XIV Conferencia Latinoamericana de Informática
17avas. Jornadas Argentinas de Informática
e Investigación Operativa
Buenos Aires, Setiembre 1988.

MACOS: Modelo Algorítmico de Costos de Software.

L. Guerra G, S. Concha C., Depto de Informática.
Universidad Técnica Federico Santa María, Casilla 110-V Valparaíso, Chile.

RESUMEN

El presente trabajo constituye un análisis de los modelos de costos de software, recopilación de datos de proyectos de desarrollo de productos de software, análisis de tal información y finalmente la formulación y evaluación de un modelo algorítmico analítico para determinar costos del software: MACOS.

1.- ANTECEDENTES DE LA MODELACION DE COSTOS DE SOFTWARE.

A partir del estudio de: lo que ha sido la " Modelación de costos de software" hasta hoy, de la clasificación de los distintos métodos de estimación y finalmente de la recopilación y análisis de los distintos modelos de tipo algorítmo encontrados en literatura (1,2,3,4,5,6,7,8,9,10, 11,12), se concluye lo siguiente:

Durante el análisis de los distintos modelos de estimación de costos de software se detectaron las siguientes características comunes a ellos:

Todos los modelos consideran el tamaño del producto como el principal determinante de los costos y el esfuerzo del desarrollo.

La mayoría dimensiona el producto utilizando medidas relacionadas íntimamente con el código producido: Instrucciones Fuentes Entregadas, Instrucciones de Código Objeto, Número y Tipo de Operandos y Operadores utilizados.

La mayoría considera otros factores distintos del tamaño del producto de tal manera de ajustar la estimación del costo o esfuerzo obtenida como función del tamaño. Algunos integran estos factores en una sola función analítica y otros lo hacen a través de multiplicadores que modifican la estimación inicial en base al tamaño.

El problema de todos radica en que no se integran los puntos de vista del desarrollador y del usuario en la estimación. Por ejemplo, COCOMO (11) tan sólo considera factores que son perceptibles por el desarrollador; por otro lado, la técnica Puntos de Función (12) tan soló incluye aquellos perceptibles por el usuario.

Además de estas características, es importante destacar los dos hechos siguiente:

Se han hecho algunos intentos con modelos que no dimensionan el producto usando medidas relacionadas íntimamemte con el còdigo escrito. Estos usan medidas altamente subjetivas: Número de Líneas de Especificación, Número de Funciones, etc.

Un estudio de Kitchenham y Taylor (13) acerca del modelo COCOMO y la distribución del esfuerzo y tiempo en las distintas fases del desarrollo concluyó que los resultados de aquel son del mismo orden que el esfuerzo y tiempo reales y que, en promedio, los porcentajes de esfuerzo y tiempo dedicados a cada fase son muy similares a los determinados en COCOMO pero que cada proyecto individual varía mucho respecto del promedio.
Lo anterior se tomó como base para no realizar una distribución (en términos porcentuales) de esfuerzo y tiempo en las fases de desarrollo.

Los hechos anteriores y las características mencionadas llevaron a establecer las siguientes bases para establecer un modelo.

1.- Considerar el tamaño del producto como el principal determinante de los costos de desarrollo de software. Esta consideración a priori fue comprobada posteriormente mediante una ronda de consultas con expertos.

2.- Dimensionar el producto con alguna medida independiente del código de tal forma de permitir una mejor estimación del tamaño en fases tempranas del desarrollo.

3.- Integrar en el modelo a desarrollar los puntos de vista del desarrollador y del usuario a través de otros factores distintos del tamaño que fueran perceptibles para cada uno.

4.- Considerar estimaciones globales (para el desarrollo como un todo) de esfuerzo y tiempo.

2.- FACTORES CONSIDERADOS "A PRIORI EN MACOS.

Sobre la base de las conclusiones obtenidas anteriormente se presentan los factores que se consideran en MACOS como los principales determinantes de los

costos y el esfuerzo de desarrollo. El conjunto de factores considerados es el resultado del análisis de todos aquellos incluídos en los modelos existentes, de la realización de una encuesta (que resulta ser muy interesante de analizar a fondo) a personas que han estado mucho tiempo relacionada con el desarrollo de software en Chile y de la realización de un experimento que permitió determinar la medida de dimensionamiento del tamaño del producto de software (considerado el principal determinante de los cotos y el esfuerzo).
Se definieron algunas características que debiera tener una medida del tamaño del producto:

Significativa al usuario no técnico.
Independiente de la tecnología de desarrollo de software.
Servir como estimador.
Objetiva.
Disponible antes de la "fabricación" del producto".
Contabilizada automáticamente.

Analizando estas características en los siguientes candidatos:

Código objeto.
Código fuente.
Instrucciones categorizadas.
Puntos de Función.
Métricas de Halstead.

se concluyó que la medida "Puntos de Función" cumplía cabalmente con las características 1, 2 y 5. Respecto de la característica 6, los "Puntos de Función" no se pueden contabilizar automáticamente.

En lo relacionado con la característica 3 se estableció:

Los "Puntos de Función" pueden ser estimados razonablemente bien en fases tempranas del desarrollo.
Existe correlación entre la medida y el esfuerzo (en HH, por ejemplo) dedicado al desarrollo.

Albretch (12) definió los puntos de función considerando, además del conteo, una complejidad asociada a cada Entrada, Salida, Archivo Lógico Interno, Interfase y Consulta que componen el producto de software. La razón de asociar esta complejidad es cuantificar el procesamiento asociado con cada función (y por ende el código asociado a ellas).

En el modelo MACOS se optó por considerar aparte los aspectos relacionados con la complejidad de procesamiento, es decir, los puntos de función sólo se consideran como una medida del tamaño del producto y no una forma de cuantificar el procesamiento asociado con ellos. Se han eliminado además las ponderaciones asociadas con cada nivel de complejidad (Albretch explica estas ponderaciones como la importancia que tiene la función para el usuario y son determinadas en base a prueba y error).

Esta decisión hace que la contabilización de los Puntos de Función sea mucho más sencilla y requiera menos tiempo. Por lo que se soluciona la principal desventaja de la medida: la dificultad de su contabilización.

Por otro, lado, se disminuye la subjetividad de la medida al no tener que asociar a cada función un grado de complejidad. Respecto de esta última característica (la número 4.-) quedan dudas aún y es necesario estudiar más acerca de ello para disminuir esta subjetividad lo más posible; es probable que a nivel de instalación sea factible el definir más acabadamente que se entiende por "Puntos de Función" (definiendo estándares) y de esta manera disminuir su subjetividad.
Se definieron otros factores que inciden en el esfuerzo dedicado al desarrollo de software, a saber:

- INTUSU :Interfase con el Usuario.
- ESPREQ :Especificación de requerimientos.
- RECONT :Plan de Recursos y control del proyecto.
- PLAPRO :Plan Detallado del Proyecto.
- TESTVV :Plan Detallado de Testing y Verificación y Validación
- AMBTRA :Ambiente de Trabajo.
- DOCEXI :Documentación Existente.
- CONPER :Continuidad del Personal.
- CALADM :Calidad de la administración
- DATARC :Características de Captura, Validación Actualización y Almacenamiento de Datos.

- FAFLCA :Facilidad de Cambio y Flexibilidad.
- CONPRO :Confiabilidad del Producto.
- RESTPO :Restricciones en el Tiempo de Desarrollo.
- MODDOP :Modo de Desarrollo y Operación del producto.
- EXUSUA :Experiencia del Usuario.
- CAPAN :Capacidad de los Analistas.
- EXPANA :Experiencia de los Analistas.
- CAPPRO :Capacidad de los programadores.
- EXPLEN :Experiencia en el lenguaje.
- RAPCPU :Restricciones de A.P. y CPU.
- TRESPC :Tiempo Respuesta Promedio del Computador.
- HERRAD :Herramientas de Ayuda al Desarrollo.
- PMOPRO :Prácticas Modernas de Programación.
- COMPLP :Complejidad del Producto.
- FAICMA :Facilidad de Instalación,Conversión y Mantención.

3.- RECOPILACION DE DATOS.

Para poder establecer el modelo MACOS se recopiló la información necesaria para ello: acerca de los factores determinados y de otras características de un proyecto de desarrollo: costos, esfuerzo, horas de computador, nivel de documentación, etc.

Durante la recopilación de datos se detecto que, en general, las distintas empresas del área no llevan un registro formal del proceso de desarrollo de software. Esto llevó a tener que trabajar con datos que provenían del "recuerdo" de las personas por lo que, desde el punto de vista de su fidelidad,es posible que existan algunas dudas.

La principal conclusión de la recopilación de datos es, entonces,la necesidad de contar con un registro formal del desarrollo de proyectos de software.
Lo anterior significa que el registro debe ser fácil de llevar a cabo pues de otra manera significaría distraer el esfuerzo en la tarea principal que es el desarrollo.

4.- ANALISIS DE LA INFORMACION RECOPILADA.

Además de la recopilación de datos, se hizo necesario realizar una serie de análisis de ella para obtener algunas conclusiones acerca de las relaciones existentes, por ejemplo entre: esfuerzo y tamaño del producto, horas de

computador y tiempo de desarrollo, etc. Todas estas relaciones se integran en el modelo MACOS.
Durante el análisis de los datos, considerando la cantidad de información recopilada y la fidelidad de información, se determino que:

Existe correlación entre PF y esfuerzo.

Por otro lado, es interesante el aspecto de la productividad del desarrollo. Es destacable el hecho de que ésta permanezca como una constante no dependiente del lenguaje ni del tamaño del producto.

Se determinó que los montos de documentación y de tiempos de computador varían mucho entre los distintos sistemas, por lo que se confirma lo planteado por Boehm (11) respecto de estas medidas.

En relación a la distribución del esfuerzo y tiempo en las distintas fases del desarrollo y a los factores distintos del tamaño del producto, no fue posible analizar sus comportamientos debido a la falta de información acerca de ellos.

5.- EL MODELO MACOS.

De acuerdo a las conclusiones de las actividades precedentes se presenta el modelo MACOS con sus definiciones y suposiciones, la calibración del modelo a una realidad particular y otros aspectos adicionales.

1.- MACOS es un modelo de estimación de tiempos de costos de software de tipo algorítmico. Considera la siguiente ecuación principal:

$$ENE = 3.55 * PF \hat{} 1.35 \qquad (1)$$

La ecuación (1) entrega una estimación inicial o nominal del esfuerzo (ENE) medido en Horas Hombre (H-H).

2.- Una Hora - Hombre se define como :
"El trabajo de una persona durante una hora considerando overhead.

Se elige H-H como medida del esfuerzo del trabajo, a cada H-H se le asocia un Costo Promedio que permite determinar el Costo Total de desarrollo del proyecto.

3.- El principal determinante de los costos es el tamaño del producto, medido en Puntos de Función (PF). Los Puntos de Función dimensionan un producto de software contabilizando el número de :

- Entradas del usuario externo.
- Salidas al usuario externo.
- Consultas del usuario externo.
- Archivos maestros desde el punto de vista lógico del usuario.
- Interfases con otros programas de aplicación.

La suma de las cantidades de cada uno de los ítems anteriores es el número de PF.

4.- Las fases cubiertas por la estimación son:

Conocimiento del problema.
Definición de requerimientos.
Análisis (Diseño Lógico, Confección de Modelo de Datos).
Diseño físico (Construcción de la estructura computacional, definición del modelo físico de datos).
Programación (Implementación y Pruebas, Construcción).
Test, Integración, Puesto en marcha, Documentación final.

El tiempo del desarrollo (TDES) se mide en Horas. La principal razón para ello es evitar al máximo los problemas que provocan las diferencias entre las horas trabajadas por mes (al usar la medida del esfuerzo Hombres-Mes). Estas diferencias se deben principalmente al trabajo en horas extraordinarias y a la dedicación parcial de alguna de las personas trabajando en el proyecto.

La estimación inicial o nominal de esfuerzo (EN) dada por la ecuación (1) es ajustada por un Factor de Ajuste (FA) definido por:

$$FA = 1 + X/100 + y/100$$

X es la suma de todos los Grados de Influencia (Gi) positivos; aumento de los costos.

y es la suma de todos los Grados de influencia (GI) negativos; disminución de los costos.

Donde los Grados de Influencia (GI) definen la incidencia de los factores ya mencionados.

Los GI son :

DISMINUCION DE LOS COSTOS.

Influencia muy importante	--->	- 5
Influencia significativa	--->	- 4
Influencia promedio	--->	- 3
Influencia moderada	--->	- 2
Influencia incidental o muy baja	--->	- 1

SIN EFECTO

No presente o sin influencia	--->	0

AUMENTO DE LOS COSTOS

Influencia incidental o muy baja	--->	1
Influencia moderada	--->	2
Influencia promedio	--->	3
Influencia significativa	--->	4
Influencia muy importante	--->	5

El producto del EN por el FA determina el Esfuerzo Ajustado (en H-H):

EAE = ENE * FA

El EAE representa el esfuerzo que se estima, debe dedicarse al desarrollo del producto considerado.

Con el EAE se utiliza

TDESE = EAE * NTPC

donde:

NTPC es el Nùmero Promedio de Personas de Tiempo Completo trabajando en el proyecto.
TDESE es el Tiempo de Desarrollo Estimado.

Para estimar el Tiempo de Desarrollo (TDESE) requerido.

El EAE multiplicado por una constante que indica el costo por hora-hombre, entrega la estimación del costo para el desarrollo del proyecto.

No se hace distribución del esfuerzo y el tiempo sobre las fases de desarrollo debido a las razones dadas anteriormente.

6.- CONCLUSIONES.

Es necesario revisar varios aspectos del modelo planteado : relación esfuerzo con PF, definición de los PF, relación EAE/tiempo de desarrollo, integración de los factores distintos del tamaño al modelo, definición del FA , etc.
A pesar de lo anterior, el Error Promedio de Estimación de COCOMO es muchisimo más alto que el de MACOS.
Lo anterior significa que COCOMO no se ajusta a la realidad del desarrollo de software en Chile, por lo que definir un modelo como MACOS permite hacer estimaciones realistas en nuestro ambiente.

Es importante destacar que MACOS, debido a los proyectos considerados, está orientado a sistemas administrativos de tamaño pequeño a mediano (basándose en el número de líneas de código fuente de los proyectos considerados). Sera interesante estudiar su aplicabilidad en otro tipo de sistemas.

La concepción de MACOS pretende suplir las fallas encontradas en otros modelos. Muchas de sus características son una primera aproximación a la solución de tales problemas . Lo anterior significa que aún queda muchisimo por hacer.

- Falta una definición más acabada de la medida del dimensionamiento. En este tema sería interesante buscar alternativas aunque Puntos de Función es un buen candidato.

- Es necesario definir mejor la integración de los factores distintos del tamaño . Para ello se recomienda "dimensionar" la influencia de estos factores de manera multiplicativa y además definir un rango en que el "ajuste " sea válido pues, fuera de ese rango es probable que ese rango no sea válido

- Debe definirse otra relación para estimar el Tiempo de Desarrollo.

- Falta una definición completa de las Fases de Desarrollo (y sus actividades) cubiertas por la estimación.

- Si es posible, debiera obtenerse los datos a medida que el proyecto avanza. Esto se hace muy difícil debido a los problemas de restricción en el tiempo de desarrollo a que se enfrentan en general los proyectos de software.

- Lo anterior significa que es necesario definir procedimientos automáticos para recopilación de datos pues, de otra manera, es prácticamente un hecho que se dejará de lado el proceso de recopilación.

7.- REFERENCIAS.

1.-Quantitative analysis of programming cost factors: a progress report. L. Farr , H. J. Zagorski, A.B.Frielnik, 1965.

2.-Management handbook for estimation of computer programming cost. E. A. Nelson, Sistems Development Co.,1967.

3.-Estimating resources for large programming systems J. D Aron, NATO Science Comittee, Rome, 1969.

4.-The cost of developing large-scale software. R. W. Wolverton, IEEE Trans. Computers, 1974

5.-A macro-estimating methodology for software development. L. H. Putnam.Proceedings IEEE. COMPCON 76, 1976.

6.-Prediction of software effort and project duration : four new formulas. V. Schneider. ACM SIGPLAN notices, 1978.

7.-Software cost measuring and reporting, Boening Co. , U. S. Air Force, ASD Document, 1979.

8.-System life cycle estimation (SLICE): A new approach to estimating resources for aplication program development. A. L. Kustanowitz, IEEE COMPSAC, 1977.

9.-A method of programming measurement and estimation. C. E. Walston, C. P. Felix, IBM Systems Journal, 1977.

10.-Estimation of computer sequirements and software development cost. Taback and Ditmore General Research Co. , 1974.

11.-Software Engineering Economics. Barry W. Boehm Prentice Hall, 1981.

12.-Measuring Application development productivity. A. J. Albrecht. SHARE-GUIDE, 1979.

13.-Software Proyect development cost estimation. B. A. Kitchenham, N. R. Taylor. The Journal of Systems and Software , 5, 1985.