

## Práctica 3: Estimación



Componentes.

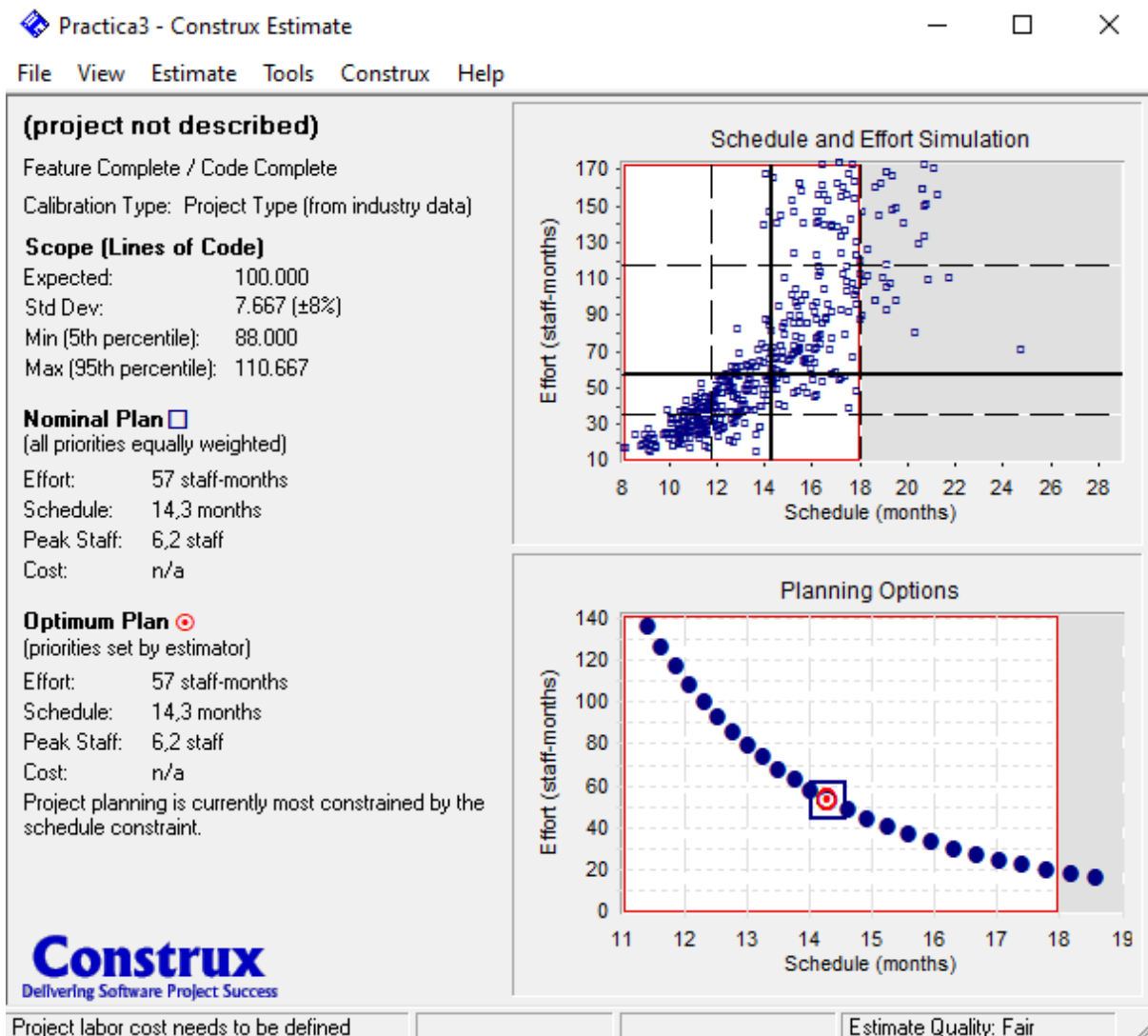
- Arturo Cortés Sánchez
- Abel José Sánchez Alba
- David Castro Salazar

Para desarrollar esta práctica hemos utilizado la herramienta construx, el método COCOMO y el método Putnam. Vamos a comparar y decidir cual es el método que mejor nos puede venir para realizar esta práctica.

## 1. CONSTRUX

Construx es una herramienta que es fácil de usar y que además es rápida ya que el proceso que propone consiste en responder preguntas y al final te saca una estimación. Por otro lado se puede considerar que la versión que estamos usando puede estar un poco desactualizada y que las estimaciones puede que no sean muy exactas.

Debido a la facilidad de uso del programa hemos decidido usar construx para realizar la estimación de nuestro proyecto.



## 2. Método COCOMO

Para realizar el método COCOMO hemos usado la herramienta de la página web <http://softwarecost.org/tools/COCOMO/>. Esta herramienta es más difícil de entender de cara a introducir los datos y eso hace que se tarde más tiempo en realizar la estimación. Además, una vez terminada hemos tenido que ir reajustando algunos datos que le introducimos. Pero la forma de mostrar los datos de salida de las estimaciones es muy gráfica y hace que se comprenda mucho mejor que las otras. Se podría decir que la visualización de resultados es mejor que las demás. Aun así, personalmente creemos que esta no sería la herramienta que usamos para este proyecto.

**COCOMO II - Constructive Cost Model**

Software Size      Sizing Method

**SLOC**      % Design Modified      % Code Modified      % Integration Required      Assessment and Assimilation (0% - 8%)      Software Understanding (0% - 50%)      Unfamiliarity (0-1)

New       Reused       Modified

**Software Size Probability Distribution**

**Software Scale Drivers**

Precedentedness       Architecture / Risk Resolution       Process Maturity

Development Flexibility       Team Cohesion

**Software Cost Drivers**

**Product**

Required Software Reliability       Data Base Size       Product Complexity       Developed for Reusability       Documentation Match to Lifecycle Needs

**Personnel**

Analyst Capability       Programmer Capability       Personnel Continuity       Application Experience       Platform Experience       Language and Toolset Experience

**Platform**

Time Constraint       Storage Constraint       Platform Volatility

**Project**

Use of Software Tools       Multisite Development       Required Development Schedule

Maintenance       Annual Change Size (ESLOC)       Maintenance Duration (Years)       Software Understanding (0%-50%)       Unfamiliarity (0-1)

**Software Labor Rates**

Cost per Person-Month (Dollars)

---

**Results**

**Software Development (Elaboration and Construction)**

Effort = 45.6 Person-months  
Schedule = 16.8 Months  
Cost = \$182405

Total Equivalent Size = 40000 SLOC  
Effort Adjustment Factor (EAF) = 0.28

**Acquisition Phase Distribution**

Phase	Effort (Person-months)	Schedule (Months)	Average Staff	Cost (Dollars)
Inception	2.7	2.1	1.3	\$10944
Elaboration	10.9	6.3	1.7	\$43777
Construction	34.7	10.5	3.3	\$138629
Transition	5.5	2.1	2.6	\$21889

**Staffing Profile**

**Software Effort Distribution for RUP/MBASE (Person-Months)**

Phase/Activity	Inception	Elaboration	Construction	Transition
Management	0.4	1.3	3.5	0.8
Environment/CM	0.3	0.9	1.7	0.3
Requirements	1.0	2.0	2.8	0.2
Design	0.5	3.9	5.5	0.2
Implementation	0.2	1.4	11.8	1.0
Assessment	0.2	1.1	8.3	1.3
Deployment	0.1	0.3	1.0	1.6

**Maintenance**

Annual Maintenance Effort = 0.8 Person-Months  
Annual Maintenance Cost = \$3287  
Total Maintenance Cost = \$1643

**Acquisition Monte Carlo Results**

**Software Effort Distribution Function**

**Software Effort Confidence Levels**

Confidence Level	Effort (Person-Months)
10%	35
20%	38
30%	41
40%	43
50%	45
60%	47
70%	49
80%	51
90%	54
100%	71

**Maintenance Monte Carlo Results**

**Software Engineering Maintenance Effort Distribution Function**

## Estimación según el modelo de Putnam

El modelo de Putnam es un modelo empírico de estimación de esfuerzo que se basa en la siguiente fórmula:

$$E = [LOC \cdot B^{0,333} / P]^3 \cdot (1/t^4)$$

Para un proyecto de 100000 líneas de código tenemos y año y medio tenemos los siguientes parámetros de entrada:

B=0.39  
LOC=100000  
P=12000  
T=1.5

Sustituyendo obtenemos que el esfuerzo es: E= 45 personas-año.

Por lo que el coste es

$$4000 \text{ €/mes} \cdot 18 \text{ meses} \cdot 45 \text{ personas/año} = 324000 \text{ € Persona/año}$$

Se trata de un método bastante rápido, pero los resultados que da son bastante diferentes de los resultados obtenidos con otros métodos. El coste de 324000 € *Persona/año* es con diferencia el más alto en comparación con COCOMO y Construx Estimate. Además, como los otros métodos están implementados en un programa con interfaz, son capaces de ofrecer diagramas para visualizar mejor los resultados, lo cual deja al modelo Putnam “manual” por debajo de ellos. Es posible que existan interfaces gráficas para este tipo de estimación, pero no las conocemos.

La diferencia entre estimaciones viene dada por el parámetro P del modelo de putnam, que indica la productividad, y ésta varía en función del ámbito del proyecto. Al desarrollar nuestro grupo un proyecto en varios ámbitos y haber seleccionado el ámbito de software científico como principal, pero no único del proyecto, se realiza una sobreestimación por norma general.

### 3. Comparativa de resultados

	Construx Estimate	COCOMO	Putnam
Esfuerzo	57 personas-mes	45,6 personas-mes	45 personas-año
Coste	-	182405\$	324000 € Persona/año
Duración	14,3 meses	16,8 meses	18 meses (Introducido por el usuario)

