

## Ingeniería del Software.

### Tema II: Elementos de gestión del proyecto de software.

#### Conferencia #5. Estimación del proyecto de software.

#### Desarrollo

Hoy en día, el software es el elemento más caro de la mayoría de los sistemas informáticos. Un error en la estimación del coste puede ser lo que marque la diferencia entre beneficios y pérdidas. Sobrepassarse en el coste puede ser desastroso para el equipo de desarrollo.

La estimación del coste y del esfuerzo del software nunca será una ciencia exacta. Son demasiadas las variables (humanas, técnicas, de entorno, políticas) que pueden afectar al coste final del software y al esfuerzo aplicado para desarrollarlo. Sin embargo, la estimación del proyecto de software puede dejar de ser un oscuro arte para convertirse en una serie de pasos sistemáticos que proporcionen estimaciones con un grado de riesgo aceptable.

Alternativas:

1. Postergar la estimación.
2. Basar las estimaciones en proyectos similares ya terminados.
3. Utilizar técnicas de descomposición relativamente sencillas para generar las estimaciones de coste y de esfuerzo del proyecto.
4. Desarrollar un modelo empírico para el cálculo de costes y esfuerzos del software.

La primera opción, aunque atractiva, no es práctica. Las estimaciones de costos han de ser proporcionadas a *priori*. Sin embargo, hay que reconocer que cuanto más tiempo esperemos, más cosas sabremos, y cuanto más sepamos, menor será la probabilidad de cometer serios errores en nuestras estimaciones.

La segunda opción funciona bien si estamos en presencia de un proyecto bastante parecido a otro realizado con anterioridad (que haya sido bien estimado).

Las técnicas de descomposición utilizan un enfoque de “divide y vencerás” para la estimación del proyecto de software. Mediante la descomposición del proyecto en sus funciones principales y en las tareas de ingeniería del software correspondientes, la estimación del coste y del esfuerzo puede realizarse de una forma escalonada idónea. Se pueden utilizar los modelos empíricos de estimación como complemento de las técnicas de descomposición, ofreciendo un enfoque de estimación potencialmente valioso por derecho propio.

Los enfoques 3 y 4 se basan en datos históricos y toman como base:

$$d = f(v_i)$$

Donde el aspecto estimado  $d$  (costo, esfuerzo, duración del proyecto, etc.) es función de los parámetros independientes  $v_i$  (LDC o PF).

#### Técnicas de descomposición.

Se descompone el problema original hasta obtener problemas que sean manejables. Se resuelven luego individualmente con la suposición de que las soluciones puedan combinarse para obtener la total.

El enfoque de descomposición se puede analizar desde dos puntos de vista diferentes: descomposición del problema y descomposición del proceso. La estimación hace uso de una o ambas formas de particionamiento. Pero antes de hacer una estimación, el planificador del proyecto debe comprender el ámbito del software a construir y generar una estimación de su tamaño.

#### Estimación de LDC y PF.

Los datos de LDC y PF se utilizan de dos formas durante la estimación del proyecto de software:

1. Como una *variable de estimación* que se utiliza para dimensionar cada elemento del software
2. Como métricas de *línea base* recopiladas de proyectos anteriores y utilizadas junto con variables de estimación para desarrollar proyecciones de costo y de esfuerzo.

Las técnicas de estimación basadas en LDC son diferentes a las basadas en PF, aunque tienen aspectos en común. El planificador parte de una declaración restringida del ámbito del software a partir del cual intenta descomponer el software en pequeñas subfunciones que puedan ser estimadas individualmente. Luego estima LDC y PF (variable de estimación) para cada subfunción. Seguidamente aplica las métricas básicas de productividad (LDC/p-m ó PF/p-m) a la variable de estimación apropiada y deriva el costo y el esfuerzo para la subfunción.

Al combinar las estimaciones para las subfunciones se obtiene la estimación total para el proyecto entero.

Cuando se utiliza LDC como variable de estimación, la descomposición es absolutamente esencial y a menudo se lleva a considerables niveles de detalle. Cuanto más grande sea el grado de particionamiento, más probable será que pueda desarrollar estimaciones más exactas.

Para estimaciones de PF, la descomposición funciona de diferente manera. En lugar de centrarse en la función, se estiman cada una de las características del dominio de información (entradas, salidas, archivos de datos, peticiones, e interfaces externas) y los catorce valores de ajuste de la complejidad. En este caso el grado de descomposición al que se llega es menor porque los datos requeridos para estimar los PF son más macroscópicos.

Recordar que LDC se mide directamente y PF indirectamente.

Independientemente de la variable de estimación que use, el planificador del proyecto normalmente especifica un intervalo de valores para cada función descompuesta. A partir de los datos históricos o usando su intuición cuando no hay referencias previas, el planificador estima valores optimistas, más probables y pesimistas de LDC o de PF para cada función. Al dar el intervalo se está especificando de hecho un margen de incertidumbre. Entonces se calcula el valor esperado (E) de LDC o de PF por una media ponderada de las estimaciones optimista (A), más probable (M) y pesimista (B) según la expresión:

$$E = \frac{A + 4M + B}{6}$$

Esta fórmula confiere mayor credibilidad a la estimación más probable y sigue una distribución de probabilidad  $\beta$ .

Se supone que la probabilidad de que el resultado real de LDC o PF quede fuera del intervalo establecido por los valores A y B es muy bajo.

Una vez calculado el valor esperado para la variable de estimación, se aplican los datos de productividad para LDC o PF.

Ejemplo 2-3-1: Si se han estimado 310PF y si la productividad media (de acuerdo a proyectos anteriores) es de 5.5PF/p-m, entonces el esfuerzo total del proyecto será:

$$\text{Esfuerzo} = 310/5.5 = 56 \text{ p-m}$$

Ejemplo 2-3-2: Considerando que la productividad media fuera 490 LDC/p-m, se podría estimar la productividad para las subfunciones considerablemente más complejas que la media es solo de 300 LDC/p-m y para las funciones simples es 650 LDC/p-m.

Debe tenerse presente que se deben arreglar las métricas de productividad media para tomar en cuenta los efectos inflacionistas, el incremento de complejidad del proyecto, el nuevo personal y otras características de desarrollo.

Cualquier técnica de estimación, no importa lo sofisticada que sea, se debe volver a comprobar con otro enfoque. Incluso entonces, va a prevalecer el sentido común y la experiencia.

Para ilustrar técnicas de estimación LDC y PF se considerará un paquete de software CAD.

Ejemplo 2-3-3: Según la especificación del sistema el software va a ejecutarse en una estación de trabajo de ingeniería y que debe interconectarse con varios periféricos de gráficos de computadora entre los que se incluyen un ratón, un digitalizador, una pantalla a color de alta resolución y una impresora láser.

La declaración preliminar el ámbito del software dice:

El software de CAD aceptará datos geométricos de dos y tres dimensiones por parte del ingeniero. El ingeniero se interconectará y controlará el sistema de CAD por medio de una interfaz de usuario que va a exhibir las características de un buen diseño de interfaz hombre-máquina. Una base de datos de CAD contiene todos los datos geométricos y la información de soporte. Se desarrollarán módulos de análisis de diseño para producir la salida requerida que se va a visualizar en varios dispositivos gráficos. El software se diseñará para controlar e interconectar dispositivos periféricos entre los que se incluyen un ratón, un digitalizador y una impresora láser.

Esta es una declaración preliminar del ámbito. Habría que ampliar más algunas frases para proveer detalles concretos y limitaciones cuantitativas. Supondremos que se lleva a cabo un refinamiento y que se identifican las siguientes funciones principales:

- Interfaz de usuario y facilidades de control (IUFC),
- Análisis geométrico de dos dimensiones (AG2D),
- Análisis geométrico de tres dimensiones (AG3D),
- Gestión de base de datos (GBD),
- Facilidades de presentación gráfica de computadora (FPGC),
- Control de periféricos (CP),
- Módulos de análisis del diseño (MAD).

Por la técnica de descomposición se desarrolla la tabla de estimación siguiente:

Función	A	M	B	E	\$/LDC	LDC/p-m	Costo	Esfzo
IUFC	1800	2400	2650	2342	14	315	32783	7.4
AG2D	4100	5200	7400	5383	20	220	107667	24.5
AG3D	4600	6900	8600	6800	20	220	136000	30.9
GBD	2950	3400	3600	3358	18	240	60450	14.0
FVGC	4050	4900	6200	4975	22	200	109450	24.9
CP	2000	2100	2450	2142	28	140	59967	15.3
MAD	6600	8500	9800	8400	18	300	151200	28.0
<b>Total</b>				<b>33400</b>			<b>657517</b>	<b>145.0</b>

Analizando la tabla se aprecia que el planificador está bastante seguro de las LDC requeridas para el control de periféricos (450 líneas de diferencia). La función de AG3D es bastante desconocida (4000 líneas de diferencia). Las métricas de productividad (columnas 5 y 6) son derivadas de la línea base histórica. Se miden en \$/LDC y en LDC/p-m. Se usan diferentes valores de estas métricas para cada función, de acuerdo a su grado de complejidad. Los valores de la columna 7 se obtienen multiplicando la columna 4 y 5. Los valores de la columna 8 se obtienen dividiendo la columna 4 y 6. Se ve que hay un estimado total de 33400 LDC y un costo total de aproximadamente 657517, con un esfuerzo aproximado de 145 p-m.

Para el caso de los PF se hace la estimación que se muestra en la siguiente tabla. Para los propósitos de esta estimación, el factor de ponderación de la complejidad se asume como la media.

Valor del dominio	Opt	Prob	Pes	Est	Peso	PF	Factor	Valor
Entradas de usuario	20	24	30	24	4	96	Copias de seguridad y recuperación	4
Salidas de usuario	12	15	22	16	5	80	Comunicaciones de datos	2
Peticiones de usuario	16	22	28	22	4	88	Procesamiento distribuido	0
Archivos	4	4	5	4	10	40	Rendimiento crítico	4
Interfaces existentes	2	2	3	2	7	14	Entorno operativo existente	3
<b>Cuenta total</b>						<b>318</b>	Entrada de datos <i>online</i>	4
							Transacciones de entrada en múltiples pantallas	5
							Archivos maestros actualizados <i>online</i>	3
							Complejidad de valores del dominio de información	5
							Complejidad del procesamiento interno	5
							Código diseñado para ser reutilizable	4
							Conversión e instalación en el diseño	3
							Instalaciones múltiples	5
							Aplicación diseñada para facilitar los cambios	5
							<b>Total</b>	<b>52</b>

**PF = 372**

Como se ve, se ha obtenido un valor estimado para los puntos de función. A partir de ahí, junto a los datos de costo y productividad históricos (\$/PF y PF/p-m) para el grupo de desarrollo se pueden calcular los valores estimados de costo y esfuerzo requerido para todo el proyecto.

### Estimación del esfuerzo.

Es la técnica más usada para calcular el costo de un proyecto de Ingeniería del Software. Para resolver cada tarea se requiere cierto número de personas-mes. Se asocia un costo a cada unidad de esfuerzo y se deriva el costo total estimado.

Al igual que las técnicas de LDC y PF, la estimación del esfuerzo comienza por la delimitación de las funciones del software, obtenidas del ámbito del proyecto (que a su vez se deriva de la especificación del sistema).

Para instrumentar cada función hay que realizar un conjunto de tareas de Ingeniería del Software (análisis de requisitos, diseño, codificación, pruebas). Se pueden representar en una tabla con las funciones encabezando filas y las tareas encabezando columnas. El planificador estima el esfuerzo en personas-mes para cada etapa de ingeniería de cada función y estos datos son las entradas en la tabla. Luego se aplican tarifas laborales (costo/esfuerzo) a cada tarea. Normalmente las tarifas cambian según la tarea (personal *senior* es analista y personal *junior* es implementador).

En un último paso se calculan esfuerzos y costos para cada función y tarea del proyecto. Si se ha hecho la estimación del esfuerzo independientemente de la estimación LDC o PF, tendremos dos estimaciones de costos y esfuerzos que se pueden comparar y conciliar. Si ambos enfoques muestran concordancias aceptables, habrá buenas razones para creer que los estimados son fiables. Si por el contrario, los resultados de ambas técnicas de descomposición no tienen mucho en común, habrá que realizar investigaciones adicionales de los datos para la estimación.

**Ejemplo 2-3-4:** El software de CAD antes analizado se enfoca ahora por los esfuerzos requeridos para las tareas de Ingeniería del Software en cada una de las funciones identificadas en el ámbito.

Tareas	Análisis de Requisitos	Diseño	Codificación	Prueba	Total
IUFC	1.0	2.0	0.5	3.5	7
AG2D	2.0	10.0	4.5	9.5	26
AG3D	2.5	12.0	6.0	11.0	31.5
GBD	2.0	6.0	3.0	4.0	15
FVGC	1.5	11.0	4.0	10.5	27
CP	1.5	6.0	3.5	5.0	16
MAD	4.0	14.0	5.0	7.0	30
<b>Total (p-m)</b>	<b>14.5</b>	<b>61</b>	<b>26.5</b>	<b>50.5</b>	<b>152.5</b>
<b>Tarifa (\$/p-m)</b>	<b>5200</b>	<b>4800</b>	<b>4250</b>	<b>4500</b>	
<b>Costo (\$)</b>	<b>75400</b>	<b>292800</b>	<b>112625</b>	<b>227250</b>	<b>708075</b>

Todos los esfuerzos están en personas-mes

Esfuerzo estimado para las tareas (total)

Costo estimado para las tareas (total)

Se puede ver que en las tareas de desarrollo “vitales” (análisis de requisitos y diseño) se usan 75 p-m, lo que indica la gran importancia relativa de estas labores.

La tarifa laboral refleja los costos laborales gravados, es decir, que incluyen gastos adicionales de la empresa. Se asume que los costos del análisis de requisitos (5200 \$/p-m) son un 22% superiores a los de las etapas de implementación y pruebas.

Si se comparan los estimados totales de costo y esfuerzo (\$708000 y 153 p-m, respectivamente) con los obtenidos por la técnica de LDC (\$657000 y 145 p-m) se aprecian diferencias no significativas (7% y 5%), o sea, un alto grado de concordancia entre ambos enfoques.

Las causas de posibles discrepancias entre resultados obtenidos por métodos diferentes pueden ser:

1. No se ha especificado adecuadamente el ámbito del software o ha sido malinterpretado por el planificador.
2. Se han usado datos de productividad (LDC) inadecuados, obsoletos o se han aplicado mal.

Se deben identificar las causas de la divergencia y llegar a reconciliar las estimaciones.

### Modelos empíricos de estimación.

Un modelo de estimación para el software de computadora utiliza fórmulas derivadas empíricamente para predecir el esfuerzo como una función de LDC y PF.

Los datos empíricos se obtienen de una muestra limitada de proyectos, por lo que el mismo modelo no es adecuado para toda clase de software ni para todos los entornos de desarrollo.

Se han desarrollado algunos modelos, como son:

Modelos univariable estáticos. Tienen la forma:

$$recurso = C_1(característica\_estimada)^{C_2}$$

Donde:

Recurso: podría ser el esfuerzo, la duración del proyecto, la cantidad de personal o las líneas requeridas de documentación del programa.

$C_1$  y  $C_2$ : se derivan de los datos de proyectos anteriores.

Característica estimada: puede ser la cantidad de LDC, el esfuerzo (si ya está estimado) u otra característica del software.

Se pueden derivar modelos de este tipo para un entorno local si hay suficientes datos históricos disponibles.

Modelos multivariable estáticos. Tienen la forma:

$$recurso = C_{i1}e_1 + C_{i2}e_2 + \dots$$

Donde:

$e_i$  es la característica i-esima del software.

$C_{i1}$  y  $C_{i2}$  constantes obtenidas empíricamente para la característica i-esima.

Modelos multivariable dinámico. Proyecta los requisitos de recursos como una función del tiempo. Si se obtiene empíricamente el modelo, los recursos se definen en una serie de pasos consecutivos en el tiempo que asignan cierto porcentaje de esfuerzo (u otro recurso) a cada etapa de ingeniería.

Ejemplos de modelos:

$$E = 5.2 \times (KLDC)^{0.91}$$

Modelo de Walston-Felix

$$E = 5.5 + 0.73 \times (KLDC)^{1.16}$$

Modelo de Bailey-Basisli

$$E = 5.288 \times (KLDC)^{1.047}$$

Modelo de Doty para  $KLDC > 9$

$E = -13.39 + 0,054 PF$	Modelo de Albretch y Gaffney
$E = 60.62 \times 7.728 \times 10 PF^3$	Modelo de Kemerer
$E = 585.7 + 15.12 PF$	Modelo de Matson, Bamett y Mellichamp

Un rápido examen de los modelos anteriores indica que cada uno producirá un resultado diferente para el mismo valor de LDC y PF. La implicación es clara: los modelos de estimación se deben calibrar para necesidades locales.

### Modelo COCOMO<sup>1</sup>. (COConstructive COSt MOdel)

Introduce una jerarquía de modelos de estimación de software.

- Modelo 1. (básico) Calcula el esfuerzo (y el coste) del desarrollo de software en función del tamaño del programa, expresado en las líneas estimadas de código (LDC).
- Modelo 2. (intermedio) Calcula el esfuerzo del desarrollo de software en función del tamaño del programa y de un conjunto de “conductores de costo” que incluyen la evaluación subjetiva del producto, del hardware, del personal y de los atributos del proyecto.
- Modelo 3. (avanzado) El modelo COCOMO avanzado incorpora todas las características de la versión intermedia y lleva a cabo una evaluación del impacto de los conductores de costo en cada fase (análisis, diseño, etc.) del proceso de ingeniería del software.

Los modelos COCOMO están definidos para tres tipos de proyectos de software:

1. Modo orgánico. Proyectos relativamente pequeños y sencillos en los que trabajan pequeños equipos, con buena experiencia en la aplicación, sobre un conjunto de requisitos un poco rígidos (por ejemplo, un programa de análisis termal desarrollado para un grupo calórico).
2. Modo semiacoplado. Proyectos de software intermedios (en tamaño y complejidad) en los que equipos, con variados niveles de experiencia, deben satisfacer requisitos poco o medio rígidos (p. ej.: un sistema de procesamiento de transacciones con requisitos fijos para un hardware de terminal o un software de gestión de base de datos).
3. Modo empotrado. Proyectos de software que deben ser desarrollados en un conjunto de hardware, software y restricciones operativas muy restringidas (p. ej.: software de control de navegación para un avión).

Las ecuaciones del COCOMO básico tienen la siguiente forma:

$$E = a_b (KLDC)^{b_b}$$

$$D = c_b (E)^{d_b}$$

Donde: E: Esfuerzo aplicado en personas-mes (p-m).

D: Tiempo de desarrollo en meses cronológicos.

Los coeficientes  $a_b$  y  $b_b$  y los exponentes  $c_b$  y  $d_b$  se muestran en la siguiente tabla:

Proyecto de Software	$a_b$	$b_b$	$c_b$	$d_b$
Orgánico	2.4	1.05	2.5	0.38
Semiacoplado	3.0	1.12	2.5	0.35
Empotrado	3.6	1.20	2.5	0.32

Ejemplo 2-3-5: Se aplicará el modelo COCOMO al software de CAD descrito anteriormente. Mediante la estimación de LDC, se utiliza el modelo semiacoplado.

Nota: En el libro de Pressman (4ta ed.) hay errores en este ejemplo a la hora de calcular los valores, porque no se seleccionan los datos adecuados de la tabla y aparentemente da un resultado correcto.

$$\begin{array}{lll}
 E = 3.0(KLDC)^{1.12} & D = 2.5(E)^{0.35} & N = E / D \\
 E = 3.0(33.4)^{1.12} & D = 2.5(153)^{0.35} & N = 153 / 14.5 \\
 E = 153p - m & D = 14.5 \text{ meses} & N \approx 11 \text{ personas}
 \end{array}$$

N puede disminuir a costa de aumentar la duración del proyecto.

Estos valores se ajustan bastante a los obtenidos en los ejemplos anteriores, aunque al ser COCOMO un modelo empírico, es necesario ajustarlo a las condiciones típicas del proyecto de software para el cual se aplica.

<sup>1</sup> BOEHM, B.: Software Engineering Economics. Prentice-Hall, 1981