CS 430 - Lecture 22 - Planning and Estimation II - Intermediate COCOMO

Collin Roberts

November 28, 2023

Outline

1. Estimating Duration and Cost
    1. Techniques for Cost Estimation
    2. Intermediate COCOMO
    3. COCOMO II
    4. Tracking Duration and Cost Estimates

Techniques for Cost Estimation

Definition 1

KDSI *stands for* **Thousand Delivered Source Instructions** *(i.e. 1000s of Lines of Code).*

Techniques for Cost Estimation

Remarks:

1. There is **no perfect technique** for estimating the cost/duration of a S/W project.
2. Some factors to consider:
   1. skill levels of project personnel (including familiarity with the S/W product)
   2. complexity of project
   3. project deadlines
   4. target hardware
   5. availability of CASE tools

Techniques for Cost Estimation

3. **Techniques of Estimation**
   1. Expert Judging by Analogy
      1. experts using history of similar past projects.
   2. Bottom-Up Approach
      1. analogous to **divide and conquer**, and
      2. most common in my SLF experience.
   3. Algorithmic Cost Estimation Models (e.g. COCOMO)
      1. Compute the **size** of the S/W product, using **function points**, or some other method.
      2. Use the size of the S/W product from 1 to estimate cost & duration of the project to build it.

Intermediate COCOMO

1. COCOMO comprises three models (highest level → lowest level):
   1. macroestimation
   2. intermediate (what we use here)
   3. microestimation
2. Two stages in Intermediate COCOMO: estimate each of
   1. **nominal effort**
   2. **estimated effort**

Intermediate COCOMO

Example like on pp278-280 in the text:

1. Compute the **nominal effort** for a software product having
   1. **organic** development mode (with multiplier 3.2 as in the text),
   2. **exponent** 1.07, (different from the 1.05 used in the text) and
   3. 12,000 LOC (i.e. 12 KDSI).

Intermediate COCOMO

Solution:

$$\begin{aligned} \text{nominal effort} &= 3.2(KDSI)^{1.07}\text{person-months} \\ &= 3.2(12)^{1.07} \\ &\approx 45.69555028, \end{aligned}$$

and so we state the nominal effort as 46 person-months. (Only whole numbers make sense here; we **always round up** to be conservative.)

Intermediate COCOMO

2. Use part a) to compute the **estimated effort**, using the given multipliers for each cost driver in Figure 9.6 from the text (reproduced here).

Intermediate COCOMO

Figure 9.6 - Intermediate COCOMO software development effort multipliers

Cost Drivers	Rating: Very Low	Low	Nominal	High	Very High	Extra High
Product Attributes						
-Required software reliability	0.75	0.88	1.00	1.15	1.40	
-Database size		0.94	1.00	1.08	1.16	
-Product complexity	0.70	0.85	1.00	1.15	1.30	1.65
Computer Attributes						
-Execution time constraint			1.00	1.11	1.30	1.66
-Main storage constraint			1.00	1.06	1.21	1.56
-Virtual machine volatility		0.87	1.00	1.15	1.30	
-Computer turnaround time		0.87	1.00	1.07	1.15	
Personnel Attributes						
-Analyst capabilities	1.46	1.19	1.00	0.86	0.71	
-Applications experience	1.29	1.13	1.00	0.91	0.82	
-Programmer capability	1.42	1.17	1.00	0.86	0.70	
-Virtual machine experience	1.21	1.10	1.00	0.90		
-Programming language experience	1.14	1.07	1.00	0.95		
Project Attributes						
-Use of modern programming practices	1.24	1.10	1.00	0.91	0.82	
-Use of software tools	1.24	1.10	1.00	0.91	0.83	
-Required development schedule	1.23	1.08	1.00	1.04	1.10	

Intermediate COCOMO

multipliers to use

Cost Drivers	Rating to Use
Product Attributes	
-Required software reliability	Nominal
-Database size	Low
-Product complexity	Low
Computer Attributes	
-Execution time constraint	High
-Main storage constraint	Nominal
-Virtual machine volatility	Low
-Computer turnaround time	Nominal
Personnel Attributes	
-Analyst capabilities	Very High
-Applications experience	Very High
-Programmer capability	High
-Virtual machine experience	Low
-Programming language experience	High
Project Attributes	
-Use of modern programming practices	Nominal
-Use of software tools	Nominal
-Required development schedule	High

Intermediate COCOMO

Solution: The effort multipliers for the given drivers are:

multipliers to use

Cost Drivers	Rating to Use	Multiplier to Use
Product Attributes		
-Required software reliability	Nominal	1.00
-Database size	Low	0.94
-Product complexity	Low	0.85
Computer Attributes		
-Execution time constraint	High	1.11
-Main storage constraint	Nominal	1.00
-Virtual machine volatility	Low	0.87
-Computer turnaround time	Nominal	1.00
Personnel Attributes		
-Analyst capabilities	Very High	0.71
-Applications experience	Very High	0.82
-Programmer capability	High	0.86
-Virtual machine experience	Low	1.10
-Programming language experience	High	0.95
Project Attributes		
-Use of modern programming practices	Nominal	1.00
-Use of software tools	Nominal	1.00
-Required development schedule	High	1.04

Intermediate COCOMO

Using the given effort multipliers gives

$$\begin{aligned} & (1.00)(0.94)(0.85) \\ & (1.11)(1.00)(0.87)(1.00) \\ & (0.71)(0.82)(0.86)(1.10)(0.95) \\ & (1.00)(1.00)(1.04)46 \\ \approx\ & 19.31377308, \end{aligned}$$

and so we state the estimated effort as 20 person-months. (Only whole numbers make sense here; we **always round up** to be conservative.)

COCOMO II

COCOMO was introduced in 1981 (before OO was widely accepted; most systems were mainframe-based; classical paradigm was prevalent), and it became less reliable as time went on.

COCOMO II

COCOMO II was a major revision to address these weaknesses.

1. COCOMO is all based on LOC (equivalently KDSI)
2. 3 applications of COCOMO II:
   1. application composition model
   2. early design model
   3. post architecture model
3. Where COCOMO outputs a single estimate, COCOMO II outputs a range of estimates for each model.

COCOMO II

4. When I have taught CS 430 in the past, I have made a note to myself to present COCOMO II instead of Intermediate COCOMO, because we make the case throughout the course that we should adopt the OO paradigm.
5. However I found that doing this was not practical. I have posted a .pdf detailing COCOMO II on LEARN. Please peruse it at your leisure.

COCOMO II

6. You may also find the following web pages about COCOMO II interesting:
   1. Overview: `http://sunset.usc.edu/csse/research/cocomoii/cocomo_main.html`
   2. Calculator: `http://csse.usc.edu/tools/COCOMOII.php`

COCOMO II

We don't have time to go into the details of COCOMO II in CS 430. See the text for references for additional reading if interested.

Tracking Duration and Cost Estimates

Key Ideas:

1. It is extremely rare for a S/W project to be completed ahead of schedule and under budget. Deviations from estimates usually make the project late and over budget.
2. Hence it is critical to detect deviations from our estimates ASAP, so that we can take **immediate** corrective action.