



Asociación de Técnicos de Informática

II Jornadas de Calidad de Software

Sesión de mañana del 2 de julio de 1998

Medición de Especificaciones Software

José Javier Dolado Cosín

***Facultad de Informática
Universidad del País Vasco
e-mail: dolado@si.ehu.es***

Este documento no puede ser reproducido o distribuido sin la autorización expresa de su autor.

Medición de Especificaciones Software

Estimación del Tamaño: Clave para la Gestión

- Casi todas las actividades dependen de esta estimación y de una *buena* estimación
- Importancia en la relación cliente-desarrollador
- Cálculo del valor del beneficio de la inversión
- Cálculo del software como activo
- La estimación del *esfuerzo* depende del tamaño
- La productividad depende del esfuerzo y del tamaño
- Grave problema: cómo medir el tamaño uniformemente



Asociación de Técnicos de Informática

II Jornadas sobre Calidad del Software

Javier Dolado

2

¿Cómo Estimar el Tamaño?

- Conteo de Bloques de Funciones y transformarlas en LOC
- Estimación estadística elemental de las LOC: a) estimar el menor número posible de LOC (u otra cosa) $X1$ b) estimar el valor máximo $X2$ c) estimar el valor más probable M .
- El valor esperado es $E = (X1 + 4M + X2)/6$
- La desviación típica es $S = (X2-X1)/6$
- Pero para poder estimar es preciso tener una unidad común de medida.
- Las LOC son difícilmente utilizables en la fase de especificación



Primera Solución : Puntos de Función

- Propuestos por Alan Albrecht en el 79
- La idea básica se ha mantenido a través de los años
- Se han modificado criterios de valoración y de identificación de los constituyentes
- Las definiciones del cálculo y de las reglas a utilizar las unifica el IFPUG
- Han surgido múltiples variantes, debido a los problemas de adaptación de la versión original a las diferentes tecnologías de desarrollo
- Variantes: MarkII, "Feature Points", "3D points", FFP



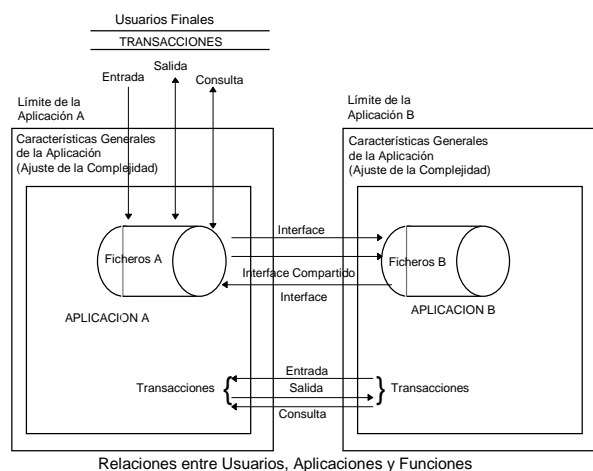
Puntos de Función “Albrecht”

- Se pueden aplicar en desarrollo y mantenimiento
- Se “suponen” independientes de la metodología
- La versión utilizada hoy día difiere ligeramente de la original
- Su autor los propuso para su entorno, no como modelo general
- Es la medida de las especificaciones más conocida, y *por este motivo* es la más utilizada.
- Soluciona el problema de la estimación de las LOC.
- Se utilizan en el modelo COCOMO 2.0



Puntos de Función “Albrecht”

- Orientados a aplicaciones de gestión



Procedimiento de cálculo

- Los pasos son:
 - Determinar cada tipo de elemento
 - Identificar del límite de la aplicación
 - Identificar la complejidad de cada tipo de elemento
 - Determinar los puntos no ajustados
 - Determinar el “Ajuste de Complejidad Técnica” mediante la evaluación de las 14 características
 - Calcular los puntos de función ajustados



Procedimiento de cálculo (cont.)

- *Entradas externas*: entradas del usuario
- *Salidas externas*: proporcionan datos al usuario
- *Consultas externas*: entradas interactivas con respuesta
- *Ficheros externos*: son interfaces con otros sistemas
- *Ficheros internos*: ficheros maestros lógicos del sistema
- Se valora cada elemento $UFC = \sum_{i=1}^{15} (n^{\circ} \text{ items variedad } i) \times (\text{peso}_i)$

ITEM	FACTOR DE PONDERACIÓN		
	Simple	Medio	Complejo
Entrada externa	3	4	6
Salida externa	4	5	7
Consulta usuario	3	4	6
Fichero externo	7	10	15
Fichero interno	5	7	10



Procedimiento de Cálculo (cont.)

- Se valoran las características del entorno de 0 a 5
- Puntos de Función = UFC · TCF
- el ajuste al original UFC es del orden del $\pm 35\%$.

$$TCF = 0.65 + 0.01 \sum_{i=1}^{14} F_i$$

Factores que contribuyen a la complejidad

F1: fiabilidad del back-up y recuperación	F2: comunicaciones de datos
F3: funciones distribuidas	F4: rendimiento
F5: configuración muy cargada	F6: entrada de datos on-line
F7: facilidad de operación	F8: actualización on-line
F9: interface complejo	F10: procesamiento complejo
F11: reusabilidad	F12: facilidad de instalación
F13: localización múltiple	F14: facilidad de cambio



Problemas

- Dificultades en la aplicación homogénea de las reglas.
- El IFPUG y los grupos de interés identifican las soluciones más apropiadas
- Dificultades en la identificación de algunos elementos en entornos de bases de datos
- Diferencias de criterios en el conteo entre novatos y expertos
- Dudas sobre cómo se han construido algunas bases de datos.



Validación de los Puntos de Función

- Disparidades en la magnitud de la relación esfuerzo-PFs
- Habitualmente se asume una relación esfuerzo-PFs
- En algún caso las LOC lo hacen mejor
- Existe relación entre los componentes primarios y el esfuerzo
- Agrupándolos (sin calculo final) se obtiene buena correlación
- Los pesos no añaden información especial
- Los pesos son robustos y se pueden utilizar sin otros datos históricos
- El cálculo final no añade ninguna información especial
- Se recomienda pues, guardar toda la información parcial



Puntos de Función Mark II

- Propuestos por Symons a finales de los 80
- Intentan paliar los problemas de los “Albrecht” en la aplicación a entornos de bases de datos.
- Considera las *transacciones lógicas: entrada, proceso y salida*
- Transacción lógica: combinación única de entrada/proceso/salida desencadenada por un único evento de interés para el usuario, o una necesidad de recuperar información



Puntos de Función Mark II

- $UFP = W_I \cdot (n^\circ \text{ de tipos de datos de entrada}) +$
- $+ W_E \cdot (n^\circ \text{ de tipos de entidades referenciadas}) + W_O (n^\circ \text{ de tipos de datos de salida})$
- W_i calibrados o ($W_I = 0.58$, $W_E = 1.66$ y $W_O = 0.26$)
- se amplía la lista de características generales
- $TCA = 0.65 + C \cdot [\text{suma del grado de Influencia para las 19 características de la aplicación, más las definidas por el cliente}]$. 'C' es 0.005.
- $PF = (UFP) \cdot (TCA)$



Método Basado en los Componentes

- Cuestiona los Puntos de Función como independientes de la Tecnología
- Define “Tipos de Componentes” dependiendo del entorno
- Los componentes pueden o no coincidir con los de los PF
- Realiza las estimaciones de LOC para cada tipo de componente
- Da buenos resultados en general, comparables a una estimación global de LOC mediante PFs
- Se ha utilizado en entornos CASE, donde es fácil identificar componentes



Puntos de Objeto

- Parece que sus resultados son comparables a los PF
- Adaptados a entornos CASE
- Método de cálculo:
 1. Estimar nº de pantallas, informes y componentes 3GL
 2. Clasificarlos como simples, medios o complejos (tablas)
 3. Darles un peso (tabla) y sumarlos
- Es posible tener en cuenta la reutilización en el cálculo
- La idea es similar a los Puntos de Función
- Se utilizan en COCOMO 2.0



Método General de Hakuta et al.

- Propuesto como método general de estimación del tamaño de un programa
- Se identifican las “Unidades de Procesamiento” o módulos (a partir de la especificación)
- Se evalúa la complejidad del procesamiento globalmente
- Se evalúan los factores del entorno
- Para todo ello hay que analizar previamente programas similares que proporcionen los valores de referencia.
- Es parecido al Método Basado en Componentes. Sólo se ha utilizado en programas pequeños.



Método Basado en Metamodelos

- “System Meter”: propuesto específicamente para la Orientación a Objetos, sustituyendo a los PF.
- Utiliza un “Modelo de la Empresa” parecido a OML
- Para cada objeto considera un “tamaño externo” y un “tamaño interno”
- Se realiza un conteo de los elementos que describen cada objeto
- Se permite tener en cuenta la reutilización de objetos
- Los resultados en la predicción de esfuerzo han sido ligeramente superiores a los PF



Otros Métodos

- **A)** Se ha propuesto un “Modelo Orientado a Objetos” en el que
- $\text{Esfuerzo} = F(\text{objetos, operaciones, interfaces})$
- Ha dado resultados medianamente buenos
- Su idea es que existen elementos de las especificaciones que permiten predecir el esfuerzo.
- **B)** Se han propuesto medidas del diseño basadas en modelos teóricos de los DFDs (Redes de Petri).
- La medida es correcta desde el punto de vista teórico. Falta amplia validación práctica.



Conclusiones

- Los niveles de estimación de **todas** las propuestas aquí mencionadas es sólo *medianamente aceptable*.
- La aproximación de los PF es una posible solución
- Puede proporcionar estándares de comparación
- Desde el punto de vista matemático su construcción no es aceptable (Puntos de Función)
- Pero poseen una amplia utilización práctica
- Es más que probable que modelos específicos adaptados a cada entorno específico proporcionen mejores resultados.



Conclusiones

- La medición de las especificaciones es un problema todavía abierto
- El tamaño (de las especificaciones) es uno de los factores involucrados en el esfuerzo.
- Los modelos matemáticos no pueden (hoy día) cuantificar esfuerzo sólo con medidas del tamaño.
- Todavía es necesario el “ojo clínico” del jefe de proyecto en las estimaciones (debido a su conocimiento histórico de los otros factores del entorno: productividad, etc.)

