

Estimación de Recursos

Alberto Cueto – ECOS

Recursos del Proyecto



Fuente: Pressman

Alberto Cueto – ECOS

Plan de Gerencia del Recurso

- Para cada recurso
 - Descripción del recurso
 - Disponibilidad
 - Fechas cuándo será requerido
 - Duración de uso del recurso

Alberto Cuelo – ECOS

Estimación del Costo y Esfuerzo

- Historia – Proyectos anteriores similares que se han terminado
- Descomposición – Descomponer el proyecto en unidades simples
- Métodos empíricos para la estimación

Alberto Cuelo – ECOS

Recursos de SW reutilizables

- Para ser utilizados eficientemente, deben ser catalogados, estandarizados y validados
- Cuatro categorías
 - Componentes Comerciales (COTS)
 - Componentes maduros
 - Especificaciones, diseños, código o datos de pruebas existentes y desarrollados para proyectos anteriores similares a este. Requieren poca modificación
 - Componentes semi-maduros
 - Especificaciones, diseños, código o datos de pruebas existentes y desarrollados para proyectos anteriores relacionados a este pero que requieren mucha modificación
 - Nuevos componentes
 - Componentes desarrollados para satisfacer necesidades específicas de este proyecto

Alberto Cueto – ECOS

Cocomo II

- COnstructive COst MOdel
- Modelo de estimación de costos. Tres modelos:
 - *Application composition model*: Etapas iniciales del SW (prototipos de interfaces de usuarios, análisis de interacción del SW y del sistema, evaluación de desempeño)
 - *Early design stage model*: Utilizado cuando los requerimientos y la arquitectura están estabilizados.
 - *Post-architecture-stage model*: Utilizado durante la construcción del software.

Alberto Cueto – ECOS

COCOMO - 81 COCOMO II

Alberto Cueto – ECOS

COCOMO - 81

Alberto Cueto – ECOS

Estimación de Costos

- **CON**structive **CO**st **MO**del **COCOMO**
desarrollado por Barry Boehm - 81
- Inicialmente para desarrollos siguiendo el ciclo de vida en cascada

Alberto Cuelto – ECOS

COCOMO

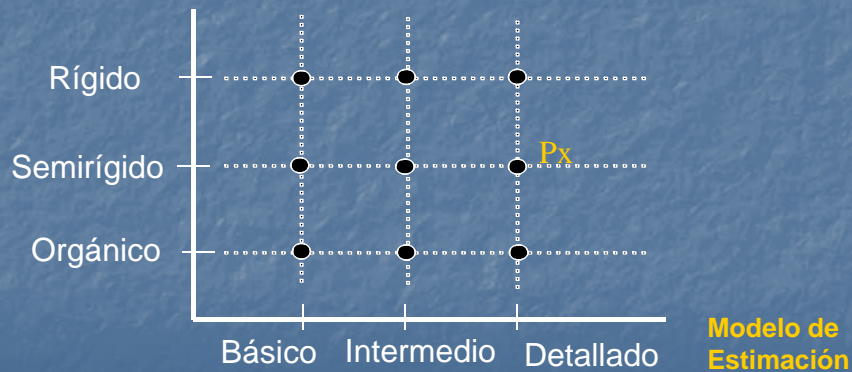
- El modelo original se basó en un estudio sobre 63 proyectos.
- Considera tres tipos de proyectos o modos de desarrollo del software:
 - Orgánico (*Organic*)
 - Semilibre (*Semi-detached*)
 - Rígido (*Embedded*)
- Tres métodos de estimación distintos:
 - Básico
 - Intermedio
 - Detallado

Alberto Cuelto – ECOS

Espacio de Proyectos

Primer paso: Situar el proyecto en el siguiente espacio:

Tipo de Proyecto



Alberto Cueto – ECOS

Ecuaciones Fundamentales

- Para todos los tipos de modelos y métodos de estimación, el modelo de cálculo, dado por dos ecuaciones fundamentales, es igual:

- Esfuerzo = $C1 * (\text{Tamaño})^{P1}$
- Tiempo = $C2 * (\text{Esfuerzo})^{P2}$

Alberto Cueto – ECOS

Parámetros Ecuaciones Fundamentales

- **Esfuerzo** = número de meses-hombre
- **C1** = constante que escala el esfuerzo
- **Tamaño** = tamaño del producto final
- **P1** = un exponente que caracteriza la economía de escala inherente al proceso usado para producir el producto final
- **Tiempo** = número total de meses
- **C2** = constante que escala el tiempo
- **P2** = un exponente que caracteriza la inercia y el paralelismo inherente en los proyectos de desarrollo

Alberto Cueto – ECOS

Tipo de Proyecto - Orgánico

- Equipos de trabajo relativamente pequeños
- Desarrollo en la propia empresa
- Personal con amplia experiencia en proyectos similares
- Todas las personas conocen el impacto del sistema sobre la organización
- Sobrecarga pequeña de comunicación
- Facilidad de usuario a acomodarse a posibles diferencias entre especificaciones y el producto
- Entorno de desarrollo estable
- Sin motivación para una rápida terminación
- Aplicaciones de tamaño pequeño (50 KDSI)

Alberto Cueto – ECOS

Tipo de Proyecto - Semi-libre

- Personal con nivel de experiencia intermedio en proyectos similares
- Equipo mixto de gente experta e inexperta
- Algunos miembros del equipo con experiencia en aspectos del proyecto y otros no
- Tipo de proyecto representativo: proyectos con interfaces muy rigurosas y otras muy flexibles
- Tamaño máximo 300 KDSI

Alberto Cueto – ECOS

Tipo de Proyecto - Rígido

- Proyectos que se desarrollan dentro de unas limitaciones muy estrictas
- El producto se explota dentro de un entorno muy restringido de HW, SW y procedimientos operativos
 - Ej. Transferencias de fondos o control de tráfico aéreo
- Alto costo de modificaciones
- No admite cambios en requerimientos o interfaces
- Proyectos en áreas poco conocidas
 - Equipos pequeños de analistas
 - Alta sobrecarga de comunicaciones
- Para proyectos de cualquier tamaño

Alberto Cueto – ECOS

Modelo de Estimación - Básico

- Productos Pequeños/Medios
- Desarrollados por personal de la propia empresa
- Generalmente en modo orgánico
 - Orgánico: $\text{Esfuerzo} = 2,4 * (\text{KDSI})^{1,05}$
 (*Organic*) $\text{Tiempo} = 2,5 * (\text{Esfuerzo})^{0,38}$
 - Semilibre: $\text{Esfuerzo} = 3,0 * (\text{KDSI})^{1,12}$
 (*Semi-detached*) $\text{Tiempo} = 2,5 * (\text{Esfuerzo})^{0,35}$
 - Rígido: $\text{Esfuerzo} = 3,6 * (\text{KDSI})^{1,2}$
 (*Embedded*) $\text{Tiempo} = 2,5 * (\text{Esfuerzo})^{0,32}$

Alberto Cueto – ECOS

Modelo de Estimación - Básico - Ejemplo

- Desarrollo de un sistema con un tamaño estimado de 32.000 LOC
- Modo Orgánico
 - $\text{Esfuerzo} = 2,4 * (32)^{1,05} = 91$ meses-hombre
 - $\text{Tiempo} = 2,5 * (91)^{0,38} = 14$ meses
 - N° medio de personas = $91/14 = 6,5$ personas

Alberto Cueto – ECOS

Modelo de Estimación - Básico - Limitaciones

- No tiene en cuenta distintos factores que pueden tener influencia en el costo del producto y en el posterior mantenimiento del mismo
 - Experiencia del equipo de trabajo
 - Experiencia de los programadores
 - Herramientas y métodos utilizados

Alberto Cueto – ECOS

Modelo de Estimación - Intermedio

- Para calcular el esfuerzo, incorpora 15 variables de predicción que influyen en el costo del proyecto
- Estimación de **Esfuerzo Nominal**
 - Orgánico: $\text{Esfuerzo} = 2,4 * (\text{KDSI})^{1,05}$
 - Semilibre: $\text{Esfuerzo} = 3,0 * (\text{KDSI})^{1,12}$
 - Rígido: $\text{Esfuerzo} = 3,0 * (\text{KDSI})^{1,2}$
- **Esfuerzo real** = Esfuerzo Nominal multiplicado por el producto de los factores de ajuste
- La Duración se calcula como en el Modelo Básico

Alberto Cueto – ECOS

M.E. - Intermedio - Factores de Ajuste

- Atributos del producto de Software
 - RELY, DATA, CPLX
- Atributos del Computador
 - TIME, STOR, VITR, TURN
- Atributos de Personal
 - ACAP, AEXP, PCAP, VEXP, LEXP
- Atributos del Proyecto
 - MODP, TOOL, SCED

Alberto Cueto – ECOS

Multiplicadores de Esfuerzo

VARIABLE	Muy Bajo	Bajo	Nominal	Alto	Muy Alto	Extra Alto
RELY	0.75	0.88	1.0	1.15	1.40	
DATA		0.94	1.0	1.08	1.16	
CPLX	0.70	0.85	1.0	1.15	1.30	1.65
TIME			1.0	1.11	1.30	1.66
STOR			1.0	1.06	1.21	1.56
VITR		0.87	1.0	1.15	1.30	
TURN		0.87	1.0	1.07	1.15	
ACAP	1.46	1.19	1.0	0.86	0.71	
AEXP	1.29	1.13	1.0	0.91	0.82	
PCAP	1.42	1.17	1.0	0.86	0.70	
VEXP	1.21	1.10	1.0	0.90		
LEXP	1.14	1.07	1.0	0.95		
MODP	1.24	1.10	1.0	0.91	0.82	
TOOL	1.24	1.10	1.0	0.91	0.83	
SCED	1.23	1.08	1.0	1.04	1.10	

Alberto Cueto – ECOS

Parámetros de Ajuste - Software

■ Atributos del Producto

- RELY : Fiabilidad requerida del SW
 - **Fiabilidad**: Probabilidad de que el SW realice sus funciones satisfactoriamente en su próxima ejecución, en un periodo de tiempo dado
 - **Bajo**: El SW se corrige sin otra consecuencia
 - **Muy alto**: Posibles pérdidas de vidas humanas
- DATA : Tamaño de la Base de Datos
 - **DATA** = Tamaño BD en caracteres / LOC
 - **Bajo**: 0-10; **Nominal**: 10-100; **Alto**: 100-1000; **Muy alto**: >1000
- CPLX : Complejidad del Producto
 - Funciones de control, cálculos, gestión de datos y operaciones dependientes de dispositivos
 - Funciones matemáticas simples (**bajo**) a ejecución dinámica (**Muy alto**)

Alberto Cueto – ECOS

Parámetros de Ajuste - Computador

■ Atributos del computador

- TIME : Limitaciones en el Tiempo de Ejecución
 - % tiempo uso producto / % uso otros productos
 - **Nominal**: 50%; **Extra alto**: 95%
- STOR : Limitaciones de Memoria Principal
 - % uso memoria por producto / % uso memoria otros sistemas
 - **Nominal**: > 50%; **Extra alto**: 95%
- VIRT : Volatilidad de la Máquina Virtual
 - Máquina virtual: Conjunto de HW + SW que usa el producto
- TURN : Tiempo de Respuesta del equipo de desarrollo
 - Tiempo de respuesta en horas desde que se inicia la ejecución de un trabajo hasta que se obtienen los resultados del proceso
 - **Bajo**: Sistema interactivo; **Muy alto**: tiempo respuesta > 12 horas

Alberto Cueto – ECOS

Parámetros de Ajuste - Personal

■ Atributos del personal

- ACAP : Capacitación de los analistas
 - Eficiencia para trabajar en equipo: Habilidad para el análisis, eficiencia y calidad en el trabajo, habilidad para comunicarse y cooperar
- AEXP : Experiencia en aplicaciones (del equipo de desarrollo)
 - **Muy bajo**: < 4 meses; **Muy alto**: > 12 años
- PCAP : Capacitación de los programadores
 - Similar a ACAP, pero para los programadores
- VEXP : Experiencia en la Máquina Virtual
 - **Muy bajo**: < 1 mes; **Alto**: > 3 años
- LEXP : Experiencia en el Lenguaje de Programación
 - **Muy bajo**: < 1 mes; **Alto**: > 3 años

Alberto Cueto – ECOS

Parámetros de Ajuste - Proyecto

■ Atributos del proyecto

- MODP : Prácticas Modernas de Programación
 - Análisis de requerimientos y diseño top-down; diseño estructurado; desarrollo incremental; revisiones de colegas; bibliotecas de programas
- TOOL : Uso de herramientas para el desarrollo del SW
 - **Muy bajo**: Herramientas básicas; **Muy alto**: CASE
- SCED : Limitaciones en la Planificación
 - Porcentaje de retraso o aceleración con respecto a la planificación nominal impuesta al equipo de desarrollo
 - **Muy bajo**: aceleración; **Muy alto**: retraso

Alberto Cueto – ECOS

Multiplicadores de Esfuerzo

VARIABLE	Muy Bajo	Bajo	Nominal	Alto	Muy Alto	Extra Alto
RELY	0.75	0.88	1.0	1.15	1.40	
DATA		0.94	1.0	1.08	1.16	
CPLX	0.70	0.85	1.0	1.15	1.30	1.65
TIME			1.0	1.11	1.30	1.66
STOR			1.0	1.06	1.21	1.56
VIRT		0.87	1.0	1.15	1.30	
TURN		0.87	1.0	1.07	1.15	
ACAP	1.46	1.19	1.0	0.86	0.71	
AEXP	1.29	1.13	1.0	0.91	0.82	
PCAP	1.42	1.17	1.0	0.86	0.70	
VEXP	1.21	1.10	1.0	0.90		
LEXP	1.14	1.07	1.0	0.95		
MODP	1.24	1.10	1.0	0.91	0.82	
TOOL	1.24	1.10	1.0	0.91	0.83	
SCED	1.23	1.08	1.0	1.04	1.10	

Alberto Cueto – ECOS

Modelo Intermedio - Ejemplo

- Por medio de contratación directa se negocia el desarrollo de un software complejo de 10 KDSI para un microprocesador comercial
- El software de comunicaciones genera necesidades de codificación de complejidad muy alta
- Se planea utilizar personal muy capacitado
- Cual es el esfuerzo y el costo del desarrollo, si el precio promedio hombre-mes es de US\$500?

Alberto Cueto – ECOS

Parámetros de Ajuste - Ejemplo

Factor	Descripción	Ratio	Multiplicador
RELY	USO LOCAL	NOMINAL	1.0
DATA	20.000 BYTES	BAJO	0.94
CPLX	COMUNICACIONES	MUY ALTO	1.30
TIME	SE USARÁ EL 70%	ALTO	1.11
STOR	45K/84K	ALTO	1.06
VIRT	HW COMERCIAL	NOMINAL	1.0
TURN	DOS HORAS MEDIA	NOMINAL	1.0
ACAP	BUENOS ANALISTAS	ALTO	0.86
AEXP	TRES AÑOS	NOMINAL	1.0
PCAP	BUENOS PROGRAMAD.	ALTO	0.86
VEXP	SEIS MESES	BAJO	1.10
LEXP	DOCE MESES	NOMINAL	1.0
MODP	MUCHAS TÉCNICAS	ALTO	0.91
TOOL	A NIVEL MICRO	BAJO	1.10
SCED	NUEVE MESES	NOMINAL	1.0

Factor de Ajuste: 1,17

Alberto Cueto – ECOS

Modelo Intermedio - Ejemplo

- De acuerdo al número de líneas (10 KDSI), se utiliza el Modo Orgánico
 - Esfuerzo Nominal = $3,2 * (10)^{1,05} = 36$ meses-hombre
 - Esfuerzo Ajustado = $36 * 1,17 = 42,12$ meses-hombre
 - Tiempo = $2,5 * (42,12)^{0,38} = 10,36$ meses
 - Costo Aplicación: $42,12 * US\$500 = US\$21,060$

Alberto Cueto – ECOS

Modelo de Estimación - Intermedio - Limitaciones

- La distribución del esfuerzo por fases puede ser inadecuada
- Puede ser engorroso utilizarlo en un producto con muchos componentes

Alberto Cueto – ECOS

Modelo de Estimación - Detallado

- Incluye dos características que permiten hacer la estimación más exacta
 - Multiplicadores de esfuerzo por Fases
 - Multiplicadores para cada atributo en cada fase de desarrollo
 - Jerarquización del producto a Tres Niveles
 - Nivel Sistema
 - Nivel Subsistema
 - Nivel Módulo

Alberto Cueto – ECOS

M.E. - Detallado - Jerarquización

- Los factores de ajuste se aplican a tres niveles:
- Nivel módulo
 - Se describe por el número de instrucciones (LOC) producidas y por aquellos factores que tienden a modificar dicho nivel: complejidad del módulo y adaptación a partir del SW existente y de la capacidad y experiencia de los programadores
- Nivel Subsistema
 - Se describe por los factores restantes (limitaciones en tiempo y memoria, capacidad de los analistas, herramientas, planificación, etc.) que tienden a variar de un subsistema a otro, pero que son iguales para todos los módulos dentro de un subsistema

Alberto Cueto – ECOS

M.E. - Detallado - Jerarquización (2)

- Nivel Sistema
 - Se define mediante los factores correspondientes al conjunto del proyecto, como son el esfuerzo nominal y la planificación de tiempos

Alberto Cueto – ECOS

Limitaciones COCOMO-81

- Para el modelo intermedio, el 68% de las veces los valores reales están dentro del 20% de los valores estimados
- Nueva generación de procesos y productos
 - Software comercial
 - Software orientado a objetos
 - Modelos evolutivos o en espiral
 - Composición de aplicaciones comerciales (COTS)

Alberto Cueto – ECOS

COCOMO II: Estrategia

- Continuidad con COCOMO
- Ajustar COCOMO II a los futuros sectores del mercado
- Ajustar las entradas y salidas de COCOMO II al nivel de información disponible

Alberto Cueto – ECOS

Referencias Bibliográficas

- Software Project Management. A Unified Framework. Walker Royce. Addison Wesley. 1998
- Managing the Software Process. Watts Humphrey. Addison Wesley 1989.

Alberto Cueto – ECOS

COCOMO II

Alberto Cueto – ECOS

Antecedentes

- Constructive Cost Model (COCOMO-81)
- Nuevas generaciones de procesos y productos de SW, está cambiando la forma de desarrollar SW
 - Procesos de SW colaborativos, guiados por el riesgo y evolutivos
 - 4GL y generadores de aplicaciones
 - Software comercial (COTS) y software reutilizable
 - Aproximaciones de desarrollo de SW de alta velocidad
 - Iniciativas de procesos de madurez

Alberto Cueto – ECOS

Objetivos

- Desarrollar un modelo de estimación de costos y esfuerzos, sintonizado con los ciclos de vida utilizados en las décadas de 1990 y 2000
- Desarrollar una base de datos de costos de desarrollo de SW y capacidades de soportes de herramientas, para mejora continua del modelo
- Proveer un marco de trabajo analítico y cuantitativo y un conjunto de herramientas y técnicas, para evaluar los efectos de mejoras en la tecnología de SW en los costos y esfuerzos del ciclo de vida

Alberto Cueto – ECOS

Estrategia

- Mantener la apertura de COCOMO-81
- Alinear la estructura de COCOMO II a los distintos sectores del mercado del SW
- Alinear las entradas y salidas de los sub-modelos de COCOMO II a los niveles de información disponibles
- Permitir que los sub-modelos de COCOMO II se ajusten a la estrategia de proceso particular de un proyecto
- No produce una estimación puntual, sino un rango de estimación atado al grado de definición de los datos de entrada a la estimación

Alberto Cueto – ECOS



Modelo del Mercado del Software

No atendido por
COCOMO II

Programación del Usuario Final

Generadores
Aplicaciones y
Ayudas de
Composición

Composición de
Aplicaciones

Integración de
Sistemas

Infraestructura

Alberto Cueto – ECOS



Mercado SW - Generadores Aplicaciones

- Crean productos pre empacados para uso de los usuarios finales en la programación

Alberto Cueto – ECOS



Mercado SW - Composición de Aplicaciones

- Aplicaciones muy diversificadas para disponer de soluciones empacadas,
- Suficientemente simples para ser compuestas rápidamente de componentes interoperables
 - Creadores de interfaces gráficas
 - Administradores de bases de datos y objetos
 - *Middleware* para procesamiento distribuido o procesamiento transaccional
 - Administradores de hipermedia
 - Componentes de dominios específicos tales como paquetes para los sectores financiero, médico o industrial

Alberto Cueto – ECOS



Mercado SW - Integración de Sistemas

- Sistemas de gran escala, alta integración o sin precedentes
- Parte de los mismos se pueden construir por Composición de Aplicaciones, pero normalmente requieren mucho trabajo de ingeniería y de personalización

Alberto Cueto – ECOS



Modelos

- Modelo de Composición de Aplicaciones
 - Basado en Puntos de Objetos
- Modelo del Diseño Temprano
 - Exploración de alternativas de arquitecturas de software y del sistema y conceptos de operación
 - Basado en Puntos Funcionales y 7 generadores de costos
- Modelo de Post-arquitectura
 - Desarrollo y mantenimiento de software

Alberto Cueto – ECOS



Modelo de Composición de Aplicaciones

- Util para cualquier esfuerzo de prototipos
- Apropiado en las primeras fases o ciclos de espiral del ciclo de vida
- Basado en [Puntos de Objetos](#)
- El nivel de información disponible no es muy detallado
- No se requiere estimaciones muy precisas
- Util para aplicaciones que se desarrollan en equipos de pocas personas en unas pocas semanas o meses

Alberto Cueto – ECOS

Modelo del Diseño Temprano

- Útil para la exploración de alternativas de la arquitectura del software o del sistema y de conceptos de operación
- Asume que no hay suficiente información para soportar una estimación detallada
- El nivel de detalle es consistente con el nivel general de información disponible y la precisión de estimación necesaria en ese punto del desarrollo
- Basado en [Puntos Funcionales](#) y 7 generadores de costos

Alberto Cueto – ECOS

Puntos de Objetos

- Conteo de las pantallas, reportes y módulos de lenguajes de tercera generación desarrollados en la aplicación
- Cada uno recibe el peso de un factor de complejidad de tres niveles (simple, mediano y difícil)

Alberto Cueto – ECOS



Modelo de Post-Arquitectura

- Responde al desarrollo y mantenimiento real de un producto de software
- Requiere una arquitectura de ciclo de vida
- Requiere información detallada de los multiplicadores de esfuerzo
- Da información detallada sobre los estimados de costos
- Utiliza líneas de código o puntos funcionales para estimaciones de tamaño
- 17 multiplicadores de esfuerzo
- 5 factores que determinan el exponente de escala
- Es el mas similar a COCOMO-81

Alberto Cueto – ECOS



Dimensión del Tamaño del Software

- COCOMO usa tres medidas distintas:
 - Puntos de Objetos
 - Puntos de Función sin ajustar
 - Líneas de Código (SLOC)

Alberto Cuetto – ECOS



Puntos de Función sin Ajustar

- Utiliza la metodología tradicional de puntos funcionales
- No usa los 14 factores de complejidad
 - COCOMO tiene sus propios factores de ajuste via generadores de costos y relaciones

Alberto Cuetto – ECOS



SLOC

- Objetivo: Medir la cantidad de trabajo intelectual puesto en el desarrollo
- Utiliza la lista de chequeo del SEI para líneas de código
- Con Amadeus, una herramienta de recolección automática de información, unifican el criterio de conteo de líneas de código
 - Proveer estándares y datos consistentes provenientes de distintos sitios participantes

Alberto Cueto – ECOS



Ajustando el Tamaño de Desarrollo del SW

- Ajuste al Tamaño
 - BRAK : Desecho (Breakage)
 - Porcentaje de código que fue desechado, con respecto al tamaño del código final que fue entregado
 - ESLOC : Adaptación de Código
 - Equivalencia al desarrollo de un nuevo software
 - AT : Traducción Automática
 - Porcentaje del código al cual se le hace reingeniería por traducción automática

Alberto Cueto – ECOS



Breakage

- Ejemplo
 - Un proyecto entrega un producto con 100,000 líneas de código, pero descarta el equivalente de 20,000 líneas adicionales
 - El factor BRAK para este proyecto es de 20
 - Para este proyecto, se ajustará el tamaño real del proyecto a 120,000 líneas.
- No se utiliza en el modelo de Composición de Aplicaciones

Alberto Cueto – ECOS

Adaptación de Código

$$ESLOC = \frac{ASLOC[AA + AAF(1 + 0.02(SU)(UNFM))]}{100}, AAF \leq 0.5$$

$$ESLOC = \frac{ASLOC[AA + AAF + (SU)(UNFM)]}{100}, AAF > 0.5$$

- Donde:
 - **ESLOC** = Equivalente a nuevo software a ser desarrollado, mediante la reutilización
 - **ASLOC** = Número de líneas de código adaptado del software existente y utilizado en el desarrollo del nuevo producto
 - **AAF** = Factor de Ajuste de la adaptación del software
 - $AAF = 0.4 * DM + 0.3 * CM + 0.3 * IM$
 - **UNFM** = No familiaridad del programador con el software
 - **SU** = Penalización para entender el software
 - **AA** = Evaluación y Asimilación

Alberto Cueto – ECOS

AAF

■ $AAF = 0.4 * DM + 0.3 * CM + 0.3 * IM$

- **DM** = Porcentaje del **diseño** del software modificado que recibe modificaciones para satisfacer los objetivos y ambiente del nuevo producto
- **CM** = Porcentaje del **código** del software adaptado que recibe modificaciones para satisfacer los objetivos y ambiente del nuevo producto
- **IM** = Porcentaje del **esfuerzo** de integración original necesario para integrar y probar el software adaptado para combinarlo en el nuevo producto

Alberto Cueto – ECOS

UNFM

- **UNFM**: No familiaridad del Programador con el código

Nivel de UNFM	Nivel de Familiaridad
0.0	Completamente familiar
0.2	Muy familiar
0.4	Algo familiar
0.6	Considerablemente familiar
0.8	Principalmente no familiar
1.0	Completamente no familiar

Alberto Cueto – ECOS

SU - Entendimiento del Software

- SU = Penalización para entender el software
 - Medido con base a la estructura, la claridad de la aplicación y la autodocumentación
 - Se penaliza desde 10% (Very high) hasta 50% (very low)

	Very Low	Low	Nom	High	Very High
Structure	Very low cohesion, high coupling, spaghetti code.	Moderately low cohesion, high coupling.	Reasonably well-structured; some weak areas.	High cohesion, low coupling.	Strong modularity, information hiding in data / control structures.
Application Clarity	No match between program and application world views.	Some correlation between program and application.	Moderate correlation between program and application.	Good correlation between program and application.	Clear match between program and application world-views.
Self-Descriptiveness	Obscure code; documentation mission, obscure or obsolete	Some code commentary and headers; some useful documentation.	Moderate level of code commentary, headers, documentations.	Good code commentary and headers; useful documentation; some weak areas.	Self-descriptive code; documentation up-to-date, well-organized, with design rationale.
SU Increment to AAF	50	40	30	20	10

Alberto Cueto – ECOS

AA - Evaluación y Asimilación

- AA = Evaluación y Asimilación
 - Esfuerzo requerido para determinar si un módulo de SW completamente reutilizable es apropiado para la aplicación y para integrarlo en la descripción general del producto
 - Varía de 0 (ningún esfuerzo) a 8 (requiere pruebas, validación y documentación)

Incremento de AA

0

2

4

6

8

Nivel de Esfuerzo de AA

Ninguno

Búsqueda de los módulos básicos y la documentación

Prueba y evaluación (T&E) de algunos módulos y documentación

T&E considerable de algunos módulos y documentación

T&E extensiva y documentación

Alberto Cueto – ECOS

Traducción Automática

$$Esfuerzo_{Nominal} = A * (tamaño)^B + \left[\frac{ASLOC \times \left(\frac{AT}{100} \right)}{ATPROD} \right]$$

•Donde:

- **ASLOC** = Número de líneas de código adaptado del software existente y utilizado en el desarrollo del nuevo producto
- **AT** = Porcentaje del código que es traducido automáticamente
- **ATPROD** = Productividad de la traducción automática. Actualmente se ha estimado en **3400** líneas de código por persona / mes

Alberto Cueto – ECOS

Traducción Automática

Objetivo Reingeniería	AT (% Traducción Automática)
Procesamiento en batch	96%
Batch con SORT	90%
Batch con DBMS	88%
Batch, SORT, DBMS	82%
Interactivo	50%

Alberto Cueto – ECOS

Ecuaciones Básicas - Esfuerzo

- $\text{Esfuerzo}_{\text{Nominal}} = A * (\text{Tamaño})^B$
- $B = 0.91 + 0.01 \sum SF_i$

Donde:

A = Constante que expresa el efecto lineal del esfuerzo con proyectos de tamaño creciente. Actualmente = **2.45**

B = Constante que caracteriza las economías de escala encontradas cuando un proyecto de software incrementa su tamaño

SF = Factores de Escala: PREC, FLEX, RESL, TEAM y PMAT. Cada factor de escala tiene un valor entre 0 y 5

Alberto Cueto – ECOS

Factores de Escala

Factor de Escala (W _i)	Muy Bajo (5)	Bajo (4)	Nominal (3)	Alto (2)	Muy Alto (1)	Extra Alto (0)
PREC Precedence	Thoroughly unprecedented	Largely unprecedented	Somewhat unprecedented	Generally familiar	Largely familiar	Thoroughly familiar
FLEX Development Flexibility	Rigorous	Occasional relaxation	Some relaxation	General conformity	Some conformity	General goals
RESL Architecture / risk resolution	Little (20%)	Some (40%)	Often (60%)	Generally (75%)	Mostly (90%)	Full (100%)
TEAM Team cohesion	Very difficult interactions	Some difficult interactions	Basically cooperative interactions	Largely cooperative	Highly cooperative	Seamless interactions
PMAT Process maturity	Weighted average of "Yes" answers to CMM Maturity Questionnaire					

Alberto Cueto – ECOS

Factores de Escala - PREC

- **PREC**: Grado de conocimiento del producto y del ambiente

Característica	Muy bajo	Nominal / Alto	Extra Alto
Entendimiento organizacional de los objetivos del producto	General	Considerable	Completo
Experiencia en trabajar con sistemas de software relacionados	Moderado	Considerado	Amplia
Desarrollo concurrente de nuevo hardware y procedimientos operacionales	Amplio	Moderado	Alguno
Necesidad de arquitecturas de procesamiento de datos y algoritmos innovativos	Considerable	Alguna	Mínima

Alberto Cueto – ECOS

Factores de Escala - FLEX

- **FLEX**: Flexibilidad para el desarrollo

Característica	Muy baja	Nominal / Alto	Extra Alto
Necesidad de que el software cumpla con los requerimientos preestablecidos	Total	Considerable	Básica
Necesidad de que el software cumpla con las especificaciones de interfaces externas	Total	Considerado	Básica
Bonificación por terminación temprana	Alta	Media	Baja

Alberto Cueto – ECOS

Factores de Escala - RESL (1)

- **RESL:** Administración del riesgo y problemas con la arquitectura

Característica	Muy Bajo	Bajo	Nominal	Alto	Muy Alto	Extra Alto
Administración del riesgo: El plan identifica todos los riesgos críticos y establece acciones para resolverlos	Ninguna	Pocos	Algo	Generalmente	Mayoría	Total
Programación, presupuesto e hitos internos son compatibles con el plan de administración del riesgo	Ninguna	Poco	Algo	Generalmente	Mayoría	Total
Porcentaje de la programación de desarrollo destinada a establecer la arquitectura, dados los objetivos del producto	5	10	17	25	33	40
Porcentaje de arquitectos de software de alto nivel disponibles para el proyecto	20	40	60	80	100	120

Alberto Cueto – ECOS

Factores de Escala - RESL (2)

Característica	Muy Bajo	Bajo	Nominal	Alto	Muy Alto	Extra Alto
Disponibilidad de herramientas de soporte para resolver elementos de riesgo, desarrollar y verificar las especificaciones de arquitectura	Ninguna	Pocas	Algunas	Buena	Fuerte	Total
Nivel de incertidumbre en los elementos claves de la arquitectura: misión, interfaz del usuario, COTS, hardware, tecnología, rendimiento	Extrema	Significativa	Considerable	Alguna	Poca	Muy poca
Numero y criticidad de los elementos de riesgo	>10 críticos	5-10 críticos	2-4 Críticos	1 Crítico	> 5 no críticos	<5 no críticos

Alberto Cueto – ECOS

Factores de Escala - TEAM

- **TEAM**: Cohesión del Equipo de Trabajo

Característica	Muy Bajo	Bajo	Nominal	Alto	Muy Alto	Extra Alto
Consistencia de objetivos y cultura de los involucrados	Poca	Alguna	Básica	Considerable	Fuente	Total
Habilidad y deseo de los involucrados para acomodarse a los objetivos de otros involucrados	Poca	Alguna	Básica	Considerable	Fuente	Total
Experiencia de los involucrados en operar como un equipo	Ninguna	Poca	Poca	Básica	Considerable	Extensa
Construcción de equipo de los involucrados para conseguir visión y compromisos comunes	Ninguna	Poca	Poca	Básica	Considerable	Extensa

Alberto Cueto – ECOS

Factores de Escala - PMAT

- PMAT: Madurez del proceso de software
- Utiliza el modelo CMM del SEI
- Dos métodos para evaluar
- Método 1:
 - CMM Nivel 1 (parte baja) = 5; Nivel 1 (parte alta) = 4; Nivel 2 = 3; Nivel 3 = 2; Nivel 4 = 1; Nivel 5 = 0;

Alberto Cueto – ECOS

Factores de Escala - PMAT (2)

- Método 2:
 - Utilice la evaluación de CMM (si es reciente) o
 - Evalúe el porcentaje de cumplimiento de los objetivos de cada KPA:
 - Casi siempre (>90%) = 5; Frecuentemente (60-90%) = 4; Aproximadamente la mitad (40-60%) = 3; Ocasionalmente (10-40%) = 2; Nunca o casi nunca (<10%) = 1; No aplica = 0; No sabe = 0

$$PMAT = 5 - \left[\sum_{i=1}^{18} \left(\frac{KPA\%_i}{100} \times \frac{5}{18} \right) \right]$$

Alberto Cueto – ECOS

Ajuste del Esfuerzo Nominal

- El esfuerzo nominal calculado se ajusta con algunos multiplicadores de esfuerzo:
 - 7 - Modelo de diseño temprano
 - 17 - Modelo de Post-Arquitectura

Diseño Temprano	Post-Arquitectura
RCPX	RELY, DATA, CPLX, DOCU
RUSE	RUSE
PDIF	TIME, STOR, PVOL
PERS	ACAP, PCAP, PCON
PREX	AEXP, PEXP, LTEX
FCIL	TOOL, SITE
SCED	SCED

Alberto Cueto – ECOS

Ajuste Esfuerzo Nominal - Ecuaciones

$$Esfuerzo_{ajustado} = Esfuerzo_{nominal} \times \left(\prod_{i=1}^k ME_i \right)$$

Donde:

- Esfuerzo se mide en Hombres-Mes
- ME = Multiplicadores de esfuerzo
- $k = 7$ para el Modelo de Diseño temprano
- $k = 17$ para el Modelo de Post-Arquitectura

Alberto Cueto – ECOS

Multiplicadores de Esfuerzo

Variable	May Bajo	Bajo	Nominal	Alto	May Alto	Extra Alto
RELY	0.82	0.92	1	1.10	1.26	
DATA		0.90	1	1.14	1.28	
CPLX	0.73	0.87	1	1.17	1.34	1.74
DOCU	0.81	0.91	1	1.11	1.23	
RUSE		0.95	1	1.07	1.15	1.24
TIME			1	1.11	1.29	1.63
STOR			1	1.05	1.17	1.46
PVOL		0.87	1	1.15	1.30	
ACAP	1.42	1.19	1	0.85	0.71	
AEXP	1.22	1.10	1	0.88	0.81	
PCAP	1.34	1.15	1	0.88	0.76	
PEXP	1.19	1.09	1	0.91	0.85	
LTEX	1.20	1.09	1	0.91	0.84	
PCON	1.29	1.12	1	0.90	0.81	
TOOL	1.17	1.09	1	0.90	0.78	
NCED	1.43	1.14	1	1	1	
SITE	1.22	1.09	1	0.93	0.86	0.80

Alberto Cueto – ECOS

Multiplicadores de Esfuerzo - Clases

■ Cuatro clases de Modificadores:

- Atributos del Producto
- Atributos de la Plataforma
- Atributos del Personal
- Atributos del Proyecto

Alberto Cueto – ECOS



Multiplicadores de Esfuerzo - Producto

■ Atributos del Producto

- RELY : Fiabilidad requerida del SW
 - **Fiabilidad**: Probabilidad de que el SW realice sus funciones satisfactoriamente en su próxima ejecución, en un periodo de tiempo dado
 - **Bajo**: Una falla genera apenas un pequeño inconveniente
 - **Muy alto**: Posibles pérdidas de vidas humanas
- DATA : Mide el impacto de grandes volúmenes de datos en el desarrollo (v.g. Generar datos de prueba)
 - **Bajo**: < 10; **Muy alto**: >1000
- CPLX : Complejidad del Producto
 - Determina la complejidad del producto contra el tipo de módulo

Alberto Cueto – ECOS



Multiplicadores de Esfuerzo - RELY

Característica	Muy Bajo	Bajo	Nominal	Alto	Muy Alto	Extra Alto
RELY	Ligeramente inconveniente	Baja, pérdidas fácilmente recuperables	Moderado, pérdidas fácilmente recuperables	Alta pérdida financiera	Riesgo de vidas humanas	

Alberto Cueto – ECOS

Multiplicadores de Esfuerzo - DATA

D Tamaño de la base de datos (bytes)

--- = -----

P Tamaño del programa (SLOC)

Característica	Muy Bajo	Bajo	Nominal	Alto	Muy Alto	Extra Alto
DATA		$D/P < 10$	$10 \leq D/P < 100$	$100 \leq D/P < 1000$	$D/P \geq 1000$	

Alberto Cueto – ECOS

Multiplicadores de Esfuerzo - CPLX

- La complejidad se divide en cinco áreas:
 - Operaciones de control
 - Operaciones computacionales
 - Operaciones dependientes de dispositivos
 - Operaciones de administración de datos
 - Operaciones de administración de la interfaz de usuarios
- La complejidad es el promedio subjetivo de estas áreas

Alberto Cueto – ECOS

Multiplicadores de Esfuerzo - Producto (2)

- Atributos del Producto (Cont.)
 - RUSE: Reutilización requerida
 - Esfuerzo adicional requerido para construir componentes reutilizables
 - **Bajo**: Ninguno; **Extra alto**: a lo largo de múltiples plataformas
 - DOCU:
 - Adecuación de la documentación al ciclo de vida
 - **Muy Bajo**: No cubre muchas de las necesidades del ciclo de vida; **Muy alto**: Muy excesiva para las necesidades del ciclo de vida

Alberto Cueto – ECOS

Multiplicadores de Esfuerzo - RUSE

El esfuerzo es consumido en la creación de un diseño de software mas genérico, documentación mas elaborada y pruebas mas extensas para asegurar que los componentes están listos para ser usados en otras aplicaciones

Característica	Muy Bajo	Bajo	Nominal	Alto	Muy Alto	Extra Alto
RUSE		Ninguna	A lo largo del proyecto	A lo largo del programa	A lo largo de la línea de productos	A lo largo de múltiples plataformas

Alberto Cueto – ECOS

Multiplicadores de Esfuerzo - DOCU

Característica	Muy Bajo	Bajo	Nominal	Alto	Muy Alto	Extra Alto
DOCU	Muchas necesidades del ciclo de vida no cubiertas	Algunas necesidades del ciclo de vida no cubiertas	Adecuada para el ciclo de vida	Excesiva para las necesidades del ciclo de vida	Muy excesiva para las necesidades del ciclo de vida	

Alberto Cueto – ECOS

Multiplicadores de Esfuerzo - Plataforma

- Atributos de la Plataforma
 - **TIME**: Restricciones en el tiempo de ejecución sobre el sistema
 - % tiempo uso producto / % tiempo disponible
 - Nominal: 50%; Extra alto: 95%
 - **STOR**: Limitaciones de Memoria Principal
 - % uso memoria por producto / % uso memoria disponible
 - Nominal: <= 50%; Extra alto: 95%
 - **PVOL**: Complejidad de la plataforma (HW + SW) que el sistema utiliza
 - Máquina virtual: Conjunto de HW + SW que usa el producto
 - Bajo: Cambio mayor cada año; Muy Alto: Cambio mayor cada 2 semanas

Alberto Cueto – ECOS

Multiplicadores de Esfuerzo - TIME

Característica	Muy Bajo	Bajo	Nominal	Alto	Muy Alto	Extra Alto
TIME			<= 50% de uso del tiempo de ejecución disponible	70%	85%	95%

Alberto Cueto – ECOS

Multiplicadores de Esfuerzo - STOR

Característica	Muy Bajo	Bajo	Nominal	Alto	Muy Alto	Extra Alto
STOR			$\leq 50\%$ de uso de la memoria principal disponible	70%	85%	95%

Alberto Cueto – ECOS

Multiplicadores de Esfuerzo - PVOL

- Volatilidad de la plataforma:
 - Plataforma: HW + SW que el sistema utiliza
 - La plataforma puede variar desde solo hardware (desarrollo del sistema operacional), hasta el conjunto de redes, hardware, sistemas operacionales y repositorios de información distribuida (*browser* de texto en la red)
- Mide la frecuencia de cambios grandes y pequeños

Característica	Muy Bajo	Bajo	Nominal	Alto	Muy Alto	Extra Alto
PVOL		Cambio mayor cada 12 meses; cambio menor cada mes	Mayor: 6 meses; menor: 2 semanas	Mayor: 2 meses; menor: 1 semana	Mayor: 2 semanas; Menor: 2 días	

Alberto Cueto – ECOS

Multiplicadores de Esfuerzo - Personal

- Atributos del Personal
 - ACAP: Capacitación de los analistas
 - Habilidad para el análisis y diseño, eficiencia y calidad en el trabajo, habilidad para comunicarse y cooperar
 - **Muy bajo**: < percentile 15; **muy alto**: percentile 90
 - PCAP: Capacidad de los programadores, como equipo
 - Similar a ACAP, pero para los programadores
 - AEXP: Experiencia en el nivel de aplicaciones (del equipo de desarrollo)
 - **Muy bajo**: < 2 meses; **Muy alto**: > 6 años
 - PEXP: Experiencia en la plataforma
 - Similar a AEXP, pero con respecto a la plataforma

Alberto Cueto – ECOS



Multiplicadores de Esfuerzo - Personal (2)

- Atributos del Personal
 - LTEX: Experiencia en el lenguaje de programación y en las herramientas de software del equipo desarrollando el sistema
 - Similar a LTEX, pero con respecto a los lenguajes de programación y herramientas de desarrollo
 - PCON: Rotación del personal
 - **Muy bajo**: 48% por año; **Muy alto**: 3% por año

Alberto Cueto – ECOS



Multiplicadores de Esfuerzo - ACAP/PCAP

Característica	Muy Bajo	Bajo	Nominal	Alto	Muy Alto	Extra Alto
ACAP PCAP	Percentile 15	Percentile 35	Percentile 55	Percentile 75	Percentile 90	

Alberto Cueto – ECOS

Multiplicadores de Esfuerzo- AEXP/PEXP/LTEX

Característica	Muy Bajo	Bajo	Nominal	Alto	Muy Alto	Extra Alto
AEXP PEXP LTEX	2 meses	6 meses	1 año	3 años	6 años	

Alberto Cueto – ECOS

Multiplicadores de Esfuerzo - PCON

Característica	Muy Bajo	Bajo	Nominal	Alto	Muy Alto	Extra Alto
PCON	48% / año	24% / año	12% / año	6% / año	3% / año	

Alberto Cueto – ECOS

Multiplicadores de Esfuerzo - Proyecto

- **Atributos del Proyecto**
 - **TOOL**: Uso de herramientas para el desarrollo del SW
 - **Muy bajo**: Herramientas básicas de edición y codificación; **Muy alto**: Herramientas de administración del ciclo de vida completo
 - **SITE**: Desarrollo en múltiples sitios
 - Mide el esfuerzo por tener el desarrollo distribuido en múltiples sitios
 - Tiene en cuenta distribución del sitio y soporte de comunicaciones
 - **Muy bajo**: Internacional, algunos teléfonos y correo; **extra alto**: completamente localizado, multimedia interactiva
 - **SCED**: Programación de desarrollo requerida
 - Porcentaje de retraso o aceleración con respecto a la planificación nominal impuesta al equipo de desarrollo
 - **Muy bajo**: 75% del nominal; **Muy alto**: 160%

Alberto Cueto – ECOS

Multiplicadores de Esfuerzo - TOOL

Característica	Muy Bajo	Bajo	Nominal	Alto	Muy Alto	Extra Alto
TOOL	Edit, code, debug	Simple, front end, back end, CASE, little integration	Basic lifecycle tools, moderately integrated	Strong, mature life cycle tools, moderately integrated	Strong, mature, proactive lifecycle tools, well integrated with processes, methods, reuse	

Alberto Cueto – ECOS

Multiplicadores de Esfuerzo - SITE

Característica	Muy Bajo	Bajo	Nominal	Alto	Muy Alto	Extra Alto
SITE: Distribución	Internacional	Multi-ciudad y Multi-compañía	Multi-ciudad o Multi-Compañía	Misma ciudad o área metropolitana	Mismo edificio o complejo	Completamente unido
SITE: Comunicaciones	Algunos teléfonos, correo	Teléfono individual, FAX	Email, banda angosta	Comunicación electrónica de banda ancha	Comunicación electrónica de banda ancha, ocasionalmente video conferencia	Multimedia interactiva

Alberto Cueto – ECOS

Multiplicadores de Esfuerzo - SCED

- Porcentaje de retraso o aceleración con respecto a la planificación nominal impuesta al equipo de desarrollo

Característica	Muy Bajo	Bajo	Nominal	Alto	Muy Alto	Extra Alto
SCED	75% del nominal	85%	100%	130%	160%	

Alberto Cueto – ECOS

Ecuaciones Básicas - Tiempo

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

Donde:

- $TDEV$ = Tiempo calendario en meses, desde la determinación de requerimientos hasta la aceptación del producto
- esfuerzo = El esfuerzo estimado en hombres-mes, excluyendo el multiplicador de esfuerzo [SCED](#)
- $SCED\%$ = Porcentaje de retraso o aceleración del proyecto, calculado en el multiplicador de esfuerzo [SCED](#)

Alberto Cueto – ECOS

