

Modelo COCOMO (Constructive Cost Model)

COCOMO es un modelo paramétrico desarrollado por Barry Boehm en 1981 para estimar el costo, esfuerzo y tiempo de desarrollo de software basado en el tamaño del proyecto en términos de Kilo Líneas de Código (KLOC). Se han desarrollado varias versiones del modelo, siendo COCOMO II la más reciente.

Cálculo: $\text{Esfuerzo(en persona-meses)} = A \times (\text{KLOC})^B \times \text{Factor de ajuste}$

Parámetros:

- A: Constante de ajuste basada en el tipo de proyecto (orgánico, semi-independiente o empotrado).
- B: Exponente que refleja la economía de escala.
- Factor de ajuste: Basado en diversos atributos del proyecto como la fiabilidad, complejidad, y experiencia del equipo.

Procedimiento:

- Calcular KLOC (líneas de código).
- Identificar el tipo de proyecto y seleccionar valores de A y B.
- Ajustar el esfuerzo con los factores multiplicadores.
- Calcular el esfuerzo total y, a partir de ahí, el costo.

Análisis del Valor Ganado (EVM - Earned Value Management)

EVM es una técnica que integra el alcance del proyecto, el cronograma y el costo para medir el rendimiento del proyecto. Proporciona un marco para evaluar el progreso del proyecto y prever posibles desviaciones en el costo y el cronograma.

Cálculo:

- PV (Planned Value): Valor del trabajo planificado hasta la fecha actual.
- EV (Earned Value): Valor del trabajo realmente realizado hasta la fecha actual.
- AC (Actual Cost): Costo real incurrido hasta la fecha actual.
- Fórmulas:
 - CPI (Cost Performance Index): $\text{CPI} = \text{EV} / \text{AC}$
 - EAC (Estimate at Completion): $\text{EAC} = \text{BAC} / \text{CPI}$
 - BAC (Budget at Completion): Presupuesto total planificado para el proyecto.

Procedimiento:

- Calcular PV, EV, y AC.
- Evaluar el desempeño del proyecto utilizando CPI y EAC.
- Hacer ajustes en la planificación del proyecto en base a los resultados de EVM.

Modelo de Puntos de Función (Function Points)

Los Puntos de Función (FP) miden la funcionalidad ofrecida por el software desde el punto de vista del usuario final. Este método estima el tamaño del software basado en la cantidad de entradas, salidas, consultas, archivos internos y archivos externos que maneja.

$$FP = \sum (Entradas, Salidas, Consultas, Archivos Internos, Archivos Externos) \times \text{Peso}$$

Procedimiento:

- Identificar y contar los elementos funcionales: Entradas, Salidas, Consultas, Archivos Internos y Externos.
- Asignar un peso a cada tipo de elemento basado en su complejidad (simple, promedio, complejo).
- Ajustar el conteo de puntos de función con factores de complejidad del proyecto.
- Calcular el costo estimado multiplicando los puntos de función por el costo por punto de función.

Modelo de Estimación de Costo de Software por Analogía

Este método se basa en comparar el proyecto de software actual con proyectos similares desarrollados previamente. El costo estimado se basa en los costos históricos de proyectos anteriores con ajustes por diferencias en tamaño, complejidad y tecnología.

$$\text{Costo Estimado} = \text{Costo Proyecto Anterior} \times \text{Factor de Ajuste}$$

Procedimiento:

- Seleccionar un proyecto similar ya completado.
- Ajustar el costo del proyecto anterior teniendo en cuenta diferencias en tamaño (líneas de código o puntos de función), complejidad, y tecnología.
- Realizar el ajuste final para considerar factores específicos del entorno de desarrollo actual.

Modelo Delphi

El método Delphi es una técnica de estimación colaborativa que recopila las opiniones de un grupo de expertos para estimar el costo del proyecto. Es particularmente útil en proyectos con gran incertidumbre.

Procedimiento:

- **Primera Ronda:** Cada experto proporciona una estimación anónima basada en su experiencia.
- **Consolidación:** Las estimaciones se recopilan y se comparten con todo el grupo, destacando las diferencias significativas.
- **Segunda Ronda:** Los expertos revisan sus estimaciones iniciales en base a la retroalimentación del grupo.
- **Iteración:** El proceso se repite hasta que se alcanza un consenso o las estimaciones convergen.

Referencias

- Boehm, B. W. (1981). Software Engineering Economics. Prentice-Hall.
- Albrecht, A. J., & Gaffney, J. E. (1983). Software function, source lines of code, and development effort prediction: A software science validation. IEEE Transactions on Software Engineering, 9(6), 639-648.
- Karner, G. (1993). Metrics for Objectory (Diploma thesis). University of Linköping, Sweden.
- Dalkey, N., & Helmer, O. (1963). An experimental application of the Delphi method to the use of experts. Management Science, 9(3), 458-467.
- Jensen, R. W. (1983). A Comparison of Two Software Cost Estimating Models. In Proceedings of the AFIPS National Computer Conference (Vol. 52, pp. 667-675).