
MODELO CONSTRUCTIVO DE COSTOS COCOMO

for

Repositorio WEB para el
almacenamiento y consulta de
documentos generados por los
estudiantes de UDEC

Version 1.0 review

Prepared by Ariel Camilo Sanchez Lopez
Yonatan Estiven Castañeda

Universidad de Cundinamarca

September 30, 2022

Contents

1	Introducción	3
1.1	Definición de COCOMO	3
1.2	Ecuación de Estimación de Esfuerzo	4
1.3	Ecuación de estimación de cronograma	4
1.4	Factores de Escala	5
1.5	SLOC: Reglas de conteo de líneas de código	6
1.6	AAF: Factores de Ajuste de Adaptación	7
1.7	Multiplicadores de esfuerzo	8
1.8	Ecuación de Estimación de Esfuerzo de Desarrollo	9
2	Estimaciones del Sistema de Información	10
2.1	Estimación COCOMO II Metodología RUP	10
2.2	Controladores de escala de software	10
2.3	Controladores de costos de software	10
2.3.1	Producto	10
2.3.2	Personal	10
2.3.3	Plataforma	11
2.3.4	Proyecto	11
2.4	Resultado	11

1 Introducción

1.1 Definición de COCOMO

COCOMO (Modelo de costo constructivo) es un paquete de software interactivo orientado a la pantalla que ayuda en la planificación presupuestaria y la estimación del cronograma de un proyecto de desarrollo de software. A través de la flexibilidad de COCOMO, un gerente de proyecto de software (o líder de equipo) puede desarrollar un modelo (o múltiples modelos) de proyectos para identificar problemas potenciales en recursos, personal, presupuestos y cronogramas tanto antes como durante la implementación del paquete de software potencial siendo desarrollado.

El paquete de software COCOMO se basa en el modelo de estimación de costos y cronogramas del software: Modelo de Costos Constructivos versión II.

(COCOMO II). Esta es la versión revisada recientemente del modelo de costo constructivo original (COCOMO) publicado por primera vez por el Dr. Barry Boehm en su libro *Software Engineering Economics*, Prentice-Hall (1981) y predecesores de Ada COCOMO (1989). El modelo actual se describe en [Boehm et al. 1995] Los objetivos principales del esfuerzo COCOMOII 1998 son:

- Desarrollar un modelo de estimación de costos y cronogramas de software ajustado a las prácticas del ciclo de vida de las décadas de 1990 y 2000.
- Desarrollar bases de datos de costos de software y capacidades de soporte de herramientas para la mejora continua del modelo.
- Proporcionar un marco analítico cuantitativo y un conjunto de herramientas y técnicas para evaluar los efectos de las mejoras en la tecnología del software en los costos y cronogramas del ciclo de vida del software.

las mejoras en la tecnología del software en los costos y cronogramas del ciclo de vida del software. El modelo COCOMOII completo incluye tres etapas. La etapa 1 admite la estimación de los esfuerzos de creación de prototipos o composición de aplicaciones. La etapa 2 admite la estimación en la etapa de diseño inicial de un proyecto, cuando se sabe menos sobre los factores de costo del proyecto. La etapa 3 admite la estimación en la etapa posterior a la arquitectura de un proyecto.

Esta versión de USC COCOMOII implementa fórmulas de la etapa 3 para estimar el esfuerzo, el cronograma y el costo necesarios para desarrollar un producto de software. También proporciona el desglose del esfuerzo y el cronograma en fases y actividades del ciclo de vida del software del manual COCOMO original. Estos siguen siendo razonablemente válidos para proyectos de software de modelos en cascada, pero deben

interpretarse para proyectos que no sean en cascada.

1.2 Ecuación de Estimación de Esfuerzo

$$PM = \prod_{i=1}^{17} (EM_i) \cdot A \cdot \left[1 + \frac{BRAK}{100} \cdot Size \right]^{(0.91 + 0.01 \sum_{j=1}^5 SF_j)} + \left(\frac{ASLOC \cdot \left(\frac{AT}{100} \right)}{ATPROD} \right)$$

$$Size = KNSLOC + \left[KASLOC \cdot \left(\frac{100-AT}{100} \right) \cdot \frac{(AA+SU+0.4 \cdot DM+0.3 \cdot CM+0.3 \cdot IM)}{100} \right]$$

$$B = 0.91 + 0.01 \sum_{j=1}^5 SF$$

Símbolo	Descripción
A	Constante, actualmente calibrado en 2.45.
AA	Evaluación y Asimilación
ADAPT	Porcentaje de componentes adaptados (representa el esfuerzo requerido para comprender el software).
AT	Porcentaje de componentes que se traducen automáticamente.
ATPROD	Productividad de la traducción automática.
BRAK	Rotura: porcentaje de código desechado debido a la volatilidad de los requisitos.
CM	Porcentaje de código modificado.
DM	Porcentaje de diseño modificado.
EM	Multiplicadores de esfuerzo: RELY, DATA, CPLX, RUSE, DOCU, TIME, STOR, PVOL, ACAP, PCAP, PCON, AEXP, PEXP, LTEX, HERRAMIENTA, SITIO
IM	Porcentaje de integración y prueba modificada.
KASLOC	Tamaño del componente adaptado expresado en miles de líneas de código fuente adaptadas.
KNSLOC	Tamaño del componente expresado en miles de nuevas líneas de código fuente.
PM	Persona Meses de esfuerzo estimado.
SF	Factores de escala: PREC, FLEX, RESL, EQUIPO, PMAT.
SU	Comprensión del software (cero si DM = 0 y CM = 0)

Table 1.1: Convenciones Ecuación de Estimación de Esfuerzo

1.3 Ecuación de estimación de cronograma

Determine el tiempo de desarrollo (TDEV) con un esfuerzo estimado, PM, que excluye el efecto del multiplicador de esfuerzo SCED:

$$TDEV = \left[3.67 \times (\overline{PM})^{0.28+0.2 \times (B-1.01)} \right] \cdot \frac{SCED\%}{100}$$

$$B = 0.91 + 0.01 \sum_{j=1}^5 SF_j$$

Símbolo	Descripción
PM	Persona Meses de esfuerzo estimado de los modelos Early Design o Post-Architecture (excluyendo el efecto del multiplicador de esfuerzo SCED).
SF	Factores de escala: PREC, FLEX, RESL, EQUIPO, PMAT
TDEV	Tiempo para desarrollar.
SCRED	Calendario.
SCRED%	El porcentaje de compresión/expansión en el multiplicador de esfuerzo SCED.

Table 1.2: Convenciones Ecuación de Estimación de Cronograma

1.4 Factores de Escala

La ecuación de estimación de cronograma define el exponente, B, utilizado en la ecuación de estimación de esfuerzo. La Tabla 1.1 proporciona los niveles de calificación para los conductores de la escala COCOMO II. La selección de controladores de escala se basa en la lógica de que son una fuente significativa de variación exponencial en el esfuerzo de un proyecto o variación de productividad. Cada controlador de escala tiene un rango de niveles de calificación, desde muy bajo hasta muy alto. Cada nivel de calificación tiene un peso, W, y el valor específico del peso se denomina factor de escala. Los factores de escala de un proyecto, Wi, se suman en todos los factores y se utilizan para determinar un exponente de escala, B, a través de la siguiente fórmula:

$$B = 0.91 + 0.01 \sum_{j=1}^5 SF_j$$

Por ejemplo, si a los factores de escala con una calificación Extra-Alta se les asigna un peso de (0), entonces un proyecto de 100 KSLOC con calificaciones Extra Altas para todos los factores tendrá $SF_j = 0$, $B = 1.01$ y un esfuerzo relativo $E = 1001.01 = 105$ p. m. Si a cada uno de los factores de escala con calificación Muy baja se les asigna un peso de (5), entonces un proyecto con calificaciones Muy bajas (5) para todos los factores tendrá $SF_j = 5$, $B = 1.26$ y un esfuerzo relativo $E = 331$ p. m. Esto representa una gran variación, pero el aumento involucrado en un cambio de una unidad en uno de los factores es solo alrededor del 4,7%.

Factores	Muy Bajo	Bajo	Nominal	Alto	Muy Alto	Extra Alto
PREC	En gran medida sin precedentes	Algo así como sin precedentes	Algo así como sin precedentes	Generalmente conocido	En gran medida conocido	A fondo conocido
FLEX	Riguroso	Ocasionalmente relajación	Un poco de relajación	General conformidad	Alguna conformidad	A Objetivos generales
RESL	Poco (20%)	Algunos (40%)	A menudo (60%)	Generalmente (75%)	En su mayoría (90%)	Lleno (100%)
TEAM	Muy difícil interacciones	Algunas interacciones difíciles	Básicamente cooperativa interacciones	En gran medida cooperativa	Altamente cooperativo	Altamente cooperativo

Table 1.3: Factores de escala para los modelos de diseño inicial y post-arquitectura de COCOMO II

PMAT

Media ponderada de las respuestas afirmativas al Cuestionario de Madurez CMM.3

1.5 SLOC: Reglas de conteo de líneas de código

En COCOMO II, la declaración de fuente lógica se ha elegido como la línea de código estándar. Definir una línea de código es difícil debido a las diferencias conceptuales involucradas en la contabilización de sentencias ejecutables y declaraciones de datos en diferentes lenguajes. El objetivo es medir la cantidad de trabajo intelectual puesto en el desarrollo del programa, pero surgen dificultades cuando se trata de definir medidas coherentes en diferentes idiomas. Las roturas por cambio de requisitos también complican el dimensionamiento. Para minimizar estos problemas, se utiliza la lista de verificación de definición del Instituto de ingeniería de software (SEI) para una declaración de fuente lógica al definir la línea de medida de código. El Software Engineering Institute (SEI) ha desarrollado esta lista de verificación como parte de un sistema de listas de verificación de definiciones, formularios de informe y formularios complementarios para respaldar las definiciones de medición [Park 1992] [Goethert et al. 1992].

La Tabla 1.3 muestra una parte de la lista de verificación de definiciones tal como se aplica para respaldar el desarrollo del modelo COCOMO II. Cada marca de verificación en la columna "Incluye" identifica un tipo de declaración o atributo particular incluido en la definición y viceversa para las exclusiones. Otras secciones de la definición aclaran

los atributos de la declaración para el uso, la entrega, la funcionalidad, las replicaciones y el estado de desarrollo. También hay aclaraciones para declaraciones específicas de lenguaje para ADA, C, C++, CMS-2, COBOL, FORTRAN, JOVIAL y Pascal.

Se realizaron algunos cambios en la definición de línea de código que se aparta de la definición predeterminada proporcionada en [Park 1992]. Estos cambios eliminan categorías de software, que generalmente son pequeñas fuentes de esfuerzo del proyecto. No se incluyen en la definición el software disponible comercialmente (COTS), el software proporcionado por el gobierno (GFS), otros productos, bibliotecas de soporte de idiomas y sistemas operativos u otras bibliotecas comerciales. El código generado con generadores de código fuente no está incluido, aunque se tomarán medidas con y sin código generado para respaldar el análisis.

La "definición de línea de código COCOMO II se puede calcular de varias maneras. Una forma es utilizar el programa de software Amadeus [Amadeus 1994] [Selby et al. 1991]. Otro programa de software es el contador de códigos.

1.6 AAF: Factores de Ajuste de Adaptación

Adaptación del Código Existente COCOMO no solo es capaz de estimar el costo y el cronograma de un desarrollo iniciado desde "cero", sino que también puede estimar el costo y el cronograma de productos que se basan en un código ya existente. Las consideraciones de adaptación también se han incorporado en COCOMO, donde se calculará una estimación para KSLOC. Este valor se sustituirá en lugar del SLOC que se encuentra en las ecuaciones ya discutidas. Esta adaptación del código utiliza un conjunto adicional de ecuaciones que se utilizan para calcular el recuento final de las instrucciones de origen y el costo y el cronograma relacionados. Estas ecuaciones utilizan los siguientes valores como componentes:

- Líneas de código fuente adaptadas (ASLOC). El número de líneas fuente de código adaptadas del software existente utilizado en el desarrollo del nuevo producto.
- Porcentaje de modificación del diseño (DM). El porcentaje del diseño del software adaptado que recibió modificaciones para cumplir con los objetivos y el entorno del nuevo producto.
- Porcentaje de Modificación de Código (CM). El porcentaje del código del software adaptado que recibe modificaciones para cumplir con los objetivos y entorno del nuevo producto.
- Porcentaje de integración requerida para software modificado (IM). El porcentaje de esfuerzo necesario para integrar y probar el software adaptado para combinarlo en el nuevo producto.
- Porcentaje de esfuerzo de reutilización debido a la comprensión del software (SU).

- Porcentaje de esfuerzo de reutilización debido a Evaluación y Asimilación (AA).
- Desconocimiento del programador con el software (UNFM)

El AAF es el factor de ajuste de adaptación. El AAF es el grado calculado en el que el software adaptado afectará el desarrollo general.

1.7 Multiplicadores de esfuerzo

Hay una serie de factores que contribuyen al tiempo y esfuerzo de entrega de un proyecto. Se encontró que la productividad del desarrollo se ve afectada por factores adicionales que se encuentran bajo los encabezados: atributos del producto, atributos de la plataforma, atributos del personal y atributos del proyecto.

Los atributos del producto se refieren a las restricciones y requisitos impuestos al proyecto a desarrollar. Estos incluidos:

- Confiabilidad de software requerida (RELY)
- Tamaño de la base de datos (DATOS)
- Coincidencia de la documentación con las necesidades del ciclo de vida (DOCU)
- Complejidad del producto (CPLX)
- Reutilización requerida (RUSE)

Los atributos de la plataforma se refieren a las limitaciones impuestas al esfuerzo de desarrollo por el hardware y el sistema operativo que se utilizan para ejecutar el proyecto. Estas limitaciones se enumeran a continuación.

- Restricción de tiempo de ejecución (TIME).
- Restricción de almacenamiento principal (STOR)
- Volatilidad de la plataforma (PVOL)

Los atributos del personal se refieren al nivel de habilidades que posee el personal. Las habilidades en cuestión son capacidad profesional general, capacidad de programación, experiencia con el entorno de desarrollo y familiaridad con el dominio del proyecto. Estas habilidades se caracterizan a continuación.

- Capacidades del analista (ACAP)
- Experiencia en aplicaciones (AEXP)

- Capacidades del programador (PCAP)
- Experiencia en plataforma (PEXP)
- Experiencia en lenguaje de programación (LEXP)
- Continuidad del Personal (PCON)

Los atributos del proyecto se refieren a las restricciones y condiciones bajo las cuales se lleva a cabo el desarrollo del proyecto. Los problemas que afectan el desarrollo son:

- Uso de herramientas de software (HERRAMIENTA)
- Desarrollo multisitio (SITIO)

Estos 16 factores se incorporan al cálculo de un esfuerzo y cronograma estimados. Cada uno de los factores tiene asociadas hasta seis valoraciones. Estas calificaciones son muy baja, baja, nominal, alta, muy alta y extra alta. Cada calificación tiene un número real correspondiente basado en el factor y el grado en que el factor puede influir en la productividad. Una calificación inferior a 1 denota un factor que puede disminuir el cronograma y el esfuerzo. Una calificación superior a 1 denota un factor que extiende el cronograma o el esfuerzo. Por último, una calificación igual a 1 no amplía ni disminuye el cronograma y el esfuerzo (esta calificación se denomina nominal).

Estos 16 factores (o multiplicadores de esfuerzo) se incorporan a las fórmulas de estimación de programa y esfuerzo multiplicándolos entre sí. El valor numérico del *i*-enésimo factor de ajuste (hay 16) se denomina EM_i y su producto se denomina factor de ajuste o EAF. El esfuerzo real, PM_{total} , es el producto del esfuerzo nominal por el EAF.

Además de los 16 factores EAF, hay dos factores definidos por el usuario denominados $USR1$ y $USR2$. Todos sus valores iniciales están establecidos en 1. Se pueden redefinir mediante el elemento de menú Parámetros-Definido por el usuario EAF.

Un multiplicador de esfuerzo final, el Cronograma de Desarrollo Requerido (SCED, por sus siglas en inglés) se trata por separado ya que opera a nivel general del proyecto en lugar de variar potencialmente de un módulo a otro.

1.8 Ecuación de Estimación de Esfuerzo de Desarrollo

$$PMT_{total} = (SCED) \times PM_{Nominal} \times \prod_{i=1}^{18} EM_i$$

2 Estimaciones del Sistema de Información

2.1 Estimacion COCOMO II Metodología RUP

Método de Dimensionamiento del tamaño de software **Lineas de Código Fuente**
Proyecto nuevo a 4500 líneas de código.

2.2 Controladores de escala de software

- Precedencia (**Muy Alto**)
- Flexibilidad de Desarrollo (**Alto**)
- Arquitectura/Resolución de Riesgos (**Nominal**)
- Cohesión de equipo (**Alto**)
- Madurez del proceso (**Nominal**)

2.3 Controladores de costos de software

2.3.1 Producto

- Confiabilidad de Software requerido (**Bajo**)
- Tamaño de la Base de Datos (**Bajo**)
- Complejidad del Producto (**Bajo**)
- Desarrollo para la reutilización (**Nominal**)
- Coincidencia de la documentación con las necesidades del ciclo de vida (**Bajo**)

2.3.2 Personal

- Capacidad del analista (**Muy Alto**)
- Capacidad del programador (**Muy Alto**)
- Continuidad del personal (**Muy Alto**)
- Experiencia de aplicación (**Muy Alto**)

- Experiencia de plataforma (**Muy Alto**)
- Experiencia con lenguaje y el conjunto de herramientas (**Alto**)

2.3.3 Plataforma

- Limitación de Tiempo (**Nominal**)
- Restricción de almacenamiento (**Nominal**)
- Volatilidad de la plataforma (**Nominal**)

2.3.4 Proyecto

- Uso de herramientas de software (**Alto**)
- Desarrollo Multisitio (**Alto**)
- Cronograma de Desarrollo requerido (**Nominal**)

2.4 Resultado

Fase	Esfuerzo	Horario - Meses	Personal
Inicio	0.1	0.25	2
Elaboración	0.3	0.75	2
Construcción	1.1	1.25	2
Transición	0.2	0.5	2

Table 2.1: Distribución de la fase de adquisición

Fase	Inicio	Elaboración	Construcción	Transición
Administración	0.0	0.0	0.1	0.0
CM	0.0	0.0	0.1	0.0
Requisitos	0.0	0.1	0.1	0.0
Diseño	0.0	0.1	0.2	0.0
Implementación	0.0	0.0	0.4	0.1
Evaluación	0.0	0.0	0.3	0.0
Despliegue	0.0	0.0	0.0	0.1

Table 2.2: Distribución de esfuerzo de software para RUP

Meses/Personas	Meses	Personas
1.4	2.1	2.3

Table 2.3: Tabla de Resultados