

# Modelos de estimación para Proyectos de Software

## *Estimation models for Software Projects*

Dunia Suárez Ferreiro

Universidad de las Ciencias Informáticas

[dsuarezf@uci.cu](mailto:dsuarezf@uci.cu)

### **Resumen**

La estimación de proyectos de software constituye un componente muy importante dentro de la ingeniería de software pues la realización de estimaciones adecuadas conlleva a un desarrollo satisfactorio de los Proyectos de Software. En este artículo se realiza una revisión de los principales modelos de estimación existentes que puede servir de punto de partida para la selección de los modelos a utilizar en las estimaciones de costo, tiempo y esfuerzo en proyectos de software.

**Palabras Clave:** Costo, esfuerzo, estimación, métodos, tamaño del software.

### **Abstract**

*The estimation of software projects constitutes a main component of Software Engineer because good estimations ensure a satisfactory Software Projects development. This paper contains a revision of the principal estimation models for Software Projects that permits select appropriate models to estimate cost, time and effort in software projects.*

**Key words:** Cost, effort, estimation, methods, software size.

### **Introducción**

El primer componente de un proyecto de software es la planificación del proyecto. En esta etapa una de las tareas más importantes es la estimación, tanto de recursos como de tiempo, costo y esfuerzo para el satisfactorio desarrollo del proyecto. Debido a la importancia de la estimación se han motivado múltiples investigaciones orientadas a mejorar la comprensión del proceso de desarrollo de software así como a construir y evaluar herramientas de estimación de costos del software. Estimar los factores relacionados con un proyecto (esfuerzo, personal, cronograma, costo, etc.) requiere conocer o estimar su tamaño para evaluar las posibles soluciones, comparar alternativas y calcular costos antes de decidir por un enfoque determinado (Fillottrani 2007).

Dentro del marco de un proyecto de Software pueden existir diversas variables a estimar, como pueden ser la estimación del tiempo de duración de cada una de las tareas a realizar dentro del proyecto (análisis, diseño, gestión, desarrollo, pruebas, entre otras) o del proyecto en total, así como las estimaciones de los casos de prueba por casos de uso, los riesgos, el esfuerzo a realizar, entre varios índices estimables. Cualquier intención de calcular tiempos, recursos o costos para el futuro implica estimaciones.

Cuando se realiza la gestión y planificación de proyectos lo primero que se realiza es la estimación del tamaño del software para luego proceder a estimar el esfuerzo necesario, y los posibles cronogramas. La estimación de costos es una de las primeras actividades que se realiza en el proyecto, la mayoría de las veces después del establecimiento de los requerimientos, sin embargo se sigue aplicando con regularidad a lo largo de la ejecución del proyecto con el fin de ajustar la precisión en la estimación previa.

La estimación de por sí es difícil, una buena aproximación puede ser desarrollar incrementalmente los requerimientos, revisando estimaciones a medida que se va disponiendo de más información (desarrollo en espiral). La estimación temprana puede ocasionar problemas (producir riesgos), ya que la cantidad de información que se tiene en principio del sistema suele ser limitada, lo que dificulta este proceso.

Resulta muy útil tener bien claros los usos fundamentales de la estimación de costos en la administración de proyectos:

- En la etapa de planeamiento permite decidir la cantidad de personas que se necesitan para poder ejecutar el proyecto y establecer cronogramas adecuados.
- Durante el progreso del proyecto permite controlar su progreso y evaluar si el proyecto se está ejecutando de acuerdo al cronograma y, de no ser así, poder tomar medidas a tiempo. Para esto se requiere contar con métricas que permitan medir el nivel de cumplimiento del desarrollo del software.

En el ámbito de la ingeniería de software, la estimación de costos radica básicamente en estimar el tamaño del software así como la cantidad de personas necesarias para desarrollar el producto. A diferencia de otras disciplinas de la ingeniería, en las cuales, el costo de los materiales es el principal componente a ser estimado (Gómez, C.López et al. 1999).

Resulta innegable además, la fuerte relación existente entre costo, cronograma y calidad. Estos tres aspectos están íntimamente relacionados y confrontados entre sí. De esta manera, se hace difícil incrementar la calidad sin aumentar el costo y/o el cronograma del software a desarrollar. Similarmente, el cronograma de desarrollo no puede reducirse dramáticamente sin deteriorar la calidad del producto de software y/o incrementar el costo de desarrollo. Los modelos de estimación juegan un papel importante ya que permiten equilibrar estos tres factores (Gómez, C.López et al. 1999). Es muy necesario destacar que el proceso de estimación es de por sí incierto ya que se están tratando de medir elementos como el tamaño, costo y esfuerzo de un proyecto que aún no ha comenzado y del cual es muy probable que se tenga un escaso conocimiento, es por esto que es de vital importancia que el encargado de realizar la gestión del proyecto sea una persona con experiencia en el tema, que siempre tenga en cuenta trabajos pasados para no repetir los mismos problemas anteriores y con buena visión para prever posibles riesgos. Siempre hay que tener presente que la estimación en horas tempranas es muy riesgosa, por lo que hay que realizar constantemente revisiones y ajustes a las estimaciones ya existentes.

## **Desarrollo**

### **Estimación del tamaño del Software.**

Existen varias técnicas para medir el tamaño del Software, entre las cuales se encuentran:

1. Medición de líneas de código fuente producidas (SLOC).
2. Puntos de Función (FP).
3. Puntos de Casos de Usos (UCP).

#### **1. Medición de líneas de código fuente producidas (SLOC).**

Es una de las técnicas más conocidas pero requiere disponer del código lo que representa una restricción para realizar estimaciones tempranas, ya que la información que se tiene en esos momentos y el tiempo que se dispone para las primeras estimaciones del proyecto son generalmente escasos.

De todos los puntos en los cuales puede realizarse la estimación, cuando menos datos se tienen para hacerla es en el momento inicial, cuando todavía se está evaluando la factibilidad del proyecto. Desde el punto de vista del desarrollo y venta de software específico para terceros esa estimación temprana, es una de las más importantes.

Sin embargo, la actividad de estimación no se hace una sola vez en el proyecto, sino que a medida que se cuenta con más datos se hacen estimaciones más precisas que permiten una mejor planificación de lo que resta del proyecto. Luego con SLOC resultaría muy complejo realizar mediciones en horas tempranas de desarrollo del proyecto.

#### **2. Puntos de Función (FP).**

El Análisis de Puntos Función (FPA) provee un enfoque alternativo pues permite estimar el tamaño del software basándose en los requerimientos, luego la cantidad de SLOC se puede estimar desde el tamaño en Puntos Función (FP).

En etapas tempranas del ciclo de vida, se identifican los Actores y los Casos de Uso del sistema, y se documenta cada uno de ellos mediante una breve descripción. Aplicando el Análisis de Puntos de Función a estos Casos de Uso, se podrá obtener una

estimación a groso modo del tamaño y a partir de ella del esfuerzo (Peralta 2004). Más adelante, cuando se tenga más información acerca del sistema se podrá refinar este análisis.

Posteriormente se amplía la documentación de cada Caso de Uso, describiendo los Escenarios que se producen dentro del mismo. Un Escenario relata la secuencia de pasos que efectúan los actores y el sistema durante la ejecución del Caso de Uso. Si se aplica nuevamente el Análisis de Puntos de Función sobre estos Casos de Uso detallados, la estimación del tamaño y esfuerzo será más precisa que la anterior (Peralta 2004).

El FPA mide el tamaño del software cuantificando la funcionalidad provista al usuario basándose solamente en el diseño lógico y las especificaciones funcionales.

Este método consiste (brevemente) en calcular los puntos de función teniendo en cuenta:

- Entradas al sistema (entradas externas).
- Salidas del sistema (salidas externas).
- Consultas.
- Grupos de datos lógicos del sistema.
- Grupos de datos lógicos que no son del sistema pero que el sistema usa.

Debe computarse cuántas ocurrencias de cada parámetro contiene un sistema, calificándolos según su complejidad en alta, media y baja. Cada parámetro para una complejidad dada tiene un determinado peso, ese peso son los puntos de función asignados a ese parámetro (CAO. 2006). Luego de realizado este proceso los puntos de función se ajustan a los requerimientos del sistema.

La fórmula de Albretch (Albretch, 1979) para calcular los puntos función, es la siguiente:

$$FP = UFP \times TCF$$

Donde **UFP**: Puntos Función no Ajustados y **TCF**: Factor de Complejidad Técnica.

Los **UFP** son calculados mediante la sumatoria de los pesos de todos los parámetros identificados. Para mas nivel de detalle ver (Gómez, C.López et al. 1999).

Existen varias metodologías que estiman el tamaño del software utilizando este método como por ejemplo:

- IFPUG-FPA (Function Point Analysis).
- MKII (Mark II).
- FFP (Full Function Point).
- NESMA FPA (Netherlands Software Metrics Users Association Function Point Analysis).

Este método también posee inconvenientes en su aplicación en etapas tempranas de desarrollo del software, debido a las causas previamente mencionadas de la escasez de información del sistema en esos momentos.

### 3. Puntos de Casos de Usos (UCP).

El método UCP clasifica los casos de uso en simple, promedio y complejo con factores de peso 5, 10 y 15 respectivamente. La clasificación se hace en base al número de transacciones que contiene el caso de uso, 1 a 3 para simple, 4 a 7 para promedio y 8 ó más para complejos. Por otra parte se define una transacción como un evento que ocurre entre un actor y el sistema a ser modelado (CAO. 2006).

Luego de haber especificado los casos de uso del sistema ya se tiene un mejor nivel de detalle para poder estimar de manera más exacta los puntos de función, posteriormente, con las transacciones de puntos de función (Entradas Externas, Salidas Externas, Consultas Externas) generados por las secuencias que componen un escenario, es posible determinar la complejidad de Archivos Lógicos Internos o de Interfaz Externos (Peralta 2004).

A continuación se describe de manera general el procedimiento para determinar Puntos Objeto (Casos de Uso) en un proyecto de software (Gómez, C.López et al. 1999):

1. Determinar Cantidad de Objetos: Estimar la cantidad de pantallas, reportes, componentes que contendrá la aplicación.

2. Clasificar cada instancia de un objeto según sus niveles de complejidad (simple, media o difícil).
3. Dar el peso a cada objeto según el nivel de complejidad. Los pesos reflejan el esfuerzo relativo requerido para implementar una instancia de ese nivel de complejidad.
4. Determinar la cantidad de Puntos Objeto, sumando todos los pesos de las instancias de los tipos de objetos especificados.

Solo resaltar el detalle de que los casos de uso no tienen un tamaño determinado, por lo que un método basado en casos de uso debería tener en cuenta el número de transacciones o escenarios del mismo para determinar su peso. A medida que se van completando las especificaciones de los casos de uso se pueden ir refinando las estimaciones de los puntos de función.

### **Estimación de esfuerzo (cantidad de personas necesarias).**

Para calcular el posible esfuerzo en proyecto de software se utiliza la medida de meses/persona (PM), lo que representa los meses de trabajo de una persona en el desarrollo del proyecto. Existen varios métodos de estimación del esfuerzo como son:

1. Puntos de Función Ajustados y Coeficientes de Conversión.
2. Estimación del esfuerzo a partir de los Casos de Uso.
3. Modelo Composición de Aplicación.
4. Modelo Diseño Temprano.
5. Modelo Post-Arquitectura.
6. COCOMO II.

#### **1. Puntos de Función Ajustados y Coeficientes de Conversión.**

Los Puntos de Función Ajustados se obtienen como el producto de los Puntos de Función sin ajustar por el Factor de Ajuste:

$FP = UFP \times AF$  , donde UFP son los puntos de función sin ajustar y AF es el factor de ajuste.

Para ajustar los puntos de función previamente calculados hay que evaluar cada una de las características del sistema a tener en cuenta con su factor de peso asociado, el cual indica la importancia de la característica asociada que se está analizando.

Para más nivel de detalle remitirse a (Peralta 2004).

Luego de calculados los puntos de función ajustados se pueden aplicar coeficientes que permitan estimar el tiempo, costo y esfuerzo del proyecto.

#### **2. Estimación del esfuerzo a partir de los Casos de Uso.**

El esfuerzo en horas-hombre viene dado por (Peralta 2004):

$E = UCP \times CF$  , donde

**E:** esfuerzo estimado en horas-hombre

**UCP:** Puntos de Casos de Uso ajustados

**CF:** factor de conversión

#### **3. Modelo Composición de Aplicación.**

Para este modelo la fórmula que se propone es la siguiente:

$$PM = NOP / PROD$$

Donde:

**NOP** (Nuevos Puntos Objeto): Tamaño del nuevo software a desarrollar expresado en Puntos Objeto y se calcula de la siguiente manera:

$NOP = OP \times (100 - \% \text{ reuso}) / 100$  , donde **OP** (Puntos Objeto): Tamaño del software a desarrollar expresado en Puntos Objeto y **%reuso**: Porcentaje de reuso que se espera lograr en el proyecto.

**PROD:** Es la productividad promedio determinada a partir del análisis de datos de proyectos en (Banker, 1994).

#### **4. Modelo de diseño temprano.**

Este modelo se utiliza, como su nombre lo indica, en etapas tempranas del proyecto de software cuando se conoce muy poco del sistema a desarrollar. El mismo emplea para su cálculo Multiplicadores de Esfuerzo (Gómez, C.López et al. 1999) los cuales se clasifican en categorías, estos son:

- Del Producto  
**RCPX:** Confiabilidad y Complejidad del producto  
**RUSE:** Reusabilidad Requerida
- De la Plataforma  
**PDIF:** Dificultad de la Plataforma
- Del Personal  
**PERS:** Aptitud del Personal  
**PREX:** Experiencia del Personal
- Del Proyecto  
**FCIL:** Facilidades  
**SCED:** Cronograma de Desarrollo Requerido

### **5. Modelo Post-Arquitectura.**

Este modelo utiliza:

- Puntos Función y/o Líneas de Código Fuente para estimar tamaño, con modificadores que contemplan el reuso, con y sin traducción automática, y el "desperdicio" (breakage).
- Un conjunto de 17 atributos, denominados factores de costo, que permiten considerar características del proyecto referentes al personal, plataforma de desarrollo, etc., que tienen injerencia en los costos.
- Cinco factores que determinan un exponente, que incorpora al modelo el concepto de deseconomía y economía de escala.

**Ver (Gómez, C.López et al. 1999).**

### **6. COCOMO II.**

Cocoma II está compuesto por tres de los modelos previamente descritos, Composición de Aplicación, Diseño Temprano y Post-Arquitectura. El modelo Composición de Aplicación se emplea en desarrollos de software durante la etapa de prototipación, el modelo Diseño Temprano se utiliza en las primeras etapas del desarrollo en las cuales se evalúan las alternativas de hardware y software de un proyecto y el modelo Post-Arquitectura se aplica en la etapa de desarrollo, después que se define la arquitectura del sistema, y en la etapa de mantenimiento.

#### **Estimación de Cronogramas (tiempo).**

La estimación de cronogramas (calendario) consiste en estimar el número de períodos de trabajo (horas, días, semanas) que serán necesarios para completar cada actividad dentro del proyecto.

Existen varias técnicas para la estimación de la duración de un proyecto (autores 2007):

- Juicio de expertos: utilizar la experiencia histórica de las personas.
- Estimación por analogía: basarse en la estimación realizada con anterioridad para una actividad similar (del mismo o de otro proyecto). Es un caso especial de juicio de expertos.
- Simulación: se basa en calcular múltiples estimaciones partiendo de diferentes conjuntos de asunciones. El método más común es el Análisis de Monte Carlo: a partir de una distribución de resultados probables para cada actividad se calcula una distribución de resultados probables para todo el proyecto.
- Métodos matemáticos: proporcionan estimaciones directas aplicando técnicas estadísticas o fórmulas matemáticas (lineales o no lineales) sobre ciertos datos empíricos: ejemplo COCOMO.

Siempre hay que tener en cuenta que es necesario ajustar los cronogramas a medida que el proyecto de software se va desarrollando.

### **Modelos de estimación para proyectos de software en la Universidad de las Ciencias Informáticas (UCI)**

En la Universidad de las Ciencias Informáticas existe una gran cantidad de proyectos productivos a desarrollar, sin embargo la universidad es relativamente joven (5 años) en materia de desarrollo de proyectos. A pesar de que se cuenta con el apoyo de personal altamente calificado, aún es insuficiente la experiencia de los líderes y desarrolladores que trabajan en los proyectos de software en la universidad, ya que en su mayoría tienen poco tiempo de graduados o son estudiantes. A pesar de esto es necesario organizar y realizar una planificación del proyecto lo más correcta posible, pues de la misma depende el éxito de las posteriores etapas de desarrollo.

A partir de los diferentes modelos de estimación previamente analizados se puede llegar a la conclusión que, debido a la falta de experiencia del personal y al corto tiempo de vida de la universidad, resulta necesario realizar las estimaciones utilizando la mayor cantidad posible de indicadores, ya que no existe una buena cantidad de proyectos ya concluidos que permitan acumular experiencias y estadísticas para proceder de una mejor manera a la estimación.

Dadas las características antes descritas de la universidad, para la estimación del tamaño del software uno de los modelos que más se utiliza en nuestro entorno es el de estimación por casos de uso, siempre teniendo en cuenta el número de transacciones o escenarios que incluye el modelo del proyecto para poder determinar correctamente los pesos asociados. El modelo por puntos de función, de forma similar, estima el tamaño del software a partir de los requerimientos, lo que puede tener ciertas limitantes debido a la poca información que se tiene del sistema a desarrollar en etapas tan tempranas, no obstante esta primera estimación puede refinarse posteriormente, a medida que avanza el proyecto, con el modelo de estimación por casos de uso. Luego, desde mi punto de vista, resultaría conveniente que el planificador del proyecto realizara una primera estimación en horas tempranas del desarrollo del proyecto, dígase a partir del levantamiento de requisitos, y que posteriormente (una vez se tenga diseñado el modelo de casos de usos) se refine la estimación previa utilizando el modelo de estimación por casos de usos.

No obstante estas consideraciones, la selección del modelo (o los modelos) de estimación a utilizar son decisión del planificador, en conjunto con el arquitecto principal y el líder del proyecto, teniendo en cuenta las características del proyecto en cuestión. El presente artículo puede servir de referencia a la hora de realizar dicha selección.

### **Conclusiones**

El tema de la estimación en los proyectos de software resulta de vital importancia debido a su aporte al satisfactorio desarrollo de los proyectos. En el presente artículo se realiza una revisión de los modelos de estimación existentes para proyectos de software, mostrando las principales características de cada uno de estos modelos que permiten la estimación de costo, tiempo y esfuerzo.

Es bueno destacar, como conclusión, que la selección de los modelos de estimación a utilizar varía de acuerdo a las características propias del proyecto y a la experiencia y apreciación de los responsables (planificador - líder) de realizar esta tarea, no obstante se torna necesaria la revisión de los diversos modelos posibles a utilizar para la estimación, en aras de una adecuada selección que garantice resultados favorables.

### **Referencias Bibliográficas**

- Autores, C. d. (2007). "Guía de aprendizaje - TEMA 6 Gestión del tiempo en Ingeniería del Software."
- BERTOLAMI, M. *Estimación del Tamaño Funcional en la Elicitación de Requerimientos*, 2004.
- CAO., M. I. J. I. (2006). Principios para un método de estimación de proyectos de software basado en los escenarios principales.

Fillottrani, P. R. (2007). Calidad en el Desarrollo de Software. Métricas de procesos de software.

Gómez, A., M. d. C.López, et al. (1999). –COCOMO- UN MODELO DE ESTIMACION DE PROYECTOS DE SOFTWARE.

Peralta, M. (2004). "Estimación del esfuerzo basada en casos de uso."

PRESSMAN, R. S. Ingeniería del Software. Un enfoque práctico.

---