



# Tema 07 – Estimación del coste de proyectos software

## Ingeniería del Software

Rubén Fuentes Fernández  
Dep. Ingeniería del Software e Inteligencia Artificial  
Facultad de Informática  
Universidad Complutense Madrid

Trabajando con Antonio Navarro, Juan Pavón y Pablo Gervás





# Contenidos

- Introducción
  - Problemática
- Aspectos de la solución
  - Valoración de costes
  - Estimación del proyecto software
    - Técnicas de estimación
    - Modelos paramétricos de estimación





# Coste del desarrollo

- Costes de hardware y software a utilizar
- Costes de esfuerzo
  - Es el factor dominante en la mayoría de los proyectos
    - Sueldos de los ingenieros involucrados
    - Seguridad social y seguro
- Costes de viajes y formación
- Costes de infraestructura de la organización
  - Costes del edificio, calefacción y luz
  - Costes de la red y de comunicaciones
  - Costes de recursos compartidos
    - Ej. biblioteca o cantina





# La decisión de desarrollar

- Muchas veces es más rentable adquirir el software que desarrollarlo.
- Hay diversas opciones de adquisición:
  - Comprar ya desarrollado bajo licencia.
  - Adquirir componentes ya experimentados o parcialmente experimentados.
  - Construir de forma personalizada por una empresa externa para cumplir las especificaciones del comprador.
    - Subcontratar (*outsourcing*)



# Costes y precios

- No hay una relación simple entre el coste de desarrollo y el precio que se pide al cliente.
  - En el precio influyen consideraciones económicas, políticas y de negocio.

Factor	Description
Market opportunity	A development organization may quote a low price because it wishes to move into a new segment of the software market. Accepting a low profit on one project may give the opportunity of more profit later. The experience gained may allow new products to be developed.
Cost estimate uncertainty	If an organization is unsure of its cost estimate, it may increase its price by some contingency over and above its normal profit.
Contractual terms	A customer may be willing to allow the developer to retain ownership of the source code and reuse it in other projects. The price charged may then be less than if the software source code is handed over the customer.
Requirements volatility	If the requirements are likely to change, an organization may lower its price to win a contract. After the contract awarded, high prices may be charged for changes to the requirements.
Financial health	Developers in financial difficulty may lower their price to gain a contract. It is better to make a small profit of break even than go out of business.





# Estimación del esfuerzo

- Estimar el coste del software es vital.
- Las estimaciones nunca podrán ser exactas.
  - Los requisitos iniciales no están totalmente delimitados.
  - Puede que necesitemos utilizar tecnologías nuevas.
  - Las personas involucradas en el proyecto pueden tener distintos grados de experiencia.
  - ....
- Cuanto mejor se estime, más rentable será el proyecto.





# Esfuerzo de un proyecto

- El *esfuerzo de un proyecto* es la cantidad de trabajo a dedicar para desarrollar el proyecto.
  - Depende del número de personas y del tiempo que estén trabajando.
- Suele medirse en Personas-Mes (PM).
  - Una medida de la rapidez con que cada ingeniero produce software y la documentación asociada.
  - No es orientada a la calidad, aunque se tiene en cuenta si se aplica gestión de calidad.
  - Se mide funcionalidad útil por unidad de tiempo.
- Las técnicas de estimación persiguen determinar el esfuerzo para realizar un proyecto.
- El esfuerzo se liga a unas tarifas para establecer el coste de desarrollo del proyecto.





# Esfuerzo estándar

- Con vistas a los cálculos basados en Persona-Mes (PM) se considera que:
  - Un mes tiene 22 días laborables en media.
  - Una jornada completa de una persona son 8 horas de trabajo.
    - La media jornada son 4 horas.







# Técnicas de estimación

- Malas
  - Retrasar la estimación lo máximo posible
    - Cuanto más la retrasemos, más precisa será.
  - Ley de Parkinson
    - El trabajo se extiende para rellenar el tiempo disponible.
  - Precio para ganar
    - El coste se estima en todo el dinero que el cliente puede gastar en el proyecto.
- Apropriadas
  - Estimación por analogía
    - Utilizar el coste de proyectos similares ya terminados.
  - Técnicas de descomposición
    - Estiman el coste descomponiendo el producto y/o el proceso.
  - Modelos empíricos
    - Modelos de regresión que relacionan esfuerzo con tamaño o funcionalidad.





# DESCOMPOSICIÓN BASADA EN EL PRODUCTO





# Descomposición basada en el producto

- Las técnicas de descomposición basadas en el problema o producto se basan en la descomposición del sistema.
  - Módulos que contienen
  - Funciones (métodos)
- La descomposición del sistema facilita estimar su tamaño.
- Existen distintas formas de medir el tamaño del software.
  - Líneas de Código (LDC)
  - Puntos de Función (PF)
  - *Punto Objeto (PO)*
- En general, cuanto más grande sea la partición, más precisión tendrá la estimación.





## Ejemplo: CAD – ámbito (1/7)

- Supongamos que para un programa CAD (*Computer-Aided Design*) tenemos identificado el siguiente ámbito (simplificando mucho):
  - El programa aceptará datos geométricos bi y tridimensionales.
  - Sobre los dibujos se podrá especificar una serie de transformaciones.
  - El sistema de archivo incluirá intercambio DXF (*Drawing eXchange Format*).
- El sistema puede disponer de varios periféricos, en particular, ratón, digitalizadora, uno o dos monitores y plotter.
- Se utilizará una interfaz basada en menús y ventanas.





## Ejemplo: CAD – módulos (2/7)

- Supuesto un refinamiento (muy simplificado) podemos obtener los siguientes módulos:
  - Módulo de dibujo (MD)
  - Módulo de transformaciones (MT)
  - Módulo de archivos (MA)
  - Módulo de periféricos (MP)
  - Interfaz (I)



## Ejemplo: CAD – LDC (3/7)

Luego veremos porque hacer varias estimaciones. Por ahora, usamos el valor esperado.

Función	V_optimista	V_probable	V_pesimista	V_esperado
MD	8500	10500	13800	10717
MT	11000	15000	17200	14700
MA	6200	7700	8500	7583
MP	4000	5900	7400	5833
I	2700	3500	5000	3617
Total				42450

- Una revisión de datos históricos indica que la productividad media de la organización para este tipo de sistemas es de 650 LDC/PM, y con una tarifa laboral de 7800 €/PM:
  - esfuerzo =  $42450 \text{ LDC} / 650 \text{ LDC/PM} = 65 \text{ PM}$
  - coste =  $65 \text{ PM} * 7800 \text{ €/PM} = 507000 \text{ €}$





# Puntos de función

- El tamaño del software se puede estimar basándose en su funcionalidad.
  - Utilizando los PF o los PO.
- Se parte de la descomposición del problema pero sin llegar al nivel de detalle requerido para las LDC.
- Se estiman:
  - Elementos de funcionalidad
  - Factores generales de ajuste de complejidad
- Con estos valores se calcula un valor esperado en PF y se estima el tamaño aplicando los datos históricos.
- Muy orientado a sistemas de gestión.
  - Centrados en la gestión de bases de datos



# Cálculo de valores

- Para cada elemento de función:
  - Se obtiene un valor numérico
    - ¿Cuántos hay?
  - Se clasifica según su complejidad
    - Simple, media, alta
- Hay que aplicar una corrección adicional determinada por la complejidad general de la aplicación.
  - Factores de complejidad
- Los PF ajustados resultantes se obtienen como:

$$F = \left( \sum_{\substack{\text{elemento de función,} \\ \text{complejidad}}} \left( PF_{\substack{\text{elemento de función,} \\ \text{complejidad}}} * P_{\substack{\text{eso} \\ \text{elemento de función,} \\ \text{complejidad}}} \right) \right) * \left( 0,65 + 0,01 * \sum_{i=1..14} FC_i \right)$$





# Elementos de función

- Entradas (del usuario o sistema externo)
- Salidas (al usuario o sistema externo)
- Consultas del usuario (o interacciones)
- Archivos (o ficheros) lógicos usados por el sistema
- Interfaces externas (o ficheros de sistemas externos)

$$F = \left( \sum_{\substack{\text{elemento de función,} \\ \text{complejidad}}} \left( PF_{\substack{\text{elemento de función,} \\ \text{complejidad}}} * P_{\substack{\text{eso elemento de función,} \\ \text{complejidad}}} \right) \right) * \left( 0,65 + 0,01 * \sum_{i=1..14} FC_i \right)$$

Estos son la  
cuenta inicial



## Ejemplo: clasificación de las entradas

Dificultad de las entradas – Ficheros accedidos	Número de campos o atributos de la entrada		
	1-4	5-15	> 15
0-1	Baja	Baja	Media
2	Baja	Media	Alta
> 2	Media	Alta	Alta

Elementos de información que se manejan

$$F = \left( \sum_{\text{elemento de función,}} \left( P F_{\text{elemento de función,}} * P_{\text{elemento de función,}} \right) \right)$$

*complejidad* *complejidad*

$$* \left( 0,65 + 0,01 * \sum_{i=1..14} FC_i \right)$$

Estas son las  
complejidades



# Puntos de función sin ajustar

Puntos de función	Complejidad						Total
	Simple		Media		Compleja		
	#N	Peso	#N	Peso	#N	Peso	
Entradas		3		4		6	
Salidas		4		5		7	
Consultas del usuario		3		4		6	
Ficheros lógicos		7		10		15	
Interfaces externas		5		7		10	
Total de puntos de función sin ajustar							

$$F = \left( \sum_{\substack{\text{elemento de función,} \\ \text{complejidad}}} (PF_{\text{elemento de función, complejidad}} * Peso_{\text{elemento de función, complejidad}}) \right) * \left( 0,65 + 0,01 * \sum_{t=1..14} FC_t \right)$$

PF ajustados

PF sin ajustar

Factores de complejidad

# Factores de complejidad

Id.	Factor de complejidad	Valor (0..5)
1	Comunicación de datos	
2	Proceso distribuido	
3	Objetivos de rendimiento	
4	Integración de la aplicación	
5	Tasa de transacciones	
6	Entrada de datos interactiva	
7	Eficiencia para el usuario final	
8	Actualizaciones interactivas	
9	Lógica de proceso interna compleja	
10	Reusabilidad del código	
11	Conversión e instalación	
12	Facilidad de operación	
13	Instalaciones múltiples	
14	Facilidad de cambios	
<b>Factor de complejidad total</b>		<b><math>\Sigma</math></b>

De 0 (no es importante o no es aplicable) a 5 (absolutamente esencial)

Cada punto afecta en +/- 2,5% en los PF finales.





# Estimación del esfuerzo requerido

Proyecto	PFA	Lenguaje	Esfuerzo en horas	Horas / PFA
A	200	COBOL	5017	25
B	150	PASCAL	2569	17
C	375	4GL	3011	8
D	500	PASCAL	9479	19
E	425	4GL	3342	8
F	800	PASCAL	13349	17
G	180	PASCAL	2800	16
H	325	4GL	2541	8
I	225	PASCAL	4528	20
J	470	COBOL	13218	28

Lenguaje	Horas / PFA	LDC / PFA
Ensamblador	20-30	320
COBOL	10-20	100
4GL	5-10	40

Ejemplo





## Ejemplo: CAD – PF (4/7)

Parám. medición	V_optimista	V_probable	V_pesimista	V_esperado	Peso medio	PF ajustado
Entradas	22	27	30	27	4	108
Salidas	15	20	29	21	5	105
Peticiones	17	25	28	23	4	92
Archivos	4	4	5	4	10	40
Interfaces externas	2	2	3	2	7	14
Total						359

- F1: 4; F2: 2; F3: 0; F4: 4; F5: 4; F6: 5; F7: 4; F8: 3; F9: 5; F10: 5; F11: 4; F12:3; F13: 5; F14: 5

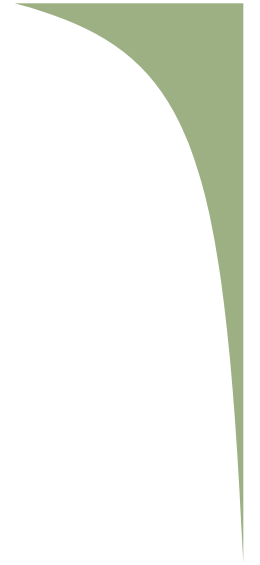


## Ejemplo: CAD – PF (5/7)

$$F = \left( \sum_{\substack{\text{elemento de función,} \\ \text{complejidad}}} \left( PF_{\substack{\text{elemento de función,} \\ \text{complejidad}}} * Peso_{\substack{\text{elemento de función,} \\ \text{complejidad}}} \right) \right) * \left( 0,65 + 0,01 * \sum_{i=1..14} FC_i \right) = 359 * (0,65 + 0,01 * 53) = 424 PF$$

- Una revisión de datos históricos indica que la productividad media de la organización para este tipo de sistemas es de 6 PF/PM, y una tarifa laboral de 7800 €/PM.
  - esfuerzo = 424 PF / 6 PF/PM = 71 PM
  - coste = 71 PM \* 7800 €/PM = 553800 €





# DESCOMPOSICIÓN BASADA EN EL PROCESO







# Descomposición basada en el proceso

- La técnica más común para estimar un proyecto es basar la estimación en el proceso que se va a utilizar.
- Se realiza en varios pasos:
  1. Descomposición del problema en grandes bloques funcionales.
  2. Utilizando el proceso, identificar un conjunto pequeño de actividades de trabajo o tareas de trabajo.
    - Constituyen la Estructura de Descomposición del Trabajo (EDT)
    - En inglés *Work Breakdown Structure* (WBS).
  3. Se estima el esfuerzo requerido para llevar a cabo cada tarea.





## Ejemplo: CAD – descomposición basada en el proceso (6/7)

AE	Comu. cliente	Plan.	Análisis riesgos	Ingeniería		Construcción y adaptación		Eval. cliente	Esti. total
Acción				Análisis.	Diseño	Codif.	Prueba		
Función									
MD				1	6,5	2	5	Sin esfuerzo asignado	14,5
I				0,25	3	1	1,25		5,5
MT				2	9,5	3	6,5		21
MA				0,75	4,5	1,75	3,5		10,5
MP				0,25	4	0,75	3		8
Estim. total	0,5	0,5	0,5	4,25	27,5	8,5	19,25		61 PM
% esfuerzo	0,8	0,8	0,8	6,9	45	13,9	31,5		100 (99,7)

- $\text{coste} = 61 \text{ PM} * 7800 \text{ €/PM} = 475800 \text{ €}$

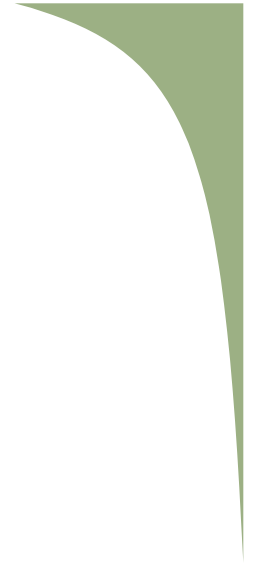




# Precisión

- La precisión de la estimación depende de varios factores:
  - El grado en el que el planificador ha estimado adecuadamente el tamaño del producto a construir.
  - La habilidad para traducir la estimación del tamaño en esfuerzo y dinero.
    - Depende fundamentalmente de la existencia de métricas.
  - El grado en que el plan del proyecto refleja las habilidades del equipo de software.
  - La estabilidad de los requisitos y del entorno que soporta el esfuerzo de la Ingeniería del Software.





# MODELOS PARAMÉTRICOS DE ESTIMACIÓN





# Introducción

- Utilizan formulas derivadas empíricamente para predecir el esfuerzo como una función del tamaño del producto.
  - Validez restringida al entorno donde se dedujo la fórmula.
- El tamaño del producto se obtiene por descomposición del problema.
- La estructura común es:
$$E = A + B * x^C$$
- donde:
  - E es el esfuerzo, medido generalmente en PM.
  - A, B y C son constantes obtenidas empíricamente que reflejan ciertas características clave del problema o el contexto de resolución.
  - x es una variable de estimación del tamaño del producto, medida generalmente en LDC o PF.



# Modelos paramétricos: la ecuación del software

- La ecuación del software [Putnam, 1992] es:

$$E = B * \left( \frac{LDC}{P} \right)^3 * \frac{1}{t^4}$$

- donde:

- $E$  es el esfuerzo medido en PM.
- $t$  es la duración del proyecto en meses.
- $B$  es el factor especial de destrezas.
  - Ej.  $5 \leq KLDC \leq 15 \rightarrow B = 0,16$
  - Ej.  $KLDC > 70 \rightarrow B = 0,39$
- $P$  es el parámetro de productividad.
  - Ej. sistemas empotrados de tiempo real,  $P = 2000$
  - Ej. software de telecomunicaciones y sistemas software,  $P = 10000$
  - Ej. aplicaciones comerciales  $P = 28000$





# COCOMO II: modelos

- COCOMO II (*CO*nstructive *CO*st Model) [Boehm et al., 1995] es un modelo paramétrico basado en la construcción de una serie de modelos iniciales.
- Los modelos tienen una estructura común y una serie de parámetros que se calibran sobre una base de proyectos previos.
- Los modelos son:
  - Modelo de diseño previo
    - Primeras etapas de Ingeniería del Software
    - Prototipado y evaluación
  - Modelo de fase posterior a la arquitectura
    - Requisitos establecidos
    - Arquitectura básica del software establecida
    - Construcción del software



## COCOMO II: ecuaciones (1/2)

$$PM = A * Tam^E * \prod_i EM_i \quad \text{en PM}$$

$$E = B + 0,01 * \sum_{i=1..5} SF_i$$

$$T = C * PM^F \quad \text{en meses}$$

$$F = D + 0,2 * (E - B)$$

- donde
  - $Tam$  es el tamaño en KLDC (o PF haciendo la conversión).
  - $SF_i$  son los factores de escala.
  - $EM_i$  son los multiplicadores de esfuerzo.
    - 7 diseño temprano o previo
    - 16 posteriores a la arquitectura
  - $A, B, C$  y  $D$  son constantes que dependen de históricos de proyectos.





## COCOMO II: ecuaciones (2/2)

Scale Factor	Description
PREC	<i>Precedentness</i>
FLEX	<i>Development Flexibility</i>
RESL	<i>Architecture/Risk Resolution</i>
TEAM	<i>Team Cohesion</i>
PMAT	<i>Process Maturity</i>

Effort multiplier	Description
RCPX	<i>Product Reliability and Complexity</i>
RUSE	<i>Required Reuse</i>
PDIF	<i>Platform Difficulty</i>
PERS	<i>Personnel Capability</i>
PREX	<i>Personnel Experience</i>
FCIL	<i>Facilities</i>
SCED	<i>Scheduled</i>

Effort multiplier	Description
RELY	<i>Required Software Reliability</i>
DATA	<i>Data Base Size</i>
DOCU	<i>Documentation Match to Life-Cycle Needs</i>
TIME	<i>Execution Time Constraint</i>
SITE	<i>Multisite development</i>
ACAP	<i>Analyst Capability</i>
...	...

Diseño previo

Posterior a la arquitectura:  
5 para factores del producto, 3 de la plataforma, 3 del proyecto y 6 del personal

## Ejemplo: factores de escala – PREC

Feature	Very low	Nominal / High	Extra high
Organizational understading of project objectives	General	Considerable	Through
Experience in working with related software systems	Moderate	Considerable	Extensive
Concurrent development of associated new hardware and operational procedures	Extensive	Moderate	Some
Need for innovative data processing architectures, algorithms	Considerable	Some	Minimal





## COCOMO II: calibración

- La idea es que cada organización calibre los valores de  $A$ ,  $B$ ,  $C$ ,  $D$ ,  $EM_i$  y  $SF_i$  en base al conocimiento que tenga de proyectos anteriores.
- En cualquier caso, los autores proporcionan unas tablas con calibraciones para sus proyectos.
  - De *extra low* a *extra high*.



# Calibrado del modelo para el diseño previo

Baseline effort constants  $A = 2.94$   $B = 0.91$   
Baseline schedule constants  $C = 3.67$   $D = 0.28$

Driver	Symbol	VL	L	N	H	VH	EH
PERS	EM <sub>1</sub>	2.12	1.62	1.26	1.00	0.63	0.50
RCPX	EM <sub>2</sub>	0.49	0.60	0.83	1.00	1.91	2.72
PDIF	EM <sub>3</sub>			0.87	1.00	1.81	2.61
PREX	EM <sub>4</sub>	1.59	1.33	1.12	1.00	0.74	0.62
FCIL	EM <sub>5</sub>	1.43	1.30	1.10	1.00	0.73	0.62
RUSE	EM <sub>6</sub>			0.95	1.00	1.15	1.24
SCED	EM <sub>7</sub>		1.43	1.14	1.00	1.00	

VL = Very Low  
L = Low  
N = Nominal  
H = High  
VH = Very High  
EH = Extra High





# PROCESO DE ESTIMACIÓN





# Proceso de estimación

- Es conveniente hacer varias estimaciones del proyecto.
  - Siguiendo varias técnicas
  - Dando una estimación optimista, media y pesimista
    - $Valor_{estimación} = (Valor_{optimista} + 4 * Valor_{medio} + Valor_{pesimista}) / 6$
- Dicha estimación puede basarse en:
  - Datos históricos
    - Se necesita tomar métricas sobre los proyectos.
    - Las métricas han de estar actualizadas y ser aplicadas correctamente.
  - Experiencia/intuición
- Una variación *razonable* en las estimaciones es en una franja del 20%.
  - Para casos con alta incertidumbre.
  - Por encima las estimaciones no son válidas.





## Ejemplo: CAD – comparativa(7/7)

- Nótese que para el ejemplo hemos obtenido:
  - LDC: 65 PM
  - PF: 71 PM
  - Proceso: 61 PM
- Media: 65,7 PM
- Variación máxima: 8% RAZONABLE
- Lo normal es obtener variaciones razonables.





# CONCLUSIONES







# Conclusiones

- La planificación del proyecto debe basarse en una estimación del esfuerzo a realizar y la evaluación de posibles alternativas.
  - Desarrollar vs comprar vs subcontratar
- La precisión de la estimación depende de múltiples factores.
  - La estimación se adapta con el proyecto, por lo que se hace necesario actualizar la planificación.
- En la estimación se consideran varias situaciones alternativas.
  - Casos mejor, medio y peor
- Técnicas de estimación
  - Descomposición del problema
  - Descomposición del proceso
  - Modelos paramétricos





# Glosario

- CAD = *Computer-Aided Design*
- COCOMO = *CO*nstructive *CO*st Model
- DXF = *Drawing eXchange Format*
- EDT = Estructura de Descomposición del Trabajo = WBS
- IS = Ingeniería del Software
- LDC = Línea de Código
- PF = Punto de Función
- PFA = PF Ajustado
- PM = Persona-Mes
- PO = Punto Objeto
- WBS = *Work Breakdown Structure* = EDT





# Referencias

- R. Pressman: Ingeniería del Software. Un enfoque práctico, 7ª edición. McGraw-Hill, 2010.
  - Capítulo 26
- I. Sommerville: Ingeniería del Software, 7ª edición. Addison Wesley, 2007.
  - Capítulo 26.2
- B. Boehm, B. Clark, E. Horowitz, C. Westland, R. Madachy y R. Selby: *Cost Models for Future Software Life-cycle Processes: COCOMO 2.0*. En J. D. Arthur y S. M. Henry (eds.) *Annals of Software Engineering* 1, Special Volume on Software Process and Product Measurement, pp. 45–60, 1995.
- University of Southern California: *COCOMO II Model Definition Manual*. Disponible en: [http://sunset.usc.edu/csse/research/COCOMOII/cocomo\\_main.htm](http://sunset.usc.edu/csse/research/COCOMOII/cocomo_main.htm), accedido el 20/11/2011.

