

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA
INGENIERÍA INFORMÁTICA

Grupo de Investigación y Desarrollo en Ingeniería de Software



**Estimación del Esfuerzo en la etapa Post-Arquitectura usando
el modelo COCOMO II
Versión 2.0**

ELABORADO POR: Karin Meléndez
Abraham Dávila
Carla Basurto

melendez.ka@pucp.edu.pe
edavila@pucp.edu.pe
cbasurto@pucp.edu.pe

Lima, 30 de enero del 2005

Historial de Revisiones

Historial de revisiones				
Ítem	Fecha	Versión	Descripción	Equipo
1	19/09/2004	1.0	Versión inicial.	Karin Melendez Abraham Dávila
2	31/05/2005	2.0	Incluye el cálculo de la duración del proyecto	Karin Melendez Abraham Dávila

ÍNDICE

1.	Definición	1
2.	Modelos de COCOMO II.....	2
3.	Aplicación.....	2
4.	Factores de escala.....	3
5.	Multiplicadores de esfuerzo	4
6.	Duración del Proyecto	9
7.	Ejemplo de Aplicación	9
8.	Solución del ejemplo	10
9.	Referencias Bibliográficas	13

EL MODELO COCOMO II

1. Definición

El modelo constructivo de costos más conocido como COCOMO por sus siglas en inglés de *Constructive Cost Model* es uno de los más difundido y exitoso en la ingeniería de software. COCOMO es, en general, un modelo algorítmico de costos creado en los años 80 por Barry Boehm y su equipo luego de un amplio estudio estadístico de diversos proyectos informáticos. El modelo ha sido mejorado en sucesivas oportunidades y aproximadamente desde el año 1995 se le conoce como COCOMO II. Este nuevo modelo incluye los enfoques de construcción de prototipos, uso de 4GL y desarrollo basado en componentes entre otros aspectos actuales del desarrollo de software.

En COCOMO II, el esfuerzo del desarrollo de software es expresado en meses-persona, donde un mes-persona representa el trabajo de una persona en un mes dentro de un proyecto de desarrollo de software. El número de meses-persona es diferente que la duración del proyecto, este último obtenido de la diferencia entre la fecha de inicio y fin del proyecto; y también es diferente del número de personas que participan de un proyecto. Para un proyecto de 4 meses donde participan un analista programador durante todo el proyecto y un programador en los dos últimos meses, se obtendría los siguientes resultados:

duración del proyecto = 4 meses.
número de personas = 2 personas.
esfuerzo del proyecto = 6 meses-persona.

Este modelo [COCOMO II,1-2] puede ser usado para:

- Hacer inversiones o tomar otras decisiones financieras que involucren esfuerzo de desarrollo de software.
- Establecer presupuestos y calendarios de proyectos como base para la planificación y el control.
- Decidir o negociar entre los costos del software, calendario, funcionalidad, rendimiento (*performance*) o factores de calidad.
- Tomar decisiones sobre el costo de software y el manejo de riesgos del calendario.
- Decidir qué partes del sistema software se desarrollarán, re-utilizarán, sub-contratarán o comprarán.
- Tomar decisiones sobre el software existente: qué partes se van a modificar, dejar fuera, etc.
- Fijar estrategias de inversión combinadas para mejorar la capacidad de la organización, reutilización, herramientas, madurez de procesos, subcontratación, etc.
- Decidir como implementar una estrategia de mejora del proceso.

Los objetivos del modelo [COCOMO II, 3-4] son:

- Proveer estimaciones de costo y calendarios precisos tanto para los proyectos actuales como futuros. La versión COCOMO 81, construido para el ciclo de vida en cascada, ha sido mejorado en COCOMO II y ahora permite trabajar con ciclos de vida incremental e iterativo, entre otros.
- Permitir a las organizaciones re-calibrar, ajustar o extender fácilmente COCOMO II para ajustar a sus propias condiciones de trabajo. La información necesaria para dichos ajustes está disponible.
- Proporcionar definiciones cuidadosas y fáciles de comprender de las entradas, salidas y supuestos del modelo.
- Proporcionar un modelo constructivo del costo, este objetivo sigue vigente desde la versión original de COCOMO 81.
- Proporcionar un modelo normativo, que permite distribuir los recursos necesarios para un desarrollo o mantenimiento efectivo del software.

- Proporcionar un modelo evolutivo, que añada nuevas capacidades para cubrir nuevas necesidades.

En el libro de Ingeniería de Software de Ian Sommerville [Sommerville, 522] el autor señala que “*eligió presentar el modelo de COCOMO en su libro por que:*

- *Está bien documentado, es de dominio público y lo apoyan el dominio público y las herramientas.*
- *Se ha utilizado y evaluado muy ampliamente.*
- *Tienen una gran tradición desde su primera versión en 1981, pasando por su refinamiento para el desarrollo de software en ADA en 1989, hasta su versión más reciente, publicada en 1995.”*

2. Modelos de COCOMO II

La estimación de proyectos es una actividad que se debe realizar desde las primeras etapas en que se plantea la idea del proyecto informático de construcción de software, sin embargo, la estimación podría no ser adecuada al existir una gran incertidumbre sobre muchos aspectos del proyecto y del producto a construir. Considerando lo anterior, COCOMO II, presenta dos grandes modelos de acuerdo al mercado al cual está dirigido:

- El modelo para el sector de composición de aplicaciones, que está basado en puntos de objetos [COCOMO II, 8]. Estos proyectos son típicamente desarrollados por equipos pequeños y en algunas semanas.
- El modelo para la estimación del desarrollo de generación de aplicaciones, integración de sistemas o infraestructura, está basada en una fórmula ajustable de acuerdo a la etapa en que es evaluada (construcción de prototipos, diseño inicial, post-arquitectural) [COCOMO II, 9].

Para el segundo modelo, se establecen diversos momentos de la estimación y que de acuerdo a ello se ajusta la fórmula. Las etapas son:

- **Construcción de prototipos inicial.**- La estimación incluye el esfuerzo de construcción de prototipos para resolver potenciales problemas de gran riesgo como interfaces de usuario, interacción de software y sistemas, eficiencia o madurez de la tecnología. La mejor estimación se realiza mediante puntos de objeto como en el caso de modelo composición de aplicaciones.
- **Diseño inicial.**- La estimación se puede realizar a partir de un catálogo de requisitos y con un diseño inicial (idea inicial de alternativas de arquitectura y conceptos de operación). La estimación se basa en el cálculo de puntos de función que luego son convertidos a líneas de código fuente y un conjunto reducido de factores que representan los ajustes para el cálculo del costo.
- **Post-arquitectónico.**- La estimación se realiza luego de completar el diseño de la arquitectura del sistema utilizando los puntos de función que son convertidos a líneas de código fuente y un conjunto amplio de factores que ajustan los costos de acuerdo al producto, el equipamiento, el personal y el proyecto mismo.

3. Aplicación

El modelo de estimación de esfuerzo COCOMO II introdujo la siguiente ecuación: [COCOMO II, 68-70]

- **Ecuaciones del tamaño Post-Arquitectura**

$$\text{Esfuerzo} = A \times \text{Tamaño}^E \times \left(\prod \text{Multiplicador} \right) \quad \text{Ec. 3.1}$$

$$\text{Donde: } E = B + 0.01 \times (\text{? Factor}) \quad \text{Ec. 3.2}$$

los valores son:

Símbolo	Descripción
A	Coefficiente de esfuerzo, calibrado en 2.94
B	Base del exponente de escalamiento, calibrado en 0.91
Tamaño	Medida del software en líneas de código fuente
E	Exponente de escalamiento de tipo de proyecto
Multiplicador	17 multiplicadores de esfuerzo
Factor	5 factores de escala del proyecto
Esfuerzo	Esfuerzo del proyecto en meses-persona

4. Factores de escala

Los siguientes factores de escala son utilizados para el cálculo del exponente en la ecuación 3.2:

4.1 Precedentes.-Refleja la experiencia previa de la organización desarrolladora con respecto al tipo de proyectos. Muy bajo significa sin experiencia previa. Extra alto significa que la organización está completamente familiarizada con el dominio de la aplicación a desarrollarse.

4.2 Flexibilidad de desarrollo.- Refleja el grado de flexibilidad del proceso de desarrollo. Muy bajo significa que se utiliza un proceso prescrito por ejemplo: uso de determinadas técnicas, etapas, actividades o procesos. Extra alto significa que el cliente establece sólo metas generales y los desarrolladores pueden tomar decisiones sobre cómo llevar a cabo el proyecto y qué técnicas usar en cada momento.

4.3 Arquitectura/Solución de riesgo.- Refleja el alcance del análisis de riesgo que se debe llevar a cabo. Muy bajo significa que se requiere de poco análisis. Extra alto significa que es indispensable un análisis de riesgo completo y muy detallado.

Criterios de Aplicación						
Factor de escala	Muy Bajo	Bajo	Nominal	Alto	Muy Alto	Extra Alto
Precedentes	Ninguno	Pocos	Alguno	Familiar	Muy Familiar	Cotidiano
	6.20	4.96	3.72	2.48	1.24	0.00
Flexibilidad de desarrollo	Riguroso	Poco Relajado	Algo Relajado	Conformidad general	Conformidad básica	Objetivos generales
	5.07	4.05	3.04	2.03	1.01	0.00
Arquitectura / Solución de riesgo	Pequeña 20%	Alguna 40%	Frecuente 60%	General 75%	Rutinaria 90%	Completa 100%
	7.07	5.65	4.24	2.83	1.41	0.00
Cohesión del equipo / interacción	Muy difícil	Difícil	Básica	Grande	Alta	Total
	5.48	4.38	3.29	2.19	1.10	0.00
Madurez de proceso	Media ponderada de las respuestas "sí" al cuestionario de madurez del CMM					
	SW-CMM 1 Bajo	SW-CMM 1 Alto	SW-CMM 2	SW-CMM 3	SW-CMM 4	SW-CMM 5
	7.80	6.24	4.68	3.12	1.56	0.00

Tabla 4.1 Valores de los factores de escala

4.4 Cohesión del equipo/interacción.- Refleja el nivel de conocimiento de los miembros del equipo de desarrollo entre sí y qué tan bien trabajan juntos. Muy bajo significa que las interacciones podrían ser muy difíciles. Extra alto significa un equipo integrado y efectivo sin problemas de comunicaciones.

4.5 Madurez del proceso.- Refleja la madurez del proceso de la organización. El cálculo de este valor depende de un cuestionario de madurez del CMM. También se puede alcanzar una estimación a partir de las características exigidas en cada nivel de CMM y de acuerdo al grado en que se encuentra la organización.

En la tabla 4.1 se presentan los valores que corresponden a cada calificación de los factores de escala.

5. Multiplicadores de esfuerzo

Los multiplicadores de esfuerzo son usados para precisar la estimación inicial de esfuerzo. Se clasifican en cuatro grandes grupos y cada multiplicador es definido por un conjunto de niveles en los cuales se establecen determinados valores para la estimación final:

- **Producto:** se refiere a los factores del producto considerados en la variación del esfuerzo requerido para desarrollar el software, y que es causado por las características del producto mismo. Hay 5 factores de producto y su complejidad tiene una fuerte influencia en la estimación del esfuerzo.
- **Plataforma:** se refieren a las restricciones impuestas por el software y hardware.
- **Personal:** se refieren a los factores que toman en cuenta la experiencia y las capacidades de las personas que trabajan en el proyecto. Los valores asignados pueden cambiar durante el desarrollo del proyecto.
- **Proyecto:** se refieren a las características particulares del proyecto de desarrollo de software.

En las siguientes secciones se explican los multiplicadores de esfuerzo para cada una de las clases y en la tabla 5.2 se presentan los valores correspondientes para cada uno de los multiplicadores.

5.1 Multiplicadores relacionados al producto

- **RELY (Required Software Reliability – Fiabilidad del requerimiento de software):** Fiabilidad requerida del software al ejecutar sus funciones sobre un periodo de tiempo. Si el efecto de una falla en las funciones del software es poca, entonces el valor de RELY es muy bajo, pero si la falla pone en riesgo la vida humana, entonces el RELY es muy alto.
- **DATA (Database Size – Tamaño de base de datos):** En el desarrollo de un producto software, el tamaño de la base de datos utilizada es muy importante debido al esfuerzo necesario para generar los datos de prueba que serán usados para evaluar el programa. El valor del multiplicador se obtiene usando la relación D/P (D =tamaño de la base de datos en bytes y P = líneas de código fuente). Si el resultado de la fracción es menor a 10 entonces significa un valor bajo (una aplicación con poco uso de base de datos), pero si el resultado es mayor o igual a 1000 significa un valor muy alto (usa una gran base de datos).
- **CLPX (Product Complexity – Complejidad del producto):** La complejidad de los módulos del sistema está dividida en cinco áreas: operaciones de control, operaciones computacionales, operaciones dependientes del dispositivo, operaciones de gestión de datos y operaciones de gestión de interfaz de usuario. En la tabla 5.1 se puede seleccionar el área, determinar el nivel de complejidad y obtener el valor adecuado del multiplicador.

Niveles de complejidad por área					
Valor	Operaciones de control	Operaciones computacionales	Operaciones dependientes del dispositivo	Operaciones de gestión de datos	Operaciones de gestión de interfaz de usuario
Muy bajo	Líneas de código con algunos operadores de programación estructurada sin anidamiento: DOs, CASEs, IF-THEN-ELSEs. Simples composiciones de módulos con llamadas a procedimientos o simples scripts.	Evaluación de expresiones simples. Por ejemplo: $A=B+C*(D+E)$	Lectura y escritura de sentencias simple formatos sencillos.	Arreglos simples en memoria principal. Consultas o actualizaciones de componentes de bases de datos.	Formularios simples de entrada, generadores de reportes.
Bajo	Anidamientos sencillos con operadores de programación estructurada.	Evaluación de expresiones con un nivel moderado. Por ejemplo: $D=\text{SQRT}(B**2-4*A*C)$	No necesita conocimiento del procesador en particular o de las características de Entrada/Salida. Las entradas y salidas hechas a un nivel de GET/PUT.	Fijar archivos simples sin cambios en la estructura de datos, sin edición, sin archivos intermedios. Consultas y actualizaciones moderadas de componentes de bases de datos (COTS-BD).	Constructores simples de interfaces gráficas de usuarios.
Nominal	Anidamientos simples principal-mente. Algunos módulos de control intermedios. Tablas de decisión. Llamadas o mensajes simples, incluyendo, medianamente, el soporte de procesamientos distribuidos.	Uso de rutinas matemáticas y estadísticas. Operaciones básicas de matrices y vectores.	Procesos de Entrada/Salida que incluye selección de dispositivos, estados, revisión y procesamiento de errores.	Múltiples archivos de entrada y un solo archivo de salida. Cambios estructurales simples, ediciones simples. Consultas y actualizaciones complejas de COTS-BD.	Simple uso de un conjunto de elementos de interfaz gráfica de usuario.
Alto	Operadores de programación estructurada con expresiones compuestas altamente anidados. Control de cola y pila. Procesamientos homogéneos y distribuidos. Control flexible en tiempo real de un procesador.	Análisis numérico básico: interpolación multivariada, ecuaciones diferenciales ordinarios. Redondeos y truncamientos simples	Operaciones en el nivel de entrada y/o salida física (translaciones de direcciones de almacenamiento físico, búsquedas, lecturas, etc.). Traslapado de Entradas/Salidas optimizadas.	Disparadores (triggers) simples activados por cadenas de datos. Reestructuración de data compleja.	Uso intenso y ampliación de elementos de interfaz gráfica de usuario. Uso simple de multimedia y E/S de sonido.
Muy alto	Código recursivo y reentrante. Manejo de interrupciones con prioridad definida. Sincronización de tareas, llamadas complejas, procesamientos distribuidos complejos. Control estricto en tiempo real de un procesador.	Análisis numérico difícil y estructurado: ecuaciones de matrices singulares, ecuaciones diferenciales parciales. Paralelización simple.	Rutinas para el diagnóstico de interrupciones, servicio de interrupciones y enmascaramiento de interrupciones. Manejo de líneas de comunicación. Rendimiento de los sistemas embebidos.	Coordinación de base de datos distribuida. Disparadores (triggers) complejos. Optimización de búsquedas.	Uso de 2D/3D, gráficos dinámicos, multimedia con moderada complejidad.
Extra alto	Planificación de múltiples recursos con prioridades dinámicamente cambiantes. Control del nivel de micro código. Control estricto en tiempo real distribuido.	Análisis numérico difícil y no estructurado: Análisis exacto del ruido, datos estocásticos. Paralelización compleja.	Codificación y operaciones micro-programadas de dispositivos dependientes del tiempo. Sistemas embebidos de ejecución crítica.	Estructuras de objetos dinámicamente relacionados y altamente aclopadados. Gestión de datos en lenguaje natural.	Sistemas multimedia complejos de, realidad virtual, interfaz de lenguaje natural

Tabla 5.1 Componentes para niveles de clasificación de complejidad

- RUSE (Developed for Reusability – Desarrollo para reusabilidad): Evalúa el esfuerzo necesario para construir componentes que podrían ser reusados en proyectos futuros. El esfuerzo se presenta en la creación de diseños genéricos de software, elaboración de documentos y pruebas completas para garantizar que los componentes puedan ser usados en otras aplicaciones. El desarrollo para reusabilidad impone al proyecto restricciones en la clasificación de los multiplicadores RELY y DOCU.

Si el nivel de reusabilidad esperado es muy bajo entonces el valor de RUSE es bajo. Si se reutiliza a nivel de proyecto el valor es nominal, en cambio si se reutiliza dentro de múltiples líneas de producto el valor es extra alto.

- DOCU (Documentation Match to Life Cycle Needs – Documentación de las necesidades del ciclo de vida): Muchos de los modelos de costo del software consideran al nivel de documentación requerida como un factor de costo. La clasificación de costos para DOCU es evaluada en términos de conformidad con la documentación necesaria para el proyecto. Un valor muy alto significa que todas las necesidades son cubiertas (documentación muy exhaustiva) y un valor muy bajo significa que muchas necesidades no están cubiertas (se deja de documentar varias fases del ciclo de vida).

5.2 Multiplicadores relacionados a la plataforma

- TIME (Execution Time Constraint – Restricción del tiempo de ejecución): La medida del tiempo de ejecución es una restricción impuesta en un sistema software. La clasificación de los valores son expresados en términos del porcentaje esperado del tiempo de ejecución disponible a ser usado por el sistema o subsistema, utilizando como recurso el tiempo de ejecución. Un valor nominal significa que el recurso tiempo de ejecución es empleado en un 50% o menos, un valor extra alto significa que el software emplea un 95% del tiempo de ejecución (uso intensivo del computador en cálculos o procesos).
- STOR (Main Storage Constraint – Restricciones de la memoria principal): La clasificación representa el grado de almacenamiento de la memoria principal como una restricción en el sistema o subsistema software. Un valor nominal significa que menos del 50% del almacenamiento disponible (capacidad de memoria) es utilizado, en cambio si el nivel de uso es de 95% o más el valor es extra alto.
- PVOL (Platform Volatility – Volatilidad de la plataforma de desarrollo): Refleja la complejidad del hardware y software en la performance de las tareas del sistema software. Si el software a ser desarrollado es un sistema operativo entonces la plataforma es el hardware, si es un sistema administrador de base de datos entonces la plataforma es el hardware y el sistema operativo, si se trata de un sistema web entonces la plataforma es la red, el hardware, el sistema operativo y los repositorios de información distribuidos. Un valor bajo significa que un cambio drástico de la plataforma se produce cada 12 meses y cambios menores cada mes. Un valor muy alto implica que los cambios drásticos se dan cada 2 semanas y los menores cada dos días.

5.3 Multiplicadores relacionados al personal

- ACAP (Analyst Capability – Capacidad de los analistas): Se refiere a la capacidad de análisis del equipo del proyecto, quienes trabajan en requerimientos, diseños de alto nivel y detallados. Los mayores atributos que deben ser considerados en esta clasificación son la habilidad de análisis y diseño, eficiencia y perfección, y la habilidad para comunicarse y cooperar. Los valores no deben considerar el nivel de experiencia del analista que será evaluado en AEXP, LTEX y PEXP.

Un valor muy bajo significa que el equipo de analistas tiene un percentil de 15 (una capacidad de análisis muy bajo como equipo) y un valor muy alto significa que el equipo de analistas tiene un percentil de 90 (una gran capacidad de análisis como equipo).

- AEXP (Applications Experience – Experiencia en aplicaciones): La clasificación de este factor depende del nivel de experiencia, del equipo del proyecto, desarrollando aplicaciones de sistemas o subsistemas de software. Un valor muy bajo significa un nivel de experiencia menor a 2 años y un valor muy alto significa una experiencia de 6 a más años.

Criterios de Aplicación							
CC(xi)	Unidades	Muy Bajo	Bajo	Nominal	Alto	Muy Alto	Extra Alto
PRODUCTO							
RELY	Efecto de un fallo	Ligeros inconvenientes 0.82	Pequeñas pérdidas fáciles de recuperar 0.92	Moderadas pérdidas recuperables 1.00	Altos costos económicos 1.10	Riesgo de Vidas humanas 1.26	
DATA	Bytes/LCF		D/P<10 0.90	10<=D/P<100 1.00	100<=D/P<1000 1.14	D/P>1000 1.28	
CLPX		0.73	0.87	1.00	1.17	1.34	1.74
RUSE			Ninguno 0.95	Proyecto 1.00	Programa de proyectos 1.07	Línea de productos 1.15	Múltiples líneas de producto 1.24
DOCU		Muchas fases del ciclo de vida sin documentar 0.81	Algunas Fases del ciclo de vida sin documentar 0.91	Documentos adecuados 1.00	Documentos exhaustivos 1.11	Documentos muy exhaustivos 1.23	
PLATAFORMA							
TIME	Disponibilidad			<=50% 1.00	<=70% 1.11	<=85% 1.29	<=95% 1.63
STOR	Disponibilidad			<=50% 1.00	<=70% 1.05	<=85% 1.17	<=95% 1.46
PVOL	Días		M360-m30 0.87	M180-m15 1.00	M60-m7 1.15	M15-m2 1.30	
PERSONAL							
ACAP	Percentil	15 1.42	35 1.19	55 1.00	75 0.85	90 0.71	
AEXP	Meses	<=2 1.22	6 1.10	12 1.00	36 0.88	72 0.81	
PCAP	Percentil	15 1.34	35 1.15	55 1.00	75 0.88	90 0.76	
PEXP	Meses	<=2 1.19	6 1.09	12 1.00	36 0.91	72 0.85	
LTEX	Meses	2 1.20	6 1.09	12 1.00	36 0.91	72 0.84	
PCON	Cambios Anuales	48% 1.29	24% 1.12	12% 1.00	6% 0.90	3% 0.81	
PROYECTO							
TOOL	Utilización	Ninguna 1.17	Empezando 1.09	Alguna 1.00	General 0.90	Imprescindible 0.78	
SITE	Situación	Internacional 1.22	Multi ciudad y Multi empresa 1.09	Multi ciudad o Multi empresa 1.00	Misma Zona 0.93	Mismo Edificio 0.89	Mismo área 0.80
SITE	Comunicaciones	Teléfono/correo 1.22	Teléfono Individual / Fax 1.09	e-mail 1.00	Banda ancha 0.93	Video conferencia 0.86	Multi media interactiva 0.80
SCED	Disponibilidad	75% 1.43	85% 1.14	100% 1.00	130% 1.00	160% 1.00	

Tabla 5.2 Valores de los multiplicadores de esfuerzo

- PCAP (Programmer Capability - Capacidad del programador): Se refiere a la evaluación de la capacidad de los programadores como equipo, no en forma individual. Los factores que pueden ser considerados son la habilidad, eficiencia, minuciosidad, y la habilidad para comunicarse y cooperar. La experiencia del programador no debe ser considerada.

Un valor muy bajo significa que el equipo de programadores está en un percentil de 15%, esto quiere decir que el equipo tiene poca capacidad de programación. Un valor muy alto significa que el equipo de programadores tiene un percentil de 90, que se traduce como un equipo que tiene una gran capacidad de programación..

- PEXP (Platform Experience – Experiencia en la plataforma): Se refiere a la experiencia del programador en el uso de plataformas (interfaz de usuario, gráficos, base de datos, red, etc.) en el desarrollo del proyecto. Un valor muy bajo significa que la experiencia del equipo es menor o igual que 2 meses. Un valor muy alto significa que el tiempo de experiencia del equipo es de 6 años o más.
- LTEX (Language and Tool Experience - Experiencia en el lenguaje y las herramientas). Se refiere a la medición del nivel de experiencia en la herramienta de software y el lenguaje de programación del equipo del proyecto que desarrolla el sistema o subsistema de software. El desarrollo del software incluye el uso de herramientas para: implementación de requerimientos, representación del análisis y diseño, gestión de configuración, documentación, gestión de librerías, estilo y formato de programas, verificación, planeamiento, control, etc.

Un valor muy bajo significa que la experiencia del equipo es menor que 2 meses y un valor muy alto significa que la experiencia del equipo es de 6 años o más.

- PCON (Personnel Continuity - Continuidad del personal): El valor se refiere a la continuidad anual del personal del proyecto. Un valor muy alto significa un percentil de 3% (que el personal tiene mayor continuidad que sería lo mismo que una baja rotación) y un valor muy bajo significa un percentil de 48% que indica una baja continuidad (o alta rotación).

5.4 Multiplicadores relacionados al proyecto

- TOOL (Use of Software Tools - Utilización de las herramientas de software): Indica las facilidades de la herramienta, tales como la codificación, seguimiento, integración con otros procesos, etc. Un valor muy bajo significa que no utilizan herramienta y un valor muy alto significa que el uso de la herramienta es imprescindible.
- SITE (Multisite Development – Múltiples lugares para el desarrollo): Se refiere a la amplitud del trabajo en múltiples lugares y la calidad de las comunicaciones. Si tiene una colocación internacional y la comunicación es por teléfono o correo entonces el valor asignado es muy bajo pero si su colocación es en la misma área y la comunicación es multimedia interactiva entonces su valor es extra alto.
- SCED (Required Development Schedule – Planificación de desarrollo con tiempos comprometidos): Representa el nivel de compresión o expansión al que es sometido un proyecto de desarrollo de software. Una compresión obliga esfuerzos adicionales en la definición de la arquitectura y la eliminación de riesgos en las primeras etapas, mientras que en las etapas posteriores se requiere paralelizar actividades como pruebas y documentación. Un valor de compresión de 75% es considerado muy bajo y uno de ampliación de 160% tiene un valor muy alto.

6. Duración del Proyecto

Los administradores de proyecto también necesitan estimar el tiempo que durará el desarrollo software. La relación entre el esfuerzo total requerido en el proyecto, el número de personas y el tiempo de desarrollo no es proporcional por lo tanto el incrementar el número de personas no significa que la duración del proyecto será menor.

Estimación del tiempo presentada en COCOMOII es la siguiente

$$TDEV = [C \times \text{Esfuerzo}^F] \times [\text{SCED}\% / 100] \quad \text{Ec. 6.1}$$

Donde: $F = D + 0.2 * (E - B)$ Ec. 6.2

los valores son:

Símbolo	Descripción
B	Base del exponente de escalamiento, calibrado en 0.91.
C	Coficiente de planificación, calibrado en 3.67.
D	Exponente de base de escalamiento para planificación, calibrado en 0.28.
E	Exponente de escalamiento de tipo de proyecto, utilizado en la ecuación de esfuerzo.
SCED%	Porcentaje de planificación de desarrollo con tiempos comprometidos
F	Exponente de escalamiento para planificación.

La fórmula de estimación del tiempo incluida por COCOMOII puede ser utilizada para todos los niveles de las actividades en el proceso de software identificados por el modelo.

7. Ejemplo de Aplicación

A continuación presentamos un ejemplo desarrollado y que podrá completar en la separata con las indicaciones dadas en cada etapa. Lea cuidadosamente el caso antes de pasar a la solución. El ejemplo ha sido tomado y adaptado del Libro de Gestión del Proceso Software [Cuevas, 287].

Una organización de desarrollo de software va a producir una aplicación de gestión de 20 KLCF. Dicha organización sólo desarrolla sistemas de ese tipo, de los cuales ha producido ya muchos productos que están funcionando. Se debe tener en cuenta que el producto debe satisfacer unos requisitos fijos, así como unas interfaces externas que también están previamente fijadas.

La organización tiene un alto nivel de madurez en sus procesos de desarrollo, pero para desarrollar este producto ha reunido un equipo que mezcla personas que llevan tiempo trabajando juntas con otras recién llegadas, lo que da una cohesión del equipo básica.

En cuanto al tratamiento en la solución de riesgos, la cultura de la empresa es procurar la máxima calidad y realizar una revisión completa.

Entre los requisitos para el sistema anterior, se sabe que deben extremarse los controles de calidad, ya que un fallo de funcionamiento, acarrearía grandes pérdidas económicas. También se sabe que la documentación deberá ser exhaustiva porque será utilizada más adelante para proyectos similares. En todos los aspectos, la filosofía de construcción está basada en la reutilización y está implantada en todas las líneas de producto de la empresa.

Se sabe también que las personas recién llegadas al equipo, son programadores con baja experiencia de trabajo y baja experiencia en el lenguaje y en la herramienta, mientras que los

miembros más veteranos son analistas con muy alta capacidad y muy alta experiencia en el lenguaje, así como con la herramienta.

El proyecto se va a desarrollar simultáneamente en varias ciudades de distintos países, pero los miembros del mismo van a estar conectados permanentemente por un sistema multimedia interactivo que aplicará la más alta tecnología.

Calcular el esfuerzo requerido para desarrollar el sistema usando el modelo COCOMOII en la etapa de Post-Arquitectura.

8. Solución del ejemplo

Determinación de los factores de escala.

Para resolver el problema primero se debe determinar los factores de escala del proyecto. Para esta etapa se debe analizar cuidadosamente el factor con respecto al escenario descrito.

En el siguiente recuadro se ha realizado el análisis correspondiente y se ha determinado la calificación correspondiente a cada factor. Complete los valores numéricos usando la tabla 4.1 y las calificaciones obtenidas para cada factor.

- **Precedentes:** dicha organización sólo desarrolla sistemas de ese tipo, de los cuales ha producido ya muchos productos que están funcionando. Por lo tanto se debe seleccionar el valor Muy Alto. Valor:_____
- **Flexibilidad de desarrollo:** se debe tener en cuenta que el producto debe de satisfacer unos requisitos fijos, así como unas interfaces externas que también están previamente fijadas. Por lo tanto se debe seleccionar el valor Muy Bajo. Valor:_____
- **Arquitectura /Solución de riesgo:** en cuanto al tratamiento en la solución de riesgos, la cultura de la empresa es de procurar la máxima calidad y realizar una revisión completa. Por lo tanto, se debe seleccionar el valor Muy Alto. Valor:_____
- **Cohesión del equipo / interacción:** la organización tiene un alto nivel de madurez en sus procesos de desarrollo, pero para desarrollar este producto ha reunido un equipo que mezcla personas que llevan tiempo trabajando juntas con otras recién llegadas, lo que da una cohesión del equipo básica. Por lo tanto, se debe seleccionar el valor Nominal. Valor:_____
- **Madurez de proceso:** la organización tiene un alto nivel de madurez en sus procesos de desarrollo. Por lo tanto, se debe seleccionar el valor Muy Alto. Valor:_____

Determinación de los multiplicadores de esfuerzo

Ahora se determinará los multiplicadores de esfuerzo para el proyecto de acuerdo a la tabla 5.2. Para aquellos criterios para los que no se tiene descripción se ha seleccionado el valor nominal:

- PRODUCTO**
- **RELY:** entre los requisitos para el sistema anterior, se sabe que deben extremarse los controles de calidad, ya que un fallo de funcionamiento, acarrearía grandes pérdidas económicas. Por lo tanto, se debe seleccionar el valor Alto. Valor:_____
 - **DATA:** la aplicación es de 20 KLCF. Consideramos un valor nominal entre 10 y 100 Bytes (Base de datos) / LCF (Líneas de Código Fuente). Valor:_____
 - **RUSE:** en todos los aspectos, la filosofía de construcción basada en la reutilización está implantada en todas las líneas de producto de la empresa. Por lo tanto, se debe seleccionar el valor Extra Alto. Valor:_____
 - **DOCU:** también se sabe que la documentación deberá ser exhaustiva porque será utilizada más adelante para proyectos similares. Por lo tanto, se debe seleccionar el valor Muy Alto. Valor:_____

PLATAFORMA

- **PVOL:** este multiplicador indica la volatilidad de la plataforma en función de los cambios que se pueden prever. Se distinguen cambios (M-Mayores) que afectan a la aplicación u otros elementos esenciales y cambios (m-menores) que afecten a los elementos auxiliares. Como no se indica nada en el enunciado, seleccionamos Nominal por posibles cambios Mayores cada 180 días (6 meses) y menores cada 15 días. Valor: _____

PERSONAL

- **ACAP y PCAP:** no se dispone de información respecto a la aptitud del conjunto de análisis y programadores. Se considera el Nominal con una buena aptitud en el percentil 55% del personal. Valor: _____
- **AEXP:** mientras que los miembros más veteranos son analistas con muy alta capacidad. Por lo tanto, se debe seleccionar el valor Muy Alto. Valor: _____
- **PEXP:** se sabe también que las personas recién llegadas al equipo, son programadores con baja experiencia de trabajo. Por lo tanto, se debe seleccionar el valor Muy Bajo. Valor: _____
- **PCON:** que mezcla personas que llevan tiempo trabajando juntas con otras recién llegadas, lo que da una cohesión del equipo básica. Por lo tanto, se debe seleccionar el valor Muy Bajo. Valor: _____

PROYECTO

- **TOOL:** se utilizarán herramientas para el desarrollo de software. Por lo tanto, se debe seleccionar el valor Alto. Valor: _____
- **SITE:** el proyecto se va a desarrollar simultáneamente en varias ciudades de distintos países, pero los miembros del mismo van a estar conectados permanentemente por un sistema multimedia interactivo que aplicará la más alta tecnología. Por lo tanto, se debe seleccionar el valor Muy Bajo en cuanto a situación y Extra Alto en cuanto a comunicaciones por tanto nos quedamos como valor global el Nominal. Valor: _____

En los recuadros que aparecen a continuación, se han colocado los valores correspondientes y que deben coincidir con los valores que usted obtuvo en los pasos anteriores.

Factores de escala

Criterios de aplicación	
Factor de escala (yi)	Valor
Precedentes	1.24
Flexibilidad de desarrollo	5.07
Arquitectura / Solución de riesgo	1.41
Cohesión del equipo / interacción	3.29
Madurez del proceso	1.56
Total (Sumatoria)	12.57

Multiplicadores de esfuerzo

Criterios de aplicación	
Multiplicador de esfuerzo (xi)	Valor
Criterio de costo PRODUCTO	
RELY	1.10
DATA	1.00
CLPX	1.00
RUSE	1.24
DOCU	1.23
Criterio de costo PLATAFORMA	
TIME	1.00
STOR	1.00
PVOL	1.00
Criterio de costo PERSONAL	
ACAP	1.00
AEXP	0.81
PCAP	1.00
PEXP	1.19
LTEX	1.00
PCON	1.29
Criterio de costo PROYECTO	
TOOL	0.90
SITE	1.00
SCED	1.00
Total (Productoria)	1.8775

Aplicando las fórmulas Ec 3.1 y Ec 3.2 se tiene:

$$\Sigma \text{ Factores} = 12.57$$

$$\Pi \text{ Multiplicadores} = 1.8775$$

$$E = 0.91 + 0.01 * (\Sigma \text{ Factores}) = 0.91 + 0.01 * 12.57 = 1.0357$$

$$\begin{aligned} \text{Esfuerzo} = e &= 2.94 * (\text{KLFCF})^E (\Pi \text{ Multiplicadores}) \\ &= 2.94 * (20)^{1.0357} \{ 1.8775 \} = 122.85 \end{aligned}$$

Luego, el esfuerzo del proyecto es de: 123 meses-persona.

Para el cálculo de la duración del proyecto utilizamos las fórmulas Ec 6.1 y Ec. 6.2 :

$$F = 0.28 + 0.2 * (1.0357 - 0.91) = 0.30514$$

$$\text{TDEV} = (3.67 * (122.85)^{0.30514}) * (1.00) = 15.93$$

Luego, el tiempo requerido para completar el proyecto es 16 meses.

9. Referencias Bibliográficas

[COCOMO II]: Barry W. Boehm, Software cost estimation with COCOMO II, Prentice Hall PTR, 2000.

[Sommerville]: Ian Sommerville, Ingeniería de software, Addison Wesley, 6ta Edición, 2002.
COCOMO II Model Definition.

[Cuevas]: Gonzalo Cuevas, "Gestión del proceso software".

<http://sunset.usc.edu/research/COCOMOII/>