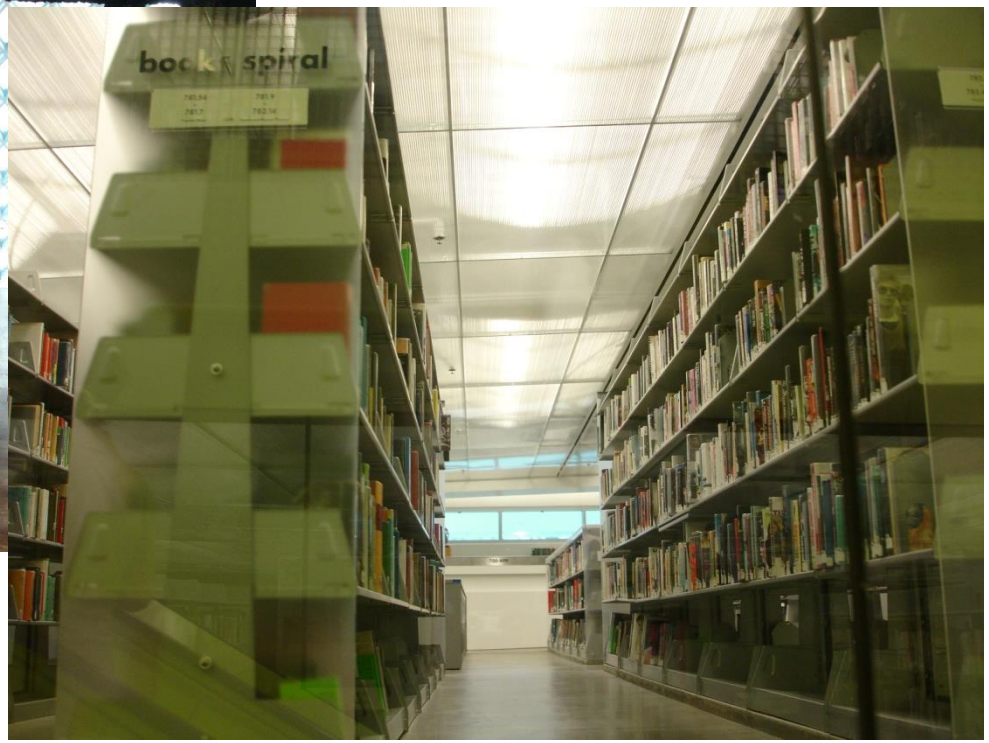
CÓMO ESTIMAR? – CASO 1

La Plata: Fiesta de la cerveza artesanal en Meridiano V

Este fin de semana se realizará en el playón de la Estación Provincial de 17 y 71 una nueva edición de la fiesta de la cerveza que organiza la Asociación de Cerveceros artesanales y cuenta con el auspicio de la Municipalidad La Plata.

Cortesia: http://www.cadenaba.com.ar/nota.php?Id=11671

CÓMO ESTIMAR? – CASO 2



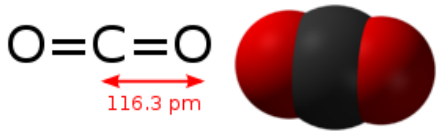
Biblioteca de la Ciudad de Seattle

CÓMO ESTIMAR? – CASO 3

El metano reacciona con el oxígeno para producir dióxido de carbono y agua.

Se sabe que en la reacción, si se mezcla 1 mol de metano que pesa 16 gramos, se produce un mol de dióxido de carbono y el peso resultante de éste es de 44 gramos.

Se necesita estimar la masa de dióxido de carbono que se formaría de la reacción de 70 gramos de metano con los suficientes gramos de oxígeno.



Óxido de carbono (IV)

Nombre (IUPAC) sistemático	
Óxido de carbono (IV)	
General	
Otros nombres	Dióxido de carbono Anhídrido carbónico Gas carbónico
Fórmula semidesarrollada	CO_2
Fórmula molecular	CO_2

Cortesía: Agustín Albanesi, Lic. en Biotecnología

LLUVIA DE IDEAS

Técnica para generar muchas ideas en un grupo.

Requiere la participación espontánea de todos.

La técnica permite obtener nuevas ideas y soluciones creativas e innovadoras, rompiendo paradigmas establecidos.

El clima de participación y motivación asegura mayor calidad en las decisiones tomadas por el grupo, más compromiso con la actividad y un sentimiento de responsabilidad compartido por todos.

TÉCNICAS DE ESTIMACIÓN 1

Opinión Experta: toma ventaja de la experiencia de un personal de desarrollo senior.

El desarrollador describe los parámetros del proyecto y el experto hace predicciones basadas en experiencias previas.

Analogía: los estimadores comparan el proyecto propuesto con proyectos pasados.

Identifican similitudes y diferencias.

Es mas visible. Exige definir características claves.

TÉCNICAS DE ESTIMACIÓN 2

Descomposición: El análisis se focaliza en el producto o en las tareas requeridas para construirlo.

Se basa en la descomposición del producto en componentes y de las actividades en tareas.

Se basan en casos promedios o experiencias pasadas.

TÉCNICAS DE ESTIMACIÓN...

Modelos: son técnicas que identifican contribuyentes claves al esfuerzo, generando fórmulas matemáticas que relacionan estos items al esfuerzo.

Estas técnicas se pueden aplicar con los siguientes enfoques:

1) *Bottom-Up*: comienza con las partes de menor nivel y provee estimaciones para cada una de ellas.

2) *Top-Down*: estima el producto o proceso completo. Las estimaciones para cada componente son calculadas como porciones relativas del todo.

ESTIMACIÓN DE COSTOS – USOS

La Estimación de Costos es parte del planeamiento de cualquier actividad de ingeniería

La diferencia en Ing.de Software es que el costo principal son los recursos humanos

La Estimación de Costos tiene dos usos:

1) en planificación: se necesita saber cuantos recursos va a insumir
2) en control: se necesita saber cuanto se hizo y cuanto falta

Se necesitan *métodos predictivos* para estimar la complejidad del software antes de que sea desarrollado.

CÁLCULO DEL ESFUERZO

El esfuerzo se calcula mediante la fórmula:

$PM_{inicial} = c\ KLOC^{k}$

donde:

1) PM: esfuerzo en personas mes
2) *c* y *k* constantes dadas por el modelo. $k > 1$

Se puede corregir mediante *Conductores de Costos*.

CONDUCTORES DE COSTOS

Se pueden clasificar en:

1) Atributos del Producto: confiabilidad, complejidad...

2) Atributos Computacionales: restricciones de tiempo de ejecución, de almacenamiento...

3) Atributos del Personal: existe personal experimentado ...

4) Atributos del Proceso: se utilizan herramientas de software sofisticadas...

CONDUCTORES DE COSTO COCOMO ORIGINAL

Atributos del Producto	Atributos del Proceso
Confiabilidad requerida	Uso de prácticas de programación modernas
Tamaño de la base de datos	Uso de herramientas de software
Complejidad del producto	Planificación requerida

Atributos computacionales	Atributos del Personal
Restricciones tiempo de ejecución	Capacidad de análisis
Restricciones de almacenamiento	Experiencia en la aplicación
Volatibilidad de la máquina virtual	Capacidad de programación
Tiempo de optimización	Experiencia en lenguaje de programación
Experiencia en la máquina virtual	

PASOS BÁSICOS

Los pasos generales son:

1) estimar el tamaño del software y usar la fórmula del modelo para estimar esfuerzo inicial

2) revisar la estimación usando conductores de costos u otro factor dado por el modelo

3) aplicar las herramientas del modelo a la estimación del paso 2 para determinar el total del esfuerzo.

No es sabio confiar ciegamente en los resultados del modelo.

Es menos sabio ignorar el valor de las herramientas que complementan el juicio experto y la intuición

COCOMO

COCOMO: Constructive Cost Model.

Desarrollado en la década del '70 por Boehm.

Revisado con una nueva release en 1995.y en el 2000.

Es una colección de tres modelos:

1) Básico: aplicable cuando se conoce muy poco del proyecto

2) Intermedio: aplicable luego de la especificación de requerimientos

3) Avanzado: aplicable cuando se termina el diseño

COCOMO – CÁLCULO DE ESFUERZO

Todos utilizan la misma fórmula:

$$E = a\, S^b\, F$$

donde:

E: esfuerzo en personas mes

S: tamaño medido en KSDI (K-delivered source instructions)

F: Factor de ajuste (igual a 1 en el modelo básico)

a,b: se obtienen de tablas del modelo en función del tipo de sistema

COCOMO – CLASIFICACIÓN DE SISTEMAS

Clasifica los sistemas en:

1) orgánicos: involucra procesamiento de datos, uso de bases de datos y se focaliza en transacciones y recuperación de datos.
   Ejemplo: sistema de facturación

2) embebido: contiene software de tiempo real que es una parte integral de un sistema mayor basado en hardware.
   Ejemplo: control de ascensores

3) semi-embebido: entre orgánico y embebido – presenta mayor procesamiento de transacciones.
   Ejemplo: monitoreo de una red

COCOMO – MODELO DE ESTIMACIÓN

El modelo básico aplica las siguientes fórmulas y valores:

	a	b	c	d
orgánico	2.40	1.05	2.50	0.38
embebido	3.00	1.12	2.50	0.35
semi-embebido	3.60	1.20	2.50	0.32

Personas Mes (PM) = a*(KDSI^b)

Tiempo de Desarrollo (TD) = c*(PM^d)

PM = Personas necesarias para hacer el proyecto en 1 mes.

KDSI = Miles de líneas de código entregables

COCOMO – APLICACIÓN 1

Cuando se conoce muy poco del proyecto, se utiliza COCOMO Básico con *F=1*

Cuando se conoce un poco mas: el lenguaje, herramientas a utilizar se puede aplicar COCOMO intermedio.

Se *eligen* los conductores de costos de una tabla que presenta 15.

La importancia de cada conductor de costo es clasificada en una escala ordinal con seis puntos: Muy Baja, Baja, Nominal, Alta, Muy Alta, Extra Alta.

COCOMO – FACTORES DE AJUSTE

Se denominan también Atributos de Costos o Conductores de Costos.

Tratan de capturar el impacto del entorno del proyecto en el costo de desarrollo.

De un análisis estadístico de más de 100 factores que influencian el costo, Boehm retuvo 15 de ellos para COCOMO.

Se agrupan en cuatro categorías:

1) atributos del producto
2) atributos del hardware
3) atributos del personal
4) atributos del proyecto.

COCOMO – ATRIBUTOS DEL PRODUCTO

1) RELY: garantía de funcionamiento requerida al software

2) DATA: tamaño de la base de datos

3) CPLX: complejidad del producto

COCOMO – ATRIBUTO RELY

Rely: indica las posibles consecuencias para el usuario en el caso que todavía existan defectos en el producto.

1) Muy Baja: el efecto de un fallo del software simplemente trae como consecuencia la inconveniencia de corregir el fallo

2) Baja: el efecto de un fallo software es una pérdida fácilmente recuperable para los usuarios

3) Nominal: el efecto es una moderada pérdida para los usuarios, pero es una situación de la que se puede salir sin excesiva dificultad

4) Alta: el efecto es una gran pérdida financiera o una inconveniencia masiva humana

5) Muy Alta: el efecto es una pérdida de vidas humanas.

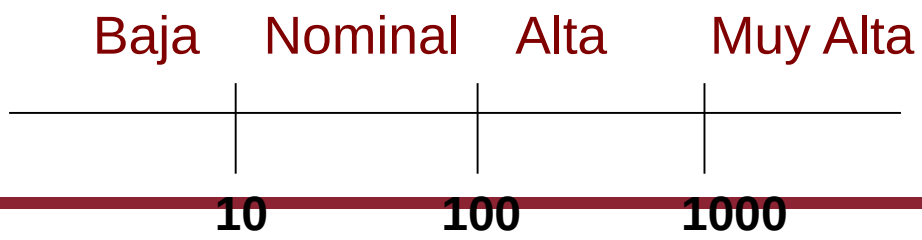
COCOMO – ATRIBUTO DATA

Data: indica el tamaño de la base de datos a desarrollar en relación con el tamaño del programa.

Existen cuatro segmentos con la razón 10-100-1000, que determinan las puntuaciones de 'bajo' a 'muy alto'.

Se define por el cociente

$$\frac{D = \text{tamaño de la base de datos en bytes}}{P \quad \text{tamaño del programa en DSI}}$$



COCOMO – ATRIBUTO COMPLEJIDAD

CPLX: indica la complejidad de cada módulo y se utiliza para determinar la complejidad compuesta del sistema.

La puntuación puede variar de Muy Baja si el módulo está compuesto de expresiones matemáticas simples a Extra Alta para módulos que utilizan muchos recursos de planificación.

COCOMO – ATRIBUTOS DEL HARDWARE

1) TIME: limitaciones en el porcentaje del uso de la CPU.

2) STOR: limitaciones en el porcentaje del uso de la memoria.

3) VIRT: volatilidad de la máquina virtual.

4) TURN: frecuencia de cambio en el modelo de explotación.

COCOMO – ATRIBUTO USO DE CPU

TIME: se expresa en el porcentaje de tiempo de ejecución disponible.

Se basa en la presunción de que siempre será más exigente para un programador escribir un programa que tiene una restricción en el tiempo de ejecución.

Es Nominal cuando el porcentaje es el 50%, y Extra Alta cuando la restricción es del 95%.

COCOMO – ATRIBUTO USO DE MEMORIA

STOR: captura el esfuerzo de programación para que el programa pueda correr en un volumen menor de almacenamiento principal.

Se basa en la presunción de el esfuerzo de programación se incrementa si el programa tiene que correr en un volumen menor del almacenamiento principal.

STOR captura este esfuerzo extra de Nominal cuando la reducción del almacenamiento principal es del 50% a Extra Alta cuando la reducción es del 95%.

COCOMO – ATRIBUTO MÁQUINA VIRTUAL

VIRT: refleja los cambios que puede sufrir la máquina virtual (hardware mas software) durante el desarrollo del software.

VIRT refleja la probabilidad de que ocurran los cambios desde Baja a Muy Alta.

COCOMO – ATRIBUTO MODELO EXPLOTACIÓN

TURN: cuantifica el tiempo de respuesta del ordenador desde el punto de vista del programador. Cuanto mayor sea el tiempo de respuesta, más alto será el esfuerzo humano.

TURN puede variar desde Baja para un sistema interactivo; a Muy Alta, cuando el tiempo medio de respuesta es de más de 12 horas.

COCOMO – ATRIBUTOS DEL PERSONAL

1) ACAP: calificación de los analistas

2) AEXP: experiencia del personal en aplicaciones similares.

3) PCAP: calificación de los programadores.

4) VEXP: experiencia del personal en la máquina virtual.

5) LEXP: experiencia en el lenguaje de programación a usar.

COCOMO – ATRIBUTO CALIFICACIÓN ANALISTAS

ACAP: mide la capacidad del grupo de analistas, en términos de habilidad de análisis, eficiencia y capacidad para cooperar.

Estas habilidades tienen un impacto significativo en el esfuerzo humano.

Cuanto más capaz sea el grupo, menos esfuerzo será necesario.

ACAP puede variar desde Muy Baja a Muy Alta.

COCOMO – EXPERIENCIA EN APLICACIONES

AEXP: mide la experiencia del grupo en una aplicación similar.

La experiencia del grupo tiene una gran influencia en el esfuerzo.

1) Muy Baja: $\leq$ 4 meses experiencia media
2) Baja: 1 año de experiencia media
3) Nominal: 3 años de experiencia media
4) Alta: 6 años de experiencia media
5) Muy Alta: $\geq$ 12 años, o reimplementación de un subsistema

COCOMO – ATRIBUTO CALIFICACIÓN PROGRAMADORES

PCAP: mide la capacidad del grupo de programadores, en términos de habilidad de programación, eficiencia y capacidad para cooperar.

Similar al atributo que mide la calificación de analistas, pero en este caso, se mide al grupo de programadores.

Cuanto más capaz sea el grupo, menos esfuerzo será necesario.

Se aplica a los programadores como grupo, pero no a los programadores individuales.

PCAP puede variar desde Muy Baja a Muy Alta.

COCOMO – EXPERIENCIA EN MÁQUINA VIRTUAL

AEXP: mide la experiencia de los programadores en la máquina virtual.

1) Muy Baja: ≤ 1 mes experiencia media
2) Baja: 4 meses
3) Nominal: 1 año
4) Alta: ≥ 3 años

COCOMO – EXPERIENCIA EN LENGUAJE

LEXP: mide la experiencia de los programadores en la máquina virtual.

Un grupo de programadores con amplia experiencia en un lenguaje determinado programará de una manera mucho más segura, generando un menor número de defectos y de requerimientos humanos.

Puede variar desde Muy Baja a Alta para un grupo de un mes a tres años de experiencia, respectivamente.

1) Muy Baja: $\leq$ 1 mes experiencia media
2) Baja: 4 meses experiencia media
3) Nominal: 1 año experiencia media
4) Alta: $\geq$ 3 años

COCOMO – ATRIBUTOS DEL PROYECTO

1) MODP: uso de prácticas modernas de programación.

2) TOOL: uso de herramientas de desarrollo de software.

3) SCED: limitaciones en el cumplimiento de la planificación.

COCOMO – USO DE PRÁCTICAS MODERNAS

MODP: indica la utilización de modernas prácticas de programación.

Estas prácticas incluyen, por ejemplo, programación estructurada y desarrollo 'top-down'.

1) Muy Baja: no se utilizan prácticas modernas de programación (PMP)

2) Baja: uso experimental de algunas PMP

3) Nominal: experiencia razonable en el uso de algunas PMP

4) Alta: experiencia razonable en gran parte de PMP

5) Extra Alta: uso habitual de PMP

COCOMO – USO DE HERRAMIENTAS

TOOL: indica el uso de herramientas de software.

El uso adecuado de herramientas de software es un multiplicador de la productividad.

La puntuación de TOOL varía desde Muy Baja cuando sólo se utilizan herramientas básicas, a Muy Alta cuando se utilizan herramientas específicas.

COCOMO – LIMITACIONES EN PLANIFICACIÓN

SCED: indica el esfuerzo necesario para cumplir con la planificación.

El tiempo nominal de desarrollo, tal como se define en el modo básico, es el plazo que requiere menor esfuerzo humano.

Cualquier apresuramiento (Muy Baja) o retraso (Muy Alta) demandarán más esfuerzo.

COCOMO – INDICADORES 1

Grado	Muy Baja	Baja	Nominal	Alta	Muy Alta	Extra Alta
Atributos del Producto						
RELY	0.75	0.88	1.00	1.15.	1.40	
DATA		0.94	1.00	1.08	1.16	
CPLX	0.70	0.85	1.00	1.15	1.30	1.65
Atributos del Hardware						
TIME			1.00	1.11	1.30	1.62
STOR			1.00	1.06	1.21	1.56
VIRT		0.87	1.00	1.15	1.30	
TURN		0.87	1.00	1.07	1.15	

COCOMO – INDICADORES 2

Grado	Muy Baja	Baja	Nominal	Alta	Muy Alta	Extra Alta
Atributos del Personal						
ACAP	1.46	1.19	1.00	0.86	0.71	
AEXP	1.29	1.13	1.00	0.91	0.82	
PCAP	1.42	1.17	1.00	0.86	0.70	
VEXP	1.21	1.10	1.00	0.90		
LEXP	1.14	1.07	1.00	0.95		
Atributos del Proyecto						
MODP	1.24	1.10	1.00	0.91	0.82	
TOOL	1.24	1.10	1.00	0.91	0.83	
SCED	1.23	1.08	1.00	1.04	1.10	

COCOMO – APLICACIÓN 2

GRADO	MUY BAJA	BAJA	NOMINAL	ALTA	MUY ALTA	EXTRA ALTA
Atributos del Personal						
ACAP	1.46	1.19	1.00	0.86	0.71	

Ejemplo: Si se estimo la *Capacidad de Análisis* como *Muy Baja*, el factor es 1.46, quiere decir que se debe aumentar el esfuerzo calculado previamente en un 46%.

COCOMO...

Para los proyectos *medios*, COCOMO Intermedio es coincidente con COCOMO Básico

El modelo debe ser calibrado al propio entorno de desarrollo.

Una vez que se han identificado los módulos del sistema, se puede utilizar COCOMO Avanzado o Detallado.

COCOMO Avanzado: se aplica la versión intermedia a nivel de componentes y luego se construye una estimación para el proyecto completo.

COCOMO – FASES DE DESARROLLO

Permite estimar los tiempos para cada una de las fases de desarrollo.

Considera cuatro fases:

1) requerimientos/planes
2) diseño del producto,
3) programación
4) prueba/integración.

COCOMO – FASE DE REQUERIMIENTOS

Requerimientos / Planes - es la primera fase del ciclo de desarrollo.

Se analiza el requerimiento, se muestra un Plan de Producto y se general una especificación completa del producto.

Esta fase consume del 6% al 8% del esfuerzo nominal PM.

Puede durar del 10% al 40% del tiempo nominal de desarrollo TD.

Estos porcentajes dependen del modo y del tamaño (de 2000 LOC a 512000 LOC).

COCOMO – FASE DE DISEÑO

Diseño del Producto - la segunda fase del ciclo de desarrollo.

COCOMO se preocupa de la determinación de la arquitectura del producto y de las especificaciones de los subsistemas.

Esta fase requiere del 16% al 18% del esfuerzo nominal PM.

Puede durar del 19% al 38% del tiempo nominal de desarrollo TD.

COCOMO – FASE DE PROGRAMACIÓN

Programación - la tercera fase del ciclo de desarrollo.

COCOMO se subdivide en dos subfases: diseño detallado y prueba del código.

Esta fase requiere del 48% al 68% del esfuerzo nominal PM.

Puede durar del 24% al 64% del tiempo nominal de desarrollo TD.

COCOMO – FASE DE INTEGRACIÓN

Prueba / Integración – la última fase.

Esta fase consiste principalmente en unir las diferentes unidades ya probadas.

Se utiliza del 16% al 34% del esfuerzo nominal PM.

Puede durar del 18% al 34% del tiempo nominal de desarrollo TD.

COCOMO 2.0

Es una actualización de COCOMO para que se adapte a las nuevas tecnologías, enfoques de OO, etc.

La estimación del proceso en COCOMO 2.0 está basado en tres etapas principales de cualquier desarrollo:

Etapa	Basado en...	Utiliza
1	Prototipos	Puntos objeto
2	Decisiones de arquitectura	Puntos función
3	Diseño detallado	KDSI

COCOMO 2.0

El modelo original de COCOMO resultó muy exitoso, sin embargo su aplicación no es práctica para entornos modernos de desarrollo.

De este modo, surge COCOMO II, cuyos objetivos son:

1) desarrollar modelos de costos y de estimación acordes a las prácticas actuales

2) desarrollar bases de datos de costos y herramientas que soporten una mejora continua del modelo

3) proveer un framework analítico cuantitativo, y un conjunto de herramientas y técnicas para evaluar los efectos de las mejoras en los costos de ciclos de vida y en las planificaciones

COCOMO II – TRES MODELOS

COCOMO II está compuesto por tres modelos:

1) Modelo de la Aplicación: basado en Puntos Objeto

2) Modelo de Diseño Temprano: usado para obtener estimaciones de costo y duración antes de finalizar el diseño de la arquitectura

3) Modelo Post-Arquitectura: el modelo más detallado, con nuevos conductores de costos, y nuevas ecuaciones

BASE DE ESTIMACIÓN PARA COCOMO

COCOMO provee un modelo para estimar costos en base a:

1) KLOC

2) Puntos Objeto (PO)

3) Puntos Función (PF)

COCOMO EN BASE A PO – PASOS 1-2

1) estimar el número de pantallas, reportes y componentes 3GL

2) clasificar la complejidad de esos objetos en: i) simple, ii) media, iii) alta, de acuerdo a:

Pantallas:

Número de vistas contenidas	Número y Fuente de las Tablas de Datos		
	Total < 4 (< 2 server, < 3 cliente)	Total < 8 (2-3 server, 3-5 cliente)	Total +8 (> 3 server, > 5 cliente)
< 3	Simple	Simple	Mediana
3 – 7	Simple	Mediana	Alta
+ 8	Mediana	Alta	Alta

COCOMO EN BASE A PO – PASO 2 CONT.

Informes:

Número de secciones contenidas	Número y Fuente de las Tablas de Datos		
	Total < 4 (< 2 server, < 3 cliente)	Total < 8 (2-3 server, 3-5 cliente)	Total +8 (> 3 server, > 5 cliente)
0 – 1	Simple	Simple	Mediana
2 – 3	Simple	Mediana	Alta
+ 4	Mediana	Alta	Alta

COCOMO EN BASE A PO – PASO 3

3) pesar los objetos de acuerdo a su complejidad en base a la siguiente tabla:

Tipo de Objeto	Peso en base a Complejidad		
	Simple	Media	Alta
Pantalla	1	2	3
Informe	2	5	8
Componente 3GL			10

COCOMO EN BASE A PO – PASOS 4-7

4) determinar los Puntos Objetos – sumar todos los pesos de las instancias y obtener el número de Puntos Objeto (PO)

5) estimar el porcentaje de reuso que se espera y calcular los Nuevos Puntos Objeto (NPO) = PO (100 - %reuso) /100

6) determinar el ratio de productividad, en base a:

PROD = NPO / personas-mes

Experiencia y capacidad de desarrolladores	Muy baja	Baja	Nominal	Alta	Muy alta
Madurez y capacidad del ICASE	Muy baja	Baja	Nominal	Alta	Muy alta
PROD	4	7	13	25	50

7) estimar personas-mes, PM = NPO/PROD

COCOMO EN BASE A PF – PASO 1

1) calcular los puntos función, en base a requerimientos y documentos de diseño. Para calcularlos, contar:

   a) inputs externos (IE)
   b) outputs externos (OE)
   c) consultas externas (CE)
   d) archivos internos (AI)
   e) archivos de interface externos (AE)

COCOMO EN BASE A PF – PASO 2

2) clasificar cada punto función de acuerdo a su complejidad de acuerdo a las siguientes tablas:

Para AI y AE			
Registros	Datos		
	1-19	20-50	+ 51
1	Baja	Baja	Media
2 – 5	Baja	Media	Alta
6 - +	Media	Alta	Alta

Para OE y CE			
Tipos de Archivos	Datos		
	1-5	6-19	+ 20
0 – 1	Baja	Baja	Media
2 – 3	Baja	Media	Alta
4 - +	Media	Alta	Alta

Para IE			
Tipos de Archivos	Datos		
	1-4	5-15	+ 16
0 – 1	Baja	Baja	Media
2 – 3	Baja	Media	Alta
4 - +	Media	Alta	Alta

COCOMO EN BASE A PF – PASOS 3 - 4

3) aplicar pesos de complejidad, de acuerdo a:

Puntos Función	Peso de Complejidad		
	Baja	Media	Alta
Inputs externos	3	4	6
Outputs externos	4	5	7
Consultas externas	3	4	6
Archivos internos	7	10	15
Archivos externos	5	7	10

4) [illegible] de función por complejidad.

COCOMO EN BASE A PF – PASO 5

5) convertir puntos de función a líneas de código, en base a:

Lenguaje	SLOC/PFNA
Ada	71
Basic (compilado)	91
Basic (interpretado)	128
C	128
C++	29
ANSI Cobol 85	91
Fortran 77	105

Lenguaje	SLOC/PFNA
Java	53
Lisp	64
Modula 2	80
Pascal	91
Prolog	64
Generador de reportes	80
Planilla de cálculo	6

COCOMO - ESTIMACIÓN DEL ESFUERZO

En COCOMO II el esfuerzo es expresado en Personas Mes (PM).

$PM_{nominal} = A * (Tamaño)^{B}$

El Tamaño es expresado en KSLOC.

A: intenta cuantificar los efectos multiplicativos en el esfuerzo de proyectos de tamaño creciente

B: intenta medir la economía (o no economía) de escala encontrada en proyectos de diferentes tamaños.

Si B < 1.0 → el proyecto exhibe economía de escala (la productividad aumenta a medida que aumenta el tamaño del producto)

Si B = 1.0 → la economía, o no economía están balanceadas

Si B > 1.0 → el proyecto no exhibe economía de escala (aumento de comunicación, problemas de integración)

COCOMO Y ECONOMÍA DE ESCALA

Fórmula : $E = A * (Tamaño)^B$

E = esfuerzo estimado en personas-mes

A = coeficiente de calibración

Tamaño = medido en Puntos Objeto, Puntos Función, LOC

B = cuenta por la economía (< 1) o no (> 1) de escala

- Ejemplo de economía de escala – uso de herramientas CASE
- Ejemplo de no economía de escala – mayor comunicación, y dependencias

B – ECONOMÍA DE ESCALA

B es una agregación de 5 factores de escala que consideran la economía o no de escala en los proyectos de diferentes tamaños.

Si B < 1.0, hay economía de escala -

Si el tamaño es el doble, el esfuerzo es menos del doble.

La productividad aumenta a medida que el tamaño del producto aumenta.

Algunas economías se pueden obtener mediante el uso de herramientas específicas (CASE, simulaciones, entornos de pruebas)

If B > 1.0, no hay economía de escala.

1) aumento de comunicaciones interpersonales y overhead

2) aumento de esfuerzo de integración.

CÁLCULO DE “B”

B es una agregación de 5 factores de escala .

Cada factor de escala tiene un rango de niveles Muy-Bajo a Extra-Alto.

Cada ratio tiene un peso. El valor específico del peso es llamado factor de escala (SF).

Los factores de escala del proyecto son sumados para determinar el exponente B, de acuerdo con la fórmula:

$$B = 1.01 + 0.01 \sum W_i$$

donde B 0.91(para COCOMO II.2000)

CÁLCULO DE "B"

$B = 1.01 + 0.01 \sum W_j$

Factores
Antecedentes – PREC
Flexibilidad de desarrollo - FLEX
Riesgo de resolución / Arquitectura -RESL
Cohesión del equipo - TEAM
Madurez del proceso - PMAT

Table 10. Scale Factor Values, SF_j, for COCOMO II Models

Scale Factors	Very Low	Low	Nominal	High	Very High	Extra High
PREC	thoroughly unprecedented	largely unprecedented	somewhat unprecedented	generally familiar	largely familiar	thoroughly familiar
SF_j:	6.20	4.96	3.72	2.48	1.24	0.00
FLEX	rigorous	occasional relaxation	some relaxation	general conformity	some conformity	general goals
SF_j:	5.07	4.05	3.04	2.03	1.01	0.00
RESL	little (20%)	some (40%)	often (60%)	generally (75%)	mostly (90%)	full (100%)
SF_j:	7.07	5.65	4.24	2.83	1.41	0.00
TEAM	very difficult interactions	some difficult interactions	basically cooperative interactions	largely cooperative	highly cooperative	seamless interactions
SF_j:	5.48	4.38	3.29	2.19	1.10	0.00
PMAT	The estimated Equivalent Process Maturity Level (EPML) or					
	SW-CMM Level 1 Lower	SW-CMM Level 1 Upper	SW-CMM Level 2	SW-CMM Level 3	SW-CMM Level 4	SW-CMM Level 5
SF_j:	7.80	6.24	4.68	3.12	1.56	0.00

EJEMPLO – FACTOR DE ESCALA - FLEX

Table 12. Development Flexibility Rating Levels

Feature	Very Low	Nominal / High	Extra High
Need for software conformance with pre-established requirements	Full	Considerable	Basic
Need for software conformance with external interface specifications	Full	Considerable	Basic
Combination of inflexibilities above with premium on early completion	High	Medium	Low

Gentileza: COCOMO II – Model Definition Manual
Center for Software Engineering, USC
http://csse.usc.edu/csse/

RESUMEN

Estimación de costos – distintas técnicas

COCOMO

- Tres modelos
- Tipos de sistemas
- Conductores de costos

COCOMO 2.0

- Puntos objeto
- Puntos función
- KDSI

Muchas gracias!

Elsa Estevez
ecestevez@gmail.com
www.elsaestevez.com