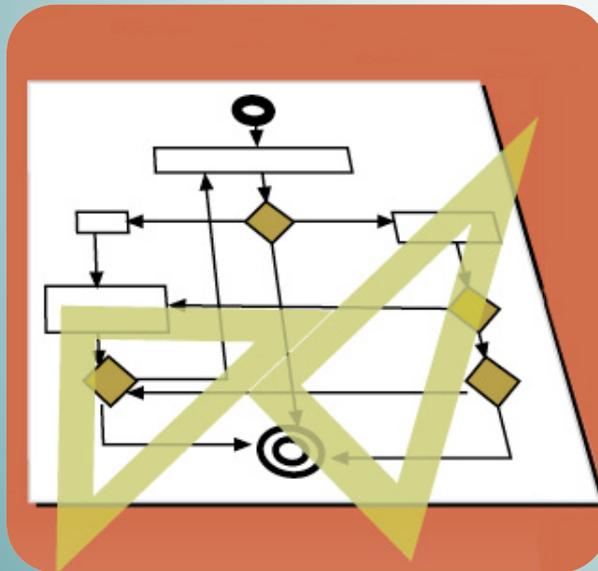


Ingeniería del Software II

Tema 05. Gestión de Costes en Ingeniería de Sistemas Software



Pablo Sánchez Barreiro

DPTO. DE MATEMÁTICAS, ESTADÍSTICA Y
COMPUTACIÓN

p.sanchez@unican.es

Este tema se publica bajo Licencia:

[Creative Commons BY-NC-SA 3.0](https://creativecommons.org/licenses/by-nc-sa/3.0/)

Índice

- 1 **Introducción**
- 2 Principales Retos de la Estimación Software
- 3 Métricas de Tamaño: Puntos de Función
- 4 Técnicas de Estimación de Esfuerzos
- 5 Estimación de Costes con Cocomo II
- 6 Control de Costes
- 7 Sumario y Bibliografía

Problemas de la Estimación de Coste

La historia de Paco, el recién licenciado

Cliente ¿Cuánto cuesta hacer la aplicación que te he descrito?

Paco (presupuestando por lo bajo) 300 €.

(Paco se arruina y cierra su empresa)

(Se retira de la escena cabizbajo entre música melancólica de violines)

Problemas de la Estimación de Coste

La historia de Lola, la recién licenciada

Cliente ¿Cuánto cuesta hacer la aplicación que te he descrito?

Lola (presupuestando por lo alto) 12.000€.

Cliente Ok, muchas gracias, (mirando al techo) lo consultaré con mi asesor financiero ... (sale por la puerta sin ninguna intención de comprar).

Problemas de la Estimación de Coste

La historia de Juan y María

Juan Oye María, ¿hay dinero para suscribirnos a este portal que vende imágenes para iconos tan chulas?

María ¿Cuánto cuesta?

Juan 9000€ por año.

María Sí, tenemos 9001€ para gastos del proyecto.

(Fin de la escena)

Juan (al día siguiente) María, me hace falta un taco nuevo de notas adhesivas amarillas.

María No tenemos dinero, cuestan 1.35€ y nos queda 1€ de dinero para gastos.

Problemas en la Gestión de Costes

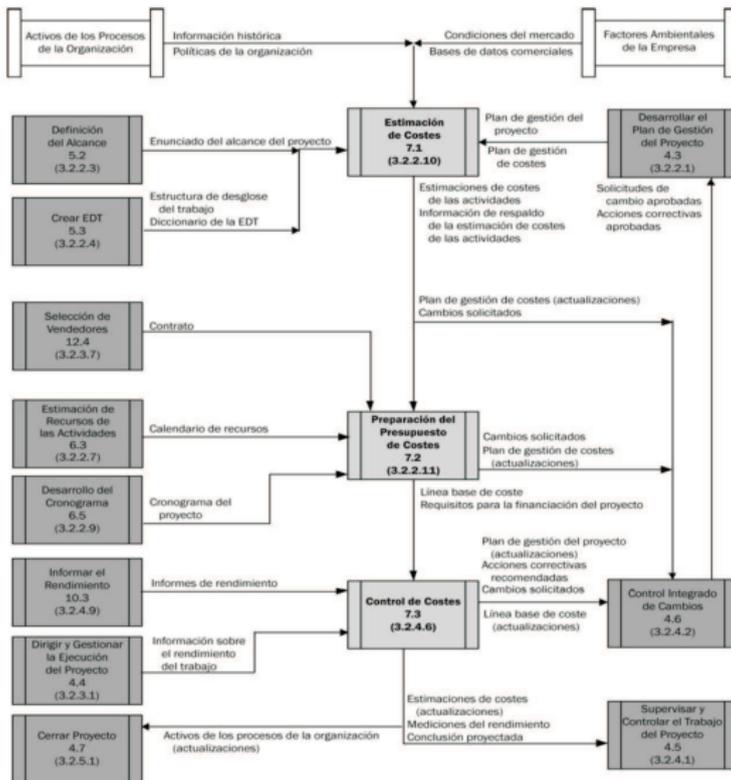
- 1 Presupuestos demasiado bajos.
- 2 Presupuestos demasiado altos.
- 3 Ingresos y Gastos no controlados.

Objetivo del Tema

Objetivo de la Gestión de Costes

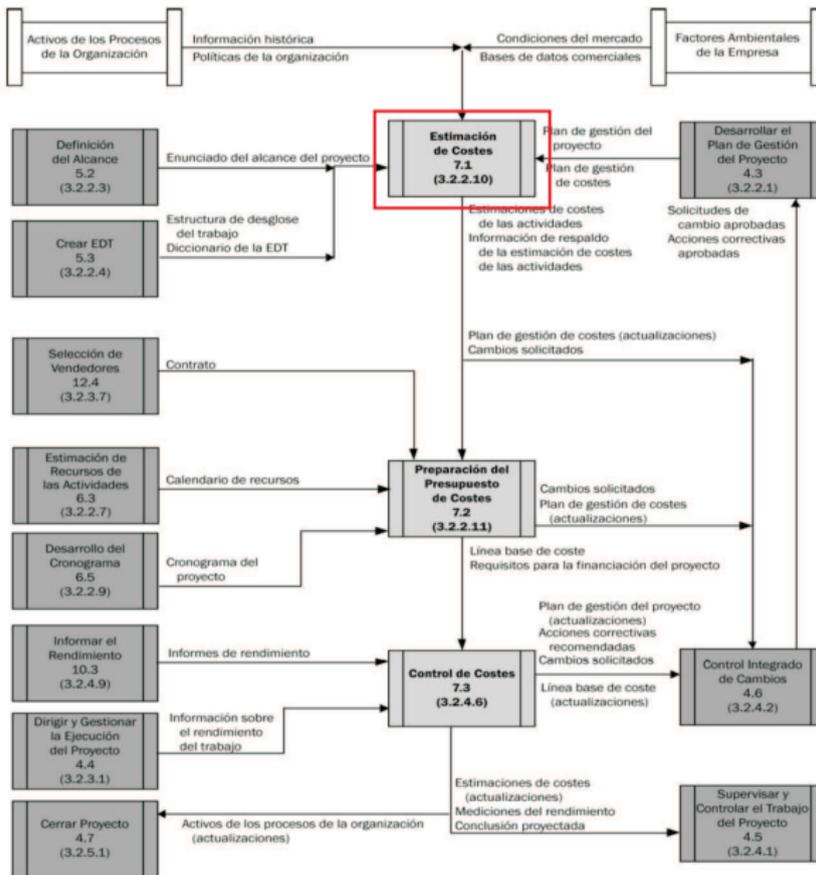
- 1 Estimar y presupuestar, con la mayor precisión posible, el coste real asociado a la construcción de un determinado sistema software.
- 2 Controlar y gestionar los ingresos y gastos durante el proceso de construcción de dicho sistema para que se cumplan las estimaciones anteriormente mencionadas.

Esquema General de la Gestión de Costes [6]



Índice

- 1 Introducción
- 2 Principales Retos de la Estimación Software
- 3 Métricas de Tamaño: Puntos de Función
- 4 Técnicas de Estimación de Esfuerzos
- 5 Estimación de Costes con Cocomo II
- 6 Control de Costes
- 7 Sumario y Bibliografía



Fases en la Estimación de Costes Software

Estimación de Tamaño ¿Cuánto (y cuán complejo) tengo que construir?

Estimación de Esfuerzo ¿Qué tiempo me lleva construir lo estimado? ¿Cuántos trabajadores necesito?

Cálculo de Costes Identificación de terceros recursos, cálculo de costes de recursos y elaboración del presupuesto.

Cálculo de Costes del Sobado Manual de la Anchoa

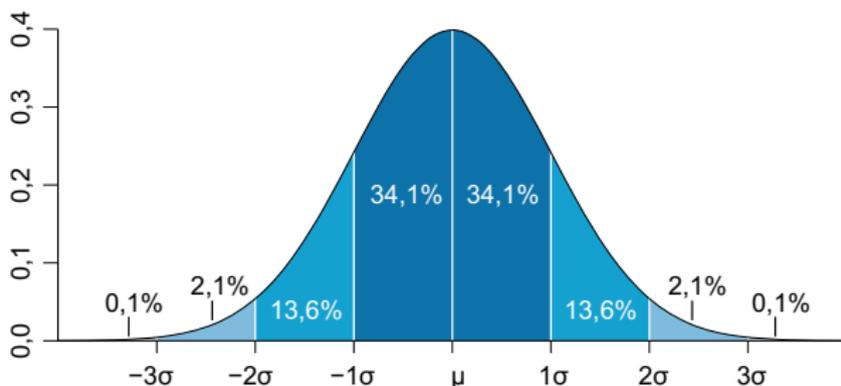
- 1 Se trata de calcular cuánto hay que cobrar por el sobado manual de 2000 anchoas, usando como máximo tres trabajadores.
- 2 Productividad media (trabajador) = 25 anchoas/hora.
- 3 Coste directo trabajador (€/hora) = 7€/hora.
- 4 Coste indirecto trabajador (€/hora) = 12€/hora.
- 5 Gastos de transporte y gestión = 200€.
- 6 Calculo esfuerzo disponible por hora, divido el total de anchoas y multiplico por el coste de cada hora.

Trabajadores	1	2	3
Coste (€)	1720	1720	1720
Tiempo (semanas)	2	1	0.67

Retos en la Gestión de Costes Software

- 1 Sistemas software con la misma funcionalidad pero diferentes atributos de calidad tienen diferente coste (ej. **encriptación, modularidad**).
- 2 ¿Cómo y en qué se mide la productividad de un arquitecto software, un diseñador software o un programador software?
- 3 Valores medios calculados con amplios márgenes de desviación **incluso en productividad por trabajador**.
- 4 ¿Cómo estimo el esfuerzo a realizar?
- 5 Volatilidad de los requisitos software.
- 6 Innovación constante: nuevo lenguajes de programación, procesos de desarrollo, herramientas de desarrollo software, dominios de aplicación.

Problemas de Altos Valores de Desviación



$z\sigma$	% dentro	% fuera	media fuera
1σ	68.2689492 %	31.7310508 %	1 / 31514871
2σ	95.4499736 %	4.5500264 %	1 / 21977894
3σ	99.7300204 %	0.2699796 %	1 / 370398
4σ	99.993666 %	0.006334 %	1 / 15788
5σ	99.9999426697 %	0.0000573303 %	1 / 1744278
6σ	99.9999998027 %	0.0000001973 %	1 / 506800000
7σ	99.999999997440 %	0.000000002560 %	1 / 390600000000

Fases en la Estimación de Costes Software

- 1 Estimación de Tamaño
- 2 Estimación de Esfuerzo
- 3 Cálculo de Costes

Índice

- 1 Introducción
- 2 Principales Retos de la Estimación Software
- 3 **Métricas de Tamaño: Puntos de Función**
- 4 Técnicas de Estimación de Costes
- 5 Estimación de Costes con Cocomo II
- 6 Control de Costes
- 7 Sumario y Bibliografía

Métricas de Estimación de Tamaño

Métricas basadas en Tamaño

Miden un producto en función de la cantidad de *algo físico* de un artefacto software (ej. líneas de código, número de atributos de una tabla).

Problemas de las líneas de código

- 1 ¿Qué es una línea de código? (ej. puedo escribir una aplicación en una sola línea muy larga)
- 2 No es lo mismo una línea LISP, que una Java, que una en ensamblador.
- 3 Dificultad de predecir cuantas LOC se van a usar para implementar una función.
- 4 Técnicas modernas de generación de código.

Métricas de Estimación de Tamaño

Métricas basadas en Función

Cantidad de funcionalidad útil producida (ej. casos de uso).

Ventajas de las Métricas basadas en Función

- 1 *Aplicables* desde fases tempranas de desarrollo.
- 2 Independientes del lenguaje, herramientas y/o metodologías.

Puntos de Función [1]

Punto de Función

Unidad de medida que expresa la cantidad de funcionalidad real que un usuario percibe en un sistema software.

Ventajas de los Puntos de Función

- 1 Basadas en la funcionalidad visible por el usuario.
- 2 *Repetible* (precisión del 10%).
- 3 Funcionan bien con técnicas de Ingeniería de Requisitos como casos de uso o escenarios.

Proceso de Cálculo de Puntos de Función

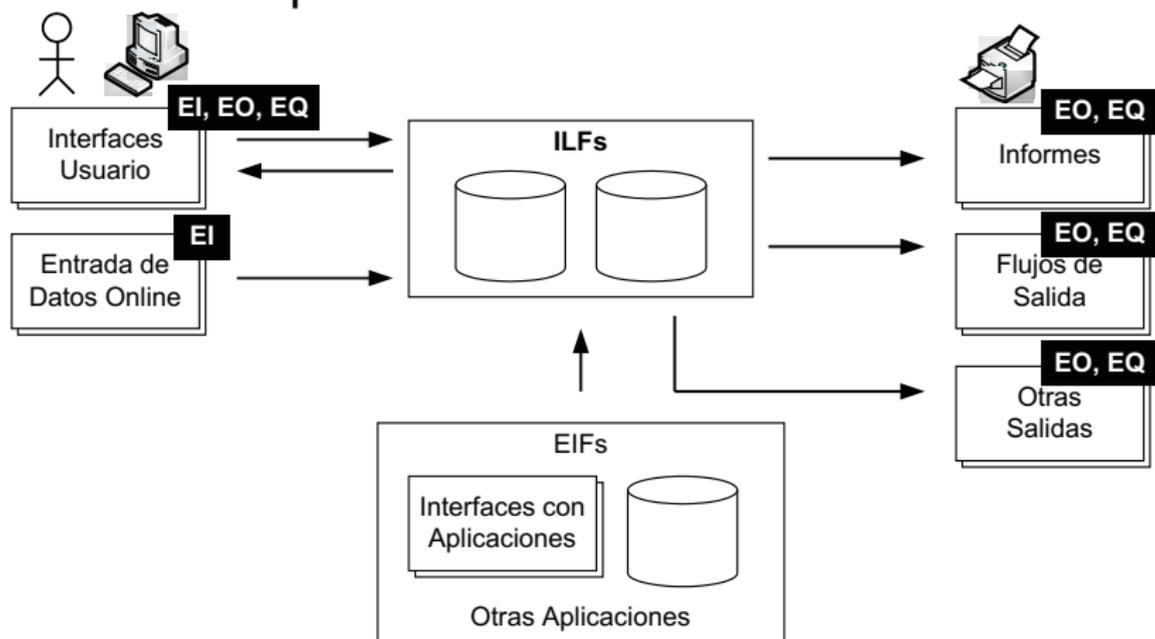
- 1 Determinar el tipo de cuenta.
- 2 Identificar límites de la aplicación.
- 3 Calcular Puntos de Función Sin Ajuste.
- 4 Calcular los Atributos de Ajuste.
- 5 Calcular el Valor Final Ajustado.

Elementos Identificados como Puntos de Función

- 1 Elementos de Datos:
 - 1 Ficheros lógicos internos (ILF).
 - 2 Ficheros lógicos externos (EIF).
- 2 Funciones *transaccionales*:
 - 1 Entradas externas (EI).
 - 2 Salidas externas (EO)
 - 3 Consultas externas (EQ)

Elementos Identificados como Puntos de Función

Aplicación Software Medida



Puntos de Función: Definiciones

Archivos Lógicos Internos (ILF)

Grupo de datos lógicamente relacionados, visibles al usuario, relativos a datos o información de control de la aplicación; y mantenidos **dentro del ámbito de la aplicación**.

Archivos Lógicos Externos (EIF)

Grupo de datos lógicamente relacionados, visibles al usuario, relativos a datos o información de control de la aplicación; y mantenidos **fuera del ámbito de la aplicación**.

Puntos de Función: Definiciones

Tipo de Datos Elemental (DET)

Campo o atributo único y no repetido, visible al usuario.

Tipo Registro Elemental (RET)

Grupo de datos lógicamente relacionados **dentro de un fichero lógico** externo o interno.

Tipo Fichero Referenciado (FTR)

Un archivo Lógico Interno o Externo.

- 1 FTR puede tener varios RET.
- 2 Un grupo de datos se considera RET si es opcional o puede existir independientemente de los otros.

Puntos de Función: Definiciones

Entradas Externas (EI)

Proceso elemental de la aplicación siendo medida que procesa datos o información de control procedentes de más allá del ámbito de la aplicación. Actualiza uno o más ILFs o modifica el comportamiento de la aplicación.

Puntos de Función: Definiciones

Salidas Externas (EO)

Proceso elemental de la aplicación siendo medida que genera datos que se envían más allá de la frontera de la aplicación. La generación de dichos datos precisa de un procesamiento más complejo que el de la simple recuperación de la información almacenada. **Dicho proceso elemental debe realizar algún cálculo, generar datos derivados, modificar algún ILF o el estado de la aplicación.**

Consultas Externas (EQ)

Proceso elemental de la aplicación siendo medida que envía datos o información de control más allá de la frontera de la aplicación. Dicho proceso elemental sólo muestra información tal como está contenida en el sistema, **sin realizar ningún tipo de procesamiento extra más allá de la simple recuperación de datos.**

Procesos Elementales y Granularidad de los Requisitos [5]

- Sky-level** Funciones muy generales del sistema (ej. realizar la declaración de la RENTA).
- Kite-level** Refinamiento de funciones generales (ej. recuperar información fiscal).
- Sea-level** Interacción simple con el sistema involucrando un único actor (ej. verificar certificado digital).
- Mud-level** Detalles de implementación (ej. comprobar integridad MD5).

Proceso elemental

Menor unidad de actividad que tiene sentido para el usuario (\simeq caso de uso *sea-level*)

Tablas de Complejidades

Consultas Externas (EQ)

FTRs / DETs	1-5	6-19	20-*
0-1	LOW	LOW	NORMAL
2-3	LOW	NORMAL	HIGH
4-*	NORMAL	HIGH	HIGH

Complexity	LOW	NORMAL	HIGH
Points	3	4	6

Tablas de Complejidades

Salidas Externas (EO)

FTRs / DETs	1-5	6-19	20-*
0-1	LOW	LOW	NORMAL
2-3	LOW	NORMAL	HIGH
4-*	NORMAL	HIGH	HIGH

Complexity	LOW	NORMAL	HIGH
Points	4	5	7

Tablas de Complejidades

Entradas Externas (EI)

FTRs / DETs	1-4	5-15	16-*
0-1	LOW	LOW	NORMAL
2	LOW	NORMAL	HIGH
3-*	NORMAL	HIGH	HIGH

Complexity	LOW	NORMAL	HIGH
Points	3	4	6

Tablas de Complejidades

Ficheros Lógicos Externos (EIF)

RETs / DETs	1-19	20-50	51-*
1	LOW	LOW	NORMAL
2-5	LOW	NORMAL	HIGH
6-*	NORMAL	HIGH	HIGH

Complexity	LOW	NORMAL	HIGH
Points	5	7	10

Tablas de Complejidades

Ficheros Lógicos Internos (ILF)

RETs / DETs	1-4	5-15	16-*
1	LOW	LOW	NORMAL
2-5	LOW	NORMAL	HIGH
6-*	NORMAL	HIGH	HIGH

Complexity	LOW	NORMAL	HIGH
Points	7	10	15

Proceso de Cálculo de Puntos de Función

- 1 Determinar el tipo de cuenta.
- 2 Identificar límites de la aplicación.
- 3 Calcular Puntos de Función Sin Ajuste.
- 4 **Calcular los Atributos de Ajuste.**
- 5 Calcular el Valor Final Ajustado.

Factores de Ajuste

- 1 Comunicaciones de datos.
- 2 Procesamiento distribuido.
- 3 Rendimiento.
- 4 Uso intensivo.
- 5 Tasas de transacciones rápidas.
- 6 Entrada de datos en línea.
- 7 Eficiencia desde el punto de vista del usuario.
- 8 Actualizaciones.
- 9 Procesamiento complejo.
- 10 Reutilización.
- 11 Facilidad de instalación.
- 12 Facilidad operacional.
- 13 Multiplataforma.
- 14 Facilidad de Cambio.

Proceso de Cálculo de Puntos de Función

- 1 Determinar el tipo de cuenta.
- 2 Identificar límites de la aplicación.
- 3 Calcular Puntos de Función Sin Ajuste.
- 4 Calcular los Atributos de Ajuste.
- 5 **Calcular el Valor Final Ajustado.**

Puntos Función Ajustados

- 1 Se valora cada factor entre 0 (mínimo) y 5 (máximo)
- 2 Se suman los 14 factores y se obtiene el TDI (Total Degree of Influence)

Fórmula Puntos Función

$$FP = WAFP * (0,65 + 0,01 * TDI)$$

Puntos de Función y Líneas de Código

Lenguaje (o entorno de programación)	LDC/PF
4GL	40
Ada 83	71
Ada 95	49
APL	32
BASIC - compilado	91
BASIC - interpretado	128
BASIC ANSI/Quick/Turbo	64
C	128
C++	29
Clipper	19
Cobol ANSI 85	91
Delphi 1	29
Ensamblador	320
Ensamblador (Macro)	213
Forth	64
Fortran 77	105
FoxPro 2.5	34
Generador de Informes	80
Hoja de Cálculo	6
Java	53
Modula 2	80
Oracle	40
Oracle 2000	23
Paradox	36
Pascal	91
Pascal Turbo 5	49
Power Builder	16
Prolog	64
Visual Basic 3	32
Visual C++	34
Visual Cobol	20

Índice

- 1 Introducción
- 2 Principales Retos de la Estimación Software
- 3 Métricas de Tamaño: Puntos de Función
- 4 **Técnicas de Estimación de Esfuerzos**
- 5 Estimación de Costes con Cocomo II
- 6 Control de Costes
- 7 Sumario y Bibliografía

Técnicas Descendentes vs Técnicas Ascendentes

1 Técnicas descendentes (Top-down):

- ▶ Se evalúa el coste de un sistema desde sus funcionalidades globales.
- ▶ Permite estimar mejor aspectos globales como integración de subsistemas o coordinación entre equipos.
- ▶ Se puede aplicar con pocos detalles, en fases tempranas.
- ▶ Puede subestimar costes asociados a detalles internos de uno o más subsistemas.
- ▶ No resultan adecuadas para tomar decisiones sobre componentes individuales.

2 Técnicas ascendentes (Bottom-up)

- ▶ Se estima el coste de cada uno de los componentes de un sistema software y luego se calcula el total.
- ▶ Permite analizar mejor detalles individuales de cada componente.
- ▶ Son en general más precisos, porque la media de los errores en las estimaciones individuales tiende a 0.
- ▶ Puede subestimar costes asociados a actividades globales.
- ▶ Precisa disponer de una estructura más o menos detallada del sistema y suelen ser más costosos

Técnicas de Estimación de Esfuerzos

- 1 Juicio Experto
- 2 Estimación por Analogía
- 3 Modelos Paramétricos o Algorítmicos
- 4 Ajuste del Producto al Precio.
- 5 Otras técnicas

Estimación por Juicio Experto

- 1 Diversos expertos elaboran estimaciones individuales.
- 2 Estas estimaciones se comparan y analizan.
- 3 Se repite el proceso hasta alcanzar un resultado satisfactorio.

Ventajas e Inconvenientes

- (+) Los expertos valoran factores subjetivos (ej. estado anímico del equipo) que no se computan mediante otros métodos.
- (-) Precisa de expertos (personas), difícilmente reemplazables.
- (-) No reproducible, resultados dispares en diferentes aplicaciones.

Estimación por Juicio Experto: Técnica Delphi

- 1 Se forma un grupo de expertos con un coordinador designado.
- 2 El coordinador proporciona una descripción del sistema y formularios para la estimación.
- 3 Se discuten en grupo cuestiones generales sobre el sistema.
- 4 Cada experto hace su propia estimación de forma anónima.
- 5 El coordinador procesa las estimaciones individuales.
- 6 En caso de discrepancias no triviales, se discuten en grupo de nuevo.
- 7 Se vuelve al punto 4 hasta alcanzar el deseado grado de convergencia.

Estimación por Analogía

- 1 Se dispone de datos fiables sobre proyectos realizados en el pasado.
- 2 Se identifican atributos del proyecto.
- 3 Se buscan proyectos con similares atributos.
- 4 Se ajustan las estimaciones de acuerdo a las características propias.
- 5 Se requiere del uso de expertos para los 3 puntos anteriores.

Ventajas e Inconvenientes

- (+) Basado en experiencias pasadas reales que incluyen factores subjetivos (ej. riesgos).
- (-) Precisa de la existencia de proyectos pasados **similares**.
- (-) ¿Cómo se mide el grado de *similitud* entre proyectos?
- (-) Lo que era válido en el pasado puede no ser válido ahora.

Matriz de Costes de Wolverton (\$/LOC) [8]

Tipo de Software	AF	AM	AD	NF	NM	ND
Control	21	27	30	33	40	49
Entrada/Salida	17	24	27	28	35	43
Pres/post procesador	16	23	26	28	34	42
Algoritmo	15	20	22	25	30	35
Gestión de Datos	24	31	35	37	46	57
Crítico en Tiempo	75	75	75	75	75	75

A: Antiguo N: Nuevo

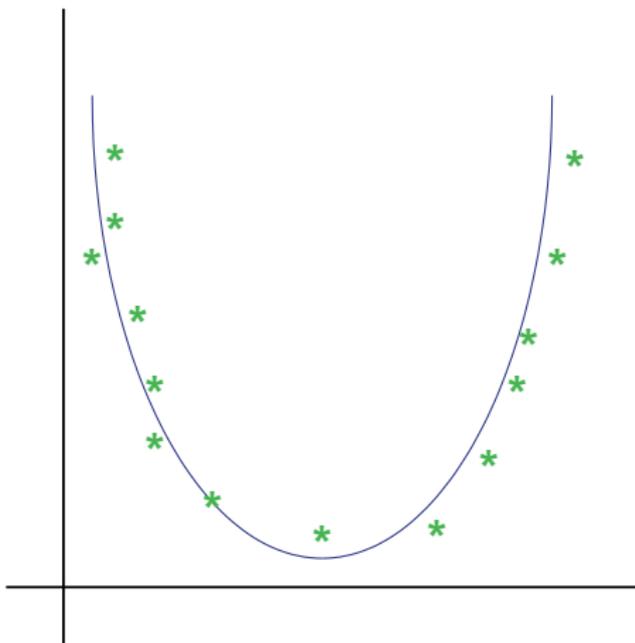
F: Fácil M: Moderado D: Difícil

Tabla de Estimación de Costes por Analogía

Tamaño del Programa (LDC)	Software de Sistemas		Software de Gestión		Software "a medida"	
	Duración (meses)	Esfuerzo (personas-mes)	Duración (meses)	Esfuerzo (personas-mes)	Duración (meses)	Esfuerzo (personas-mes)
10.000	10	48	6	9	7	15
15.000	12	76	7	15	8	24
20.000	14	110	8	21	9	34
25.000	15	140	9	27	10	44
30.000	16	185	9	37	11	59
35.000	17	220	10	44	12	71
40.000	18	270	10	54	13	88
45.000	19	310	11	61	13	100
50.000	20	360	11	71	14	115
60.000	21	440	12	88	15	145
70.000	23	540	13	105	16	175
80.000	24	630	14	125	17	210
90.000	25	730	15	140	17	240
100.000	26	820	15	160	18	270
120.000	28	1.000	16	200	20	335
140.000	30	1.200	17	240	21	400
160.000	32	1.400	18	280	22	470
180.000	34	1.600	19	330	23	540
200.000	35	1.900	20	370	24	610
250.000	38	2.400	22	480	26	800
300.000	41	3.000	24	600	29	1.000
400.000	47	4.200	27	840	32	1.400
500.000	51	5.500	29	1.100	35	1.800

Tabla de Estimación de esfuerzo y duración de proyectos software de complejidad media

Estimación Algorítmica o Paramétrica



Estimación Algorítmica o Paramétrica

- 1 Basados en construir funciones predictoras que estimen el coste de un proyecto.
- 2 Estas funciones predictoras se calculan mediante ajustes, en función de diversos parámetros, sobre datos de proyectos anteriores.

Ventajas e Inconvenientes

- (+) Cálculos reproducibles y que permiten refinamiento.
- (+) El problema se reduce a estimar el valor de ciertos parámetros (ej. tamaño, complejidad).
- (+) Basado en sólidas experiencias pasadas estadísticamente significativas.
- (-) Lo que era válido en el pasado puede no ser válido ahora.
- (-) Dificultad de adaptación a circunstancias o eventos no excepcionales.

Ajuste del Coste al Precio Esperado (Pricing to Win)

- 1 El cliente tiene un presupuesto fijo.
- 2 El cliente no puede proporcionar una descripción clara y detallada de lo que quiere.
- 3 Se establece como coste del producto el precio que el cliente está dispuesto a pagar.
- 4 A continuación, se identifican los requisitos del sistema software.
- 5 Se implementan los requisitos de forma que el proyecto se ajuste al coste.
- 6 Se pueden dejar requisitos sin implementar.

Otras Técnicas

- 1 Redes neuronales.
- 2 Técnicas de aprendizaje computacional.
- 3 Lógica difusa.
- 4 Sistemas expertos.

Cálculo de Costes

- Las técnicas presentadas calculan recursos y duraciones necesarios.
- Falta multiplicar cantidad de recurso por coste de cada recurso.
- ¿Cuánto cuesta un recurso (ej. unos auriculares bluetooth, un programador senior)?.
- En muchas ocasiones deberemos estimar también el coste de los recursos.
- Se usarán catálogos, tablas estandarizadas de precios, etc.

Determinación del Precio Final

- 1 Precio final = coste total + beneficio.
- 2 El coste total incluye:
 - 1 Gastos indirectos (ej. infraestructura, plan social).
 - 2 Reserva para contingencias (ej. devaluaciones de divisas).
- 3 El beneficio se ajusta en función de (entre otros):
 - ▶ Oportunidad de mercado (nuevo segmento).
 - ▶ Incertidumbre en las estimaciones.
 - ▶ Términos contractuales (ej. obliga a permanencia).
 - ▶ Volatilidad de los requisitos (ej. software gestión universitaria).
 - ▶ Estado financiero de la empresa.

Índice

- 1 Introducción
- 2 Principales Retos de la Estimación Software
- 3 Fundamentos de Estimación de Costes Software
- 4 Puntos de Función
- 5 Estimación de Costes con Cocomo II
- 6 Control de Costes
- 7 Sumario y Bibliografía

COCOMO II

- **Constructive Cost Model**
- Modelo de estimación algorítmico desarrollado por Dr. Barry Boehm [2].
- La primera versión tenía problemas con software empresarial, orientado a objetos, con modelos en espiral o basado en componentes.
- COCOMO II [4, 3] se publica en 1995 como reemplazo a COCOMO, solventando los problemas anteriores.

Por qué COCOMO II

- Ampliamente conocido y de dominio público.
- Soportado por herramientas tanto comerciales como libres.
- Usado en múltiples entornos industriales.
- Goza de buena reputación y se mantiene vigente.
- Abundante literatura.

Objetivos COCOMO II

- Proporcionar modelo de estimación de costes ajustado a las prácticas de los 90's y del 2000.
- Proporcionar técnicas analíticas para evaluar los efectos de ciertas decisiones sobre planificación y costes.
- Crear una bases de datos y herramientas que permitan mejorar el modelo.

Modelos COCOMO II

Application Composition Model Estimaciones de esfuerzo de desarrollo de prototipos. Se aplica en las primeras iteraciones de un ciclo de vida en espiral.

Early Design Model Estimaciones imprecisas con pocos parámetros. Se aplica en las siguientes iteraciones. Usa puntos de función.

Post-Architectural Model Una vez establecida una arquitectura, hacemos estimaciones más precisas usando más parámetros. Usa LOC ó FP. Sirve para mantenimiento.

Estimación de Esfuerzo

Personas Mes Nominales

$$PM_{nominal} = A \times Size^B (A = 2,5)$$

- 1 $B < 1$. Los esfuerzos de desarrollo mejoran cuando escalan. Si se dobla el tamaño, el esfuerzo es menor del doble.
- 2 $B = 1$. Los proyectos están balanceados. Los aumentos son proporcionales.
- 3 $B > 1$. Los esfuerzos de desarrollo empeoran cuando escalan. Si se dobla el tamaño, el esfuerzo es menor del doble.

Factor de Escala

$$B = 1,01 + 0,01 \times \sum W_i$$

Cálculo de Esfuerzo

$$PM = PM_{nominal} * \prod EM_i$$

EM : Effort Multipliers W : Factores de Escala

Factores de Escala

PREC	Experiencia en proyectos similares.
FLEX	Flexibilidad de desarrollo.
RESL	Gestión de riesgos arquitectónicos.
TEAM	Cohesión del equipo de trabajo (incluye clientes).
PMAT	Madurez del proceso (CMM).

Se puntúan entre 0 y 5. 0 representa Muy Alto y 5 Muy Bajo.

Ajustes de Tamaño

Breakage

$$BRAK = \frac{DiscardedCode}{TotalCode} 100$$

Reutilización

$$AAF = 0,4 \times DM + 0,3 \times CM + 0,3 \times IM$$

$$ESLOC = \frac{ASLOC(AA+AAF(1+0,02 \times SU \times UNFM))}{100} \quad \text{si } AAF \leq 0,5$$

$$ESLOC = \frac{ASLOC(AA+AAF+SU \times UNFM)}{100} \quad \text{si } AAF > 0,5$$

AAF: Adaptation Adjustment Factor DM: Percent Design Modified CM: Percent Code Modified

IM: Percent of Integration Required for Modified Software

SU: Software Understanding Increment AA: Assesment and Assimiliation

UNFM: Unfamiliarity with the Software

Modelo para Diseño Temprano (EDM)

- 1 Usa puntos de función sin ajustar.
- 2 Los puntos de función se han de convertir a KSLOC
- 3 Usa 7 estimadores de esfuerzo (ver tabla).

RCPX	Fiabilidad y complejidad del sistema
RUSE	Reutilización requerida
PDIF	Complejidad de la plataforma
PERS	Capacidad del personal
PREX	Experiencia del personal
FCIL	Facilidades
SCED	Ajuste al calendario

Se evalúan entre 1 y 6, donde 1 es Muy Bajo y 6 Muy Alto.

Tiempo Óptimo de Desarrollo

Duración del Proyecto

$$TDEV = 3 \times PM^{0,33+0,2 \times (B-1,01)} \times \frac{SCED\%}{100}$$

Crítica a COCOMO II

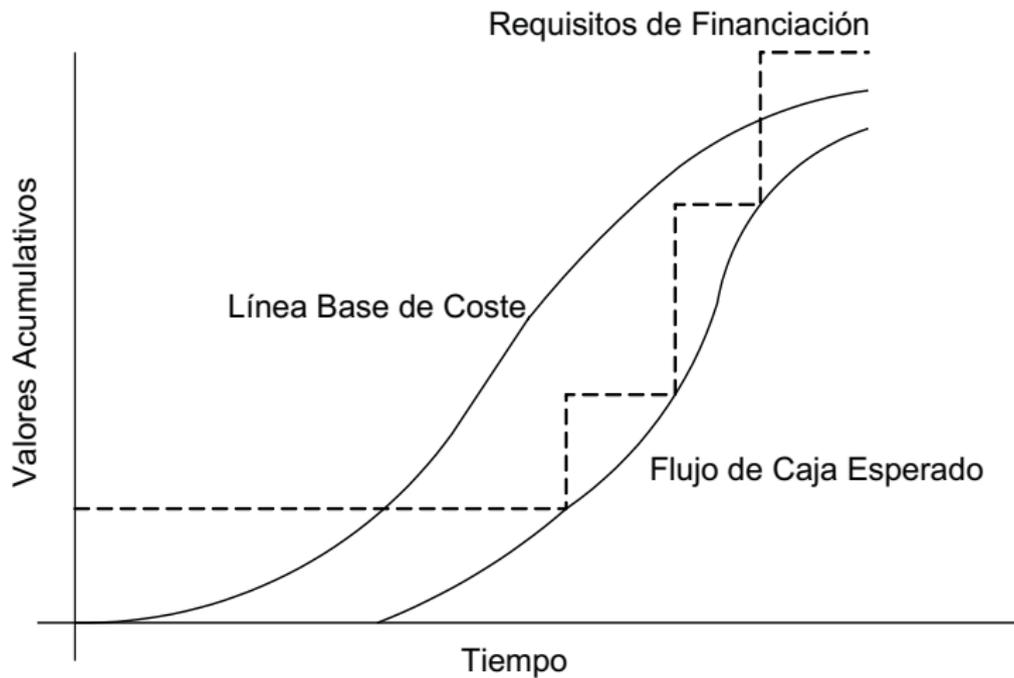
“This formulae proposed by the developers of the COCOMO II model reflects their experience and data, but it is an extremely complex model to understand and use. There are many attributes and considerable scope for uncertainty in estimating their values. In principle, each user of the model should calibrate the model and the attribute values according to its own historical project data, as this will reflect local circumstances that affect the model.”

“In practice, however, few organisations have collected enough data from past projects in a form that supports model calibration. Practical use of COCOMO II therefore has to start with the published values for the model parameters, and it is impossible for a user to know how closely these relate to their own situation. This means that the practical use of the COCOMO model is limited. Very large organisations may have the resources to employ a cost-modelling expert to adapt and use the COCOMO II models. However, for the majority of companies, the cost of calibrating and learning to use an algorithmic model such as the COCOMO model is so high that they are unlikely to introduce this approach.” [7]

Índice

- 1 Introducción
- 2 Principales Retos de la Estimación Software
- 3 Fundamentos de Estimación de Costes Software
- 4 Puntos de Función
- 5 Estimación de Costes con Cocomo II
- 6 **Control de Costes**
- 7 Sumario y Bibliografía

Línea Base de Coste



Actividades del Control de Costes

- 1 Influir sobre los factores que producen cambios en la línea base coste.
- 2 Asegurarse de que los cambios solicitados sean acordados.
- 3 Gestionar los cambios reales cuando y a medida que se produzcan.
- 4 Asegurar que los posibles sobrecostes no excedan la financiación autorizada periódica y total para el proyecto.
- 5 Realizar el seguimiento del rendimiento del coste para detectar y entender las variaciones con respecto a la línea base de coste.
- 6 Registrar todos los cambios pertinentes con precisión en la línea base de coste.
- 7 Evitar que se incluyan cambios incorrectos, inadecuados o no aprobados en el coste o en el uso de recursos informados.
- 8 Informar los cambios aprobados a los interesados pertinentes.
- 9 Actuar para mantener los sobrecostes esperados dentro de límites aceptables.
- 10 Se suelen emplear técnicas como la de *valor conseguido*.

Índice

- 1 Introducción
- 2 Principales Retos de la Estimación Software
- 3 Fundamentos de Estimación de Costes Software
- 4 Puntos de Función
- 5 Estimación de Costes con Cocomo II
- 6 Control de Costes
- 7 Sumario y Bibliografía

¿Qué tengo que saber de todo esto?

- 1 Retos de la Gestión de Costes Software.
- 2 **Cálculo de Puntos de Función.**
- 3 Técnicas de estimación de esfuerzos.
- 4 Método de Estimación de Costes COCOMO II

Referencias I



Allan J. Albrecht and John E. Gaffney.

Software Function, Source Lines of Code, and Development Effort Prediction: A Software Science Validation.

Transactions on Software Engineering, 9(6):639–648, November 1983.



Barry W. Boehm.

Software Engineering Economics.

Prentice Hall, November 1981.



Barry W. Boehm, Chris Abts, A. Winsor Brown, Sunita Chulani, Bradford K. Clark, Ellis Horowitz, Ray Madachy, Donald J. Reifer, and Bert Steece.

Software Cost Estimation with COCOMO II.

Prentice Hall, August 2009.

Referencias II

-  Barry W. Boehm, Bradford Clark, Ellis Horowitz, J. Christopher Westland, Raymond J. Madachy, and Richard W. Selby.
Cost Models for Future Software Life Cycle Processes: COCOMO 2.0.
Annals of Software Engineering, 1:57–94, 1995.
-  Alistair Cockburn.
Writing Effective Use Cases.
Addison-Wesley, October 2000.
-  Project Management Institute.
A Guide to the Project Management Body of Knowledge (PMBOK Guide).
Technical Report ANSI/PMI 99-001-2004, American National Standard, 2004.

Referencias III

 Ian Sommerville.
Software Engineering.
Addison Wesley, 7 edition, May 2004.

 Ray W. Wolverton.
The Cost of Developing Large-Scale Software.
IEEE Transactions on Computer, 23(6):615–636, June 1974.