

Práctica 2: estimación del coste con COCOMO II

Gestión de Proyectos Software – Curso 2013/2014

Contenido

1	Introducción al software COCOMO.....	1
1.1	Información por módulo	3
1.2	Información de totales	5
1.3	Calibración.....	5
1.4	Snap shot.....	6
1.5	Distribución por fases, Mantenimiento y crear Informes.	6
2	Guion de la práctica	7
2.1	Descripción del proyecto y el entorno de trabajo	7
2.2	Estimación del tamaño.....	8
2.2.1	Tarea 1: Líneas de código equivalente del software reutilizado.....	8
2.2.2	Tarea 2: FP a SLOC.....	9
2.3	Tarea 3: Estimación del personal necesario según la EDT	9
2.4	Tarea 4: Estimación del esfuerzo con COCOMO II en modo Post-Arquitectura.	9
2.5	Tarea 5: Gestión de Riesgos.	9
2.6	Tarea 6: Estimación por fases	10

1 Introducción al software COCOMO

Nota: esta sección es un resumen y traducción del Usermanual.pdf que podéis encontrar en la distribución del software COCOMO II 2000.4

Como se vio en clase, el *USC Center for Software Engineering* (formado por un supergrupo de los autores de COCOMO II), ha desarrollado varios programas derivados de COCOMO además del original.

La última implementación oficial es COCOMO II 2000.4, y puede descargarse desde el siguiente enlace: http://csse.usc.edu/csse/research/COCOMOII/cocomo_downloads.htm.

Al descomprimir el archivo descargado encontraréis la siguiente estructura:

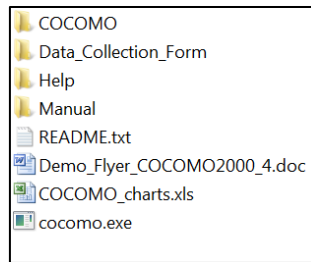


Figura 1 Contenido de COCOMO® II.2000.4 software


Podéis borrar la carpeta *COCOMO*, la cual tiene el mismo contenido (exceptuando dicha carpeta). La carpeta *Data_Collection_Form* contiene formularios de recogida de datos que tampoco utilizaremos. En *Manual* podéis encontrar 2 archivos de ayuda para el uso del software y para refrescar y ampliar teoría sobre COCOMO II: *usermanual.pdf* y *modelmanual.pdf*, respectivamente.

X	Module Name	Module Size	LABOR Rate (\$/month)	EAF	Language	NCM Effort	EST Effort	PROD	COST	INST	Staff	RISK

Total Lines of Code:	0	Estimated	Effort	Sched	PROD	COST	INST	Staff	RISK
Hours/PM:	152.00	Optimistic	0.0	0.0	0.0	0.00	0.0	0.0	
		Most Likely	0.0	0.0	0.0	0.00	0.0	0.0	0.0
		Pessimistic	0.0	0.0	0.0	0.00	0.0	0.0	

Figura 2 CLEF de COCOMO II 2000.4

Los 2 archivos más importantes son la aplicación principal: *cocomo.exe* y *COCOMO_charts.xls*, archivo que solo puede utilizarse una vez hemos editado un proyecto con el otro archivo, y sirve para crear gráficos para la gestión del proyecto, pero asume un desarrollo de Cascada así que no trabajaremos con él.

Al ejecutar este archivo, aparece la interfaz CLEF (Component Level Estimation Form), a través de la cual podemos calcular el esfuerzo para el desarrollo de un software dividido en módulos. Los modos de estimación permitidos son Early Design y Post-architecture; es decir, no nos permite calcular a nivel de Prototipos. Los módulos se añaden pulsando en el botón .

1.1 Información por módulo

Como puede verse la Figura 2, hay una banda verde con información sobre cada módulo en que dividamos el software. Esto es útil por si el software consta de diferentes lenguajes de programación (módulos) que interaccionan entre sí.. Veamos la información de las columnas más importantes:

- **Module size**

El tamaño del módulo se puede insertar de 3 maneras:

- Directamente indicando las SLOC.
- UFP: indicando la información necesaria para que el programa los calcule, o indicarlos directamente; el programa convierte estos a SLOC (la conversión referenciada en el libro de COCOMO II es la Caper Jones). Los 5 tipos de funciones que aparecen corresponden según la terminología de IFPUG vista en clase a : EI, EO, ILF, EIF y EQ.

Function Type	# of Function Points			SubTotal
	Low	Average	High	
Inputs	0	0	0	0
Outputs	0	0	0	0
Files	0	0	0	0
Interfaces	0	0	0	0
Queries	0	0	0	0
Total Unadjusted Function Points				0
Equivalent Total in SLOC				0

Figura 3 Calcular SLOC a partir de Puntos Función no ajustados

- Indicando que será código reutilizado y añadiendo la información necesaria para calcular las ESLOC.

AdaptaInitial SLOC

Language ▼ Non-Specified

0

% Design Modified (DM) 0 %

% Code Modified (CM) 0 %

% Integration Modified (IM) 0 %

Software Understanding (SU) 30 SU

Assesment & Assimilation (AA) 4 AA

Unfamiliarity with Software 0.4 UNFM

% Components Automatically Translated (AT) 0 %

Automatic Translation Productivity (ATPROD) 2400

Computed Adaptation Adjustment Factor 0

Computed ASLOC 0

Figura 4 Calcular ASLOC para módulo con código reutilizado

- **Labor Rate:** Insertar manualmente la cantidad (no importa la unidad, siempre que sea consistente) de dinero que 1 desarrollador cobra al mes por trabajar en ese módulo.
- **EAF:** Pulsando sobre EAF podemos dar valor a los controladores de coste; excepto para SCED cuyo valor se añade pulsando el botón “Schedule” del CLEF.

Como puede verse en la Figura 5, no solo se puede dar el valor para cada controlador sino que se puede indicar mediante *Incr%* (“*inter cost driver*”) que el valor deseado es superior al elegido pero sin llegar a ser el siguiente. Así, por ejemplo, si indicamos un valor High para RELY esto se corresponde con un incremento de x1.10 en EAF. Sin embargo, indicamos que está un 25% más allá de High pero sin llegar a VeryHigh; así el peso incrementado es 1.14.

base + Incr % = rating

Product: RELY DATA DOCU CPLX RUSE

base HI NOM NOM NOM NOM

Incr% 25% 0% 0% 0% 0%

Platform: TIME STOR PVOL

base NOM NOM NOM

Incr% 0% 0% 0%

Personnel: ACAP PCAP PCON APEX LTEX PLEX

base NOM NOM NOM NOM NOM NOM

Incr% 0% 0% 0% 0% 0% 0%

Project: TOOL SITE

base NOM NOM

Incr% 0% 0%

User: USR1 USR2

base NOM NOM

Incr% 0% 0%

EAF is also affected by Schedule

EAF: 1.14

Figura 5 Cálculo del factor de ajuste del esfuerzo

- **Effort Dev:** Estas dos columnas dan la estimación del esfuerzo en el desarrollo en Person-months, aplicando el EAF calculado (Est Effort Dev) y sin aplicarlo (Nom Effot Dev).

- **PROD:** SLOC/(Est Effort Dev más probable).
- **Cost:** Coste más probable.
- **Inst Cost:** coste por instrucción (Cost/SLOC).
- **Staff:** PM más probable/Schedule más probable. Si se quiere reducir el tiempo hay que disminuir SCED, pero esto añade personal y coste monetario.
- **RISK:** nivel de riesgo, calculado según el estudio *Heuristic Risk Assessment Using Cost Factors*. Madachy 1997. Esto no forma parte del modelo COCOMO II.

1.2 Información de totales

En la parte inferior del CLEF encontramos un panel con tablas de valores totales para todos los módulos.

		Module Size	LABOR Rate (\$/month)	EA	Language	NOM Effort DEV	EST Effort DEV	PROD	COST	INST COST	Staff	RISK
X	<sample>	S:10000	0.00	1.21	Non-Specified	37.0	44.9	222.6	0.00	0.0	3.4	2.6
X	<sample>	S:30000	0.00	1.00	Non-Specified	11.1	11.1	270.3	0.00	0.0	0.8	0.0

Figura 6 Panel de totales sumando todos los módulos

Además de mostrar el total de SLOC y cuál es la carga de trabajo predefinida de 1 PM (editable en Parameters → person month), podemos ver las estimaciones más probables, optimistas y pesimistas para el Esfuerzo en desarrollo, tiempo de desarrollo (en meses), productividad, Coste, coste por instrucción y cantidad de personal necesario. Ojo, Sched se refiere a meses para el desarrollo, no confundir con SCED indicado pulsando el botón Schedule de la parte superior derecha del CLEF.

1.3 Calibración

Mediante el comando “Calibrate”, podemos añadir este proyecto a una lista de calibración. En esta lista, podemos indicar en cada proyecto cuál fue el Esfuerzo y el tiempo de desarrollo **reales**. El sistema usará las diferencias con las estimaciones de todos los proyectos para calibrar los exponentes y parámetros de las ecuaciones de esfuerzo y tiempo de desarrollo al pulsar Calibrate → Compute.

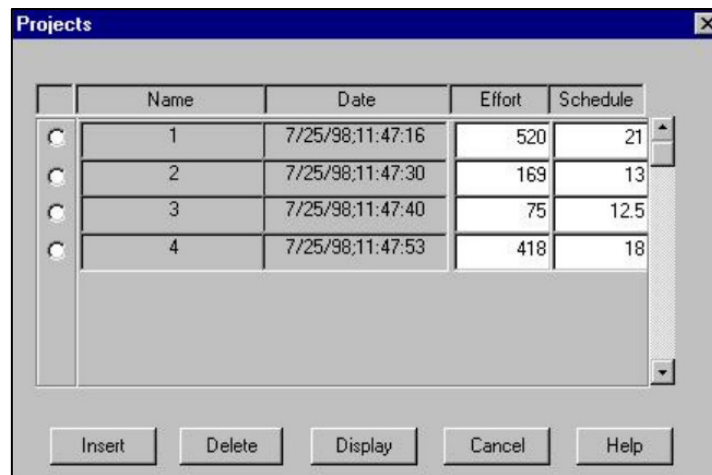


Figura 7 Calibración de COCOMO II insertando información real

1.4 Snap shot

El comando Edit→Snap Shot nos permite utilizar la mayor aportación del método COCOMO II: analizar la variación en el esfuerzo estimado según diferentes valores asignados a los exponentes y multiplicadores del esfuerzo.

Current Project		EST	Sched	PROB	COST	INST	FSWP	RISK
Total EDSI:	37791	Optimistic	683.4	31.4	55.3	4090503.35	108.2	21.7
Effort (PM):	202.7	Most Likely	854.3	34.1	44.2	5113129.19	135.3	25.1
Productivity:	186.4	Pessimistic	1067.8	36.9	35.4	6391411.49	169.1	28.9

Figura 8 snap shot de solo 1 proyecto

En la Figura 8, tras pulsar el botón Snap, se almacena la información de las estimaciones del proyecto actual. Después, podemos hacer modificaciones a los valores del proyecto y volver a hacer otro snap shot, de forma que tendremos juntas las estimaciones de 2 proyectos. Si creemos que nos interesa más la configuración actual, entonces podemos pulsar el botón *Revert*. Así, solo se pueden comparar 2 configuraciones a la vez.

1.5 Distribución por fases, Mantenimiento y crear Informes.

Mediante el menú Phase, podemos acceder a la distribución de las estimaciones de Esfuerzo y Tiempo de desarrollo según 2 ciclos de vida: Cascada y MBase. Las fases de MBase son similares a RUP, y por lo tanto puede utilizarse para desarrollos en espiral.

Respecto al mantenimiento (menú Maintenance), es posible calcular el esfuerzo de mantenimiento hasta 5 años. Para entender todos los valores que pide, es necesario que

repaséis la sección '2.9 COCOMO II – Distribución y mantenimiento' del Tema 4 de la asignatura.

Finalmente, al finalizar tu proyecto, puedes pulsar en File→Report, para generar un archivo .rpt, el cual puede abrirse con cualquier editor de texto.

2 Guion de la práctica

Nota: Esta práctica es un resumen y traducción de la sección 3.2 del libro de referencia Software Cost Estimation with COCOMO II. Boehm et al. 2000. Podéis obtener el libro en la biblioteca si queréis profundizar en este u otros ejemplos.

Vamos a estimar el coste de desarrollo de un Sistema de Procesado de Transacciones (TPS), para más tarde evaluar diferentes alternativas y gestión de riesgos. Veremos cómo el software de COCOMO II 2000.4 nos ayuda a abstraernos de las ecuaciones tan complicadas para centrarnos solo en la evaluación de distintas configuraciones del proyecto.

2.1 Descripción del proyecto y el entorno de trabajo

El TPS es un sistema cliente-servidor. La parte del cliente está formada por un nodo principal de 50 estaciones de trabajo, unidas por LAN entre sí y con otros nodos más pequeños. El servidor también está en dicha LAN. También se permite acceso a Internet a través de un cortafuegos. El tipo de transacciones realizadas pueden ser compras o búsquedas en bases de datos.

La empresa encargada del desarrollo tiene un equipo con más de 4 años de experiencia en aplicaciones cliente-servidor con lenguaje C/C++. No hay problemas de recursos materiales, y se estima necesidad en entrenarles en Java, bases de datos persistentes y programación web. Se plantea desarrollar los conocimientos del equipo al menos durante 2 semanas anuales.

Por otra parte, es una empresa cuyos procesos y prácticas están en continuo desarrollo, habiendo obtenido el nivel 2 de CMM y están cerca de obtener el nivel 3.

Los problemas que pueden servir para la identificación de riesgos son:

- Los jefes de proyecto aprenden por su cuenta a partir de las buenas prácticas de la empresa, pero no reciben educación directa. Se debate si esto sería útil.
- Se debate si usar C++ o pasarse a Java.
- El equipo suele pasarse mucho por defecto a exceso en la estimación del esfuerzo con técnicas Delphi o similares.
- El equipo está dividido en qué unidad de tamaño utilizar: SLOC o FP, así que hay inconsistencia en las estimaciones.

Las buenas prácticas de la empresa piden que se realice una estimación del esfuerzo de 2 maneras para luego comparar resultados:

- Estructura de desglose de trabajo (EDT): la EDT es uno de los métodos más clásicos; consiste en dividir los entregables a programar en unidades pequeñas y sumar la estimación del tamaño (método bottom-up) para cada una de estas unidades. La estimación de cada unidad se realizará por experiencia propia del equipo.
- COCOMO II.

La duración del proyecto se ha firmado bajo contrato que será de 18 meses.

Como ya sabemos, antes de estimar el esfuerzo hay que estimar el tamaño del software.

2.2 Estimación del tamaño

Tras varias entrevistas con el cliente, se identifican funcionalidades y se estima el tamaño mostrados en la Tabla I.

Tabla I Tamaño de distintos módulos a desarrollar

Componente	Funciones	Tamaño	Notas
Sistema	Controlar comunicaciones	18 KSLOC nuevas 10 KSLOC reutilizadas	
	Conversión de protocolos		
	Autenticación		
	Bibliotecas de clases		
Aplicaciones de usuario	Informes y pantallas	800 SLOC nuevas	Uso de Visual Basic (4GL) para diseño de interfaces
	Interfaz de creación de aplicaciones		
Diagnóstico de fallos	Tests de serie	300 FPs	Se utilizan redes-neuronales, basadas en Int. Artificial, lo cual es un concepto muy nuevo para el equipo.
	Aislamiento de fallos		
	Gestión de recuperación		

Estima el tamaño en SLOC total. Para ello tendrás que calcular las ESLOC del software reutilizado, y convertir los FPs a SLOC (indica la fuente utilizada para decidir el factor de conversión).

2.2.1 Tarea 1: Líneas de código equivalente del software reutilizado.

Para calcular el tamaño equivalente del código que se reutiliza, necesitarás dar o calcular diferentes parámetros. Para guiarte puedes rellenar la Tabla II, y luego insertar esos valores en la parte adecuada del CLEF para conseguir los ESLOC de reutilización. Algunos de estos parámetros realmente no hacen falta: indica cuáles y por qué.

Tabla II Parámetros para calcular el tamaño equivalente del código reutilizado

DM	
CM	
IM	El equipo estima un 50%
AAF	
SU	
AA	

UNFM	
ASLOC	
ESLOC	

2.2.2 Tarea 2: FP a SLOC

Tienes que transformar el tamaño funcional a longitud. Sabes que existen tablas de conversión, y que el software usado en esta práctica también te facilita la conversión. Sin embargo, en esta etapa de estimación aún no sabes si usarás C, C++ o Java en el desarrollo. Piensa cómo resolver esto y calcula los SLOC.

2.3 Tarea 3: Estimación del personal necesario según la EDT

Sabiendo que la organización ha estimado mediante la EDT que el proyecto requerirá 29 065 horas, calcula la cantidad de personal necesario según esta estimación.

2.4 Tarea 4: Estimación del esfuerzo con COCOMO II en modo Post-Arquitectura.

Dividid el software en los módulos que queráis. Un ejemplo puede ser insertar 4 módulos: Sistema-Reutilizar, Sistema-Nuevo, Aplicaciones y Diagnóstico.

Al decidir los valores de los Factores de Escalado y de los Multiplicadores del Esfuerzo, podéis dar los mismos en cada módulo como primera (o última si tenéis buen resultado) aproximación. El valor que le deis a cada factor depende de vosotros; ¡comienza la subjetividad! Si queréis un poco de guía en las decisiones, podéis utilizar las tablas en el archivo *modelman.pdf* a partir de la página 30. Podéis considerar que la estimación se hace para programación en C, y el módulo de Diagnóstico un 3GL genérico. No estamos interesados en estimación económico, dejad los costos a 0.

En la memoria de práctica debéis entregar una tabla explicando el motivo (en 1 ó 2 líneas) por el que habéis dado una puntuación a cada parámetro.

¿Qué diferencia hay en el tiempo que estima necesario COCOMO II respecto a los 18 meses prefijados en el EDT? ¿Y qué cantidad de horas y personal se estima necesario, asumiendo $1PM=152h/m$? Compara las estimaciones de la EDT, teniendo en cuenta que la estimación de la EDT incluyó la fase de captura de requisitos (1600 horas), fase que COCOMO II no incluye en su estimación.

Si las estimaciones distan mucho, revisa los valores que has dado a cada parámetro; es posible que en segunda o tercera vuelta veas que has asumido algunas ideas de forma errónea. Si acercas las predicciones <http://imapas.net/2012/12/05/5-herramientas-android-para-trabajar-con-scrum/>, incluye en tu memoria la primera y la mejor. Quizás encuentres útil usar la opción de Snap Shot.

2.5 Tarea 5: Gestión de Riesgos.

Como jefe de proyecto, has identificado los riesgos que pueden aparecer en este proyecto. A continuación tienes que averiguar, para cada riesgo, un valor cuantitativo de cuánto puede afectar a tu estimación de esfuerzo (en %). Finalmente, indica los 3 riesgos con mayor impacto

potencial y qué medida propones para mitigar el impacto en caso de aparecer, o para eliminar el riesgo.

Tabla III Influencia de los riesgos en el proyecto

Riesgo	SFs o EMs cambiados	% influencia
El jefe se enseña así mismo, esto puede afectar a las capacidades del equipo.		
Es posible que se utilice un nuevo lenguaje (JAVA)		
La empresa añade nuevos procesos a su gestión		
Por motivos económicos, es posible que la empresa sustituya trabajadores por otros menos experimentados. (PCON asume que el nuevo trabajador es igual de experimentado)		
El campo de las redes neuronales es tan complejo para tu empresa y para el cliente, que es posible que los requisitos de este módulo varíen mucho, hasta un 20%.		

2.6 Tarea 6: Estimación por fases

Supón que el ciclo de vida de tu proyecto será incremental. Describe la estimación final de esfuerzo, tiempo y personal necesario en cada fase de una metodología RUP; indicando la estimación más probable y el rango en que esto puede variar debido a los riesgos.