(SPM)

MS. SHWETA TIWARI

Published: April 2, 2022

SPM: SOFTWARE PROJECT MANAGEMENT

SPM: UNIT-2 02/17

<u>TOPIC On: COCOMO (Constructive Cost Estimation Model)</u>

2 APRIL 2022 / IT-3rd year, VIth semester

FALL SEMESTER, YEAR (VIth, 3rd)

FALL SESSION (2021-22)

COCOMO (Constructive Cost Estimation Model)

Boehm proposed COCOMO (Constructive Cost Estimation Model) in 1981.COCOMO is one of the most generally used software estimation models in the world. COCOMO predicts the efforts and schedule of a software product based on the size of the software.

The necessary steps in this model are:

Get an initial estimate of the development effort from evaluation of thousands of delivered lines of source code (KDLOC).
Determine a set of 15 multiplying factors from various attributes of the project.
Calculate the effort estimate by multiplying the initial estimate with all the multiplying factors i.e., multiply the values in step1 and step2.
The initial estimate (also called nominal estimate) is determined by an equation of the form used in the static single variable models, using KDLOC as the measure of the size. To determine the initial effort Ei in person-months the equation used is of the type is shown below

$E_i = a * (KDLOC)^b$

The value of the constant a and b depends on the project type.

In COCOMO, projects are categorized into three types:

Organic
Semidetached
Embedded

1.Organic: A development project can be treated of the organic type, if the project deals with developing a well-understood application program, the size of the development team is reasonably small, and the team members are experienced in developing similar methods of projects. Examples of this type of projects are simple

business systems, simple inventory management systems, and data processing systems.

2. Semi Detached: A development project can be treated with semi detached type if the development consists of a mixture of experienced and inexperienced staff. Team members may have finite experience in related systems but may be unfamiliar with some aspects of the order being developed. Examples of Semi Detached systems include developing a new operating system (OS), a Database Management System (DBMS), and complex inventory management system.

3. Embedded: A development project is treated to be of an embedded type, if the software being developed is strongly coupled to complex hardware, or if the stringent regulations on the operational method exist. For Example: ATM, Air Traffic control.

For three product categories, Bohem provides a different set of expression to predict effort (in a unit of person month)and development time from the size of estimation in KLOC(Kilo Line of code) efforts estimation takes into account the productivity loss due to holidays, weekly off, coffee breaks, etc.

According to Boehm, software cost estimation should be done through three stages:

Basic Model
Intermediate Model
Detailed Model

1. Basic COCOMO Model: The basic COCOMO model provides an accurate size of the project parameters. The following expressions give the basic COCOMO estimation model:

$$\text{Effort} = a_1 * (KLOC)^{a_2} \text{ PM}$$
$$\text{Tdev} = b_1 * (\text{efforts})^{b_2} \text{ Months}$$

Where

KLOC is the estimated size of the software product indicate in Kilo Lines of Code,

a_1, a_2, b_1, b_2 are constants for each group of software products,

Tdev is the estimated time to develop the software, expressed in months,

Effort is the total effort required to develop the software product, expressed in person months (PMs).

Estimation of development effort

For the three classes of software products, the formulas for estimating the effort based on the code size are shown below:

Organic: Effort = $2.4(KLOC)^{1.05}$ PM

Semi-detached: Effort = $3.0(KLOC)^{1.12}$ PM

Embedded: Effort = $3.6(KLOC)^{1.20}$ PM

**Estimation of development time**

For the three classes of software products, the formulas for estimating the development time based on the effort are given below:

Organic: Tdev = 2.5(Effort) 0.38 Months

Semi-detached: Tdev = 2.5(Effort) 0.35 Months

Embedded: Tdev = 2.5(Effort) 0.32 Months

Some insight into the basic COCOMO model can be obtained by plotting the estimated characteristics for different software sizes. Fig shows a plot of estimated effort versus product size. From fig, we can observe that the effort is somewhat superliner in the size of the software product. Thus, the effort required to develop a product increases very rapidly with project size.

Effort versus product size
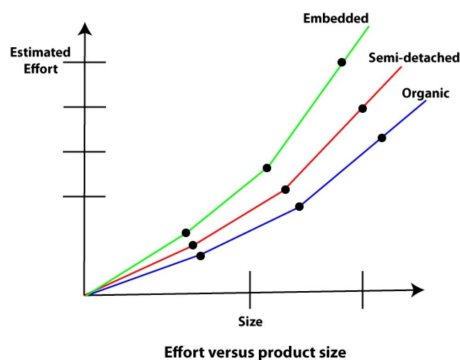
From the effort estimation, the project cost can be obtained by multiplying the required effort by the manpower cost per month. But, implicit in this project cost computation is the assumption that the entire project cost is incurred on account of the manpower cost alone. In addition to manpower cost, a project would incur costs due to hardware and software required for the project and the company overheads for administration, office space, etc.

It is important to note that the effort and the duration estimates obtained using the COCOMO model are called a nominal effort estimate and nominal duration estimate. The term nominal implies that if anyone tries to complete the project in a time shorter than the estimated duration, then the cost will increase drastically. But, if anyone completed the project over a longer period of time than the estimated, then there is almost no decrease in the estimated cost value.

Example1: Suppose a project was estimated to be 400 KLOC. Calculate the effort and development time for each of the three models i.e., organic, semi-detached & embedded.

Solution: The basic COCOMO equation takes the form:

Effort=a1*(KLOC) a2 PM
Tdev=b1*(efforts)b2 Months
Estimated Size of project= 400 KLOC

(i)Organic Mode

E = 2.4 * (400)1.05 = 1295.31 PM
D = 2.5 * (1295.31)0.38=38.07 PM

(ii)Semi Detached Mode

E = 3.0 * (400)1.12=2462.79 PM
D = 2.5 * (2462.79)0.35=38.45 PM

(iii) Embedded Mode

E = 3.6 * (400)1.20 = 4772.81 PM
D = 2.5 * (4772.8)0.32 = 38 PM

Example2: A project size of 200 KLOC is to be developed. Software development team has average experience on similar type of projects. The project schedule is not very tight. Calculate the Effort, development time, average staff size, and productivity of the project.

Solution: The semi detached mode is the most appropriate mode, keeping in view the size, schedule and experience of development time.

Hence $E = 3.0(200)1.12 = 1133.12 PM$

$D = 2.5(1133.12)0.35 = 29.3 PM$

COCOMO Model

$P = 176 \text{ LOC/PM}$

Average Staff Size (SS) $= \frac{E}{D}$ Persons

$$= \frac{1133.12}{29.3} = 38.67 \text{ Persons}$$

Productivity $= \frac{KLOC}{E} = \frac{200}{1133.12} = 0.1765 \text{ KLOC/PM}$

2. Intermediate Model: The basic Cocomo model considers that the effort is only a function of the number of lines of code and some constants calculated according to the various software systems. The intermediate COCOMO model recognizes these facts and refines the initial estimates obtained through the basic COCOMO model by using a set of 15 cost drivers based on various attributes of software engineering.

Classification of Cost Drivers and their attributes:

(i) Product attributes -
- Required software reliability extent
- Size of the application database
- The complexity of the product

Hardware attributes -
- Run-time performance constraints
- Memory constraints
- The volatility of the virtual machine environment
- Required turnabout time

Personal attributes -
- Analyst capability
- Software engineering capability
- Applications experience
- Virtual machine experience
- Programming language experience

Project attributes -
- Use of software tools
- Application of software engineering methods
- Required development schedule

Intermediate COCOMO equation:

$$E = a_i (KLOC) b_i * EAF$$
$$D = c_i (E) d_i$$

Coefficients for intermediate COCOMO

Project	a_i	b_i	c_i	d_i
Organic	2.4	1.05	2.5	0.38
Semidetached	3.0	1.12	2.5	0.35
Embedded	3.6	1.20	2.5	0.32

3. Detailed COCOMO Model: Detailed COCOMO incorporates all qualities of the standard version with an assessment of the cost driver's effect on each method of the software engineering process. The detailed model uses various effort multipliers for each cost driver property. In detailed cocomo, the whole software is differentiated into multiple modules, and then we apply COCOMO in various modules to estimate effort and then sum the effort.

The Six phases of detailed COCOMO are:

Planning and requirements
System structure
Complete structure
Module code and test
Integration and test
Cost Constructive model
The effort is determined as a function of program estimate, and a set of cost drivers are given according to every phase of the software lifecycle.