

Tutorial 05: Software Cost Estimation using COCOMO-I / COCOMO-II for Project.

COCOMO (Constructive Cost Model) is a regression model based on LOC, i.e number of Lines of Code. It is a procedural cost estimate model for software projects and often used as a process of reliably predicting the various parameters associated with making a project such as size, effort, cost, time and quality. It was proposed by Barry Boehm in 1970 and is based on the study of 63 projects, which make it one of the best-documented models.

The key parameters which define the quality of any software products, which are also an outcome of the Cocomo are primarily Effort & Schedule:

- **Effort:** Amount of labor that will be required to complete a task. It is measured in person-months units.
- **Schedule:** Simply means the amount of time required for the completion of the job, which is, of course, proportional to the effort put. It is measured in the units of time such as weeks, months.

Different models of Cocomo have been proposed to predict the cost estimation at different levels, based on the amount of accuracy and correctness required. All of these models can be applied to a variety of projects, whose characteristics determine the value of constant to be used in subsequent calculations. These characteristics pertaining to different system types are mentioned below.

Boehm's definition of organic, semidetached, and embedded systems:

1. **Organic** – A software project is said to be an organic type if the team size required is adequately small, the problem is well understood and has been solved in the past and also the team members have a nominal experience regarding the problem.
2. **Semi-detached** – A software project is said to be a Semi-detached type if the vital characteristics such as team-size, experience, knowledge of the various programming environment lie in between that of organic and Embedded. The projects classified as Semi-Detached are comparatively less familiar and difficult to develop compared to the organic ones and require more experience and better guidance and creativity. Eg: Compilers or different Embedded Systems can be considered of Semi-Detached type.
3. **Embedded** – A software project with requiring the highest level of complexity, creativity, and experience requirement fall under this category. Such software requires a larger team size than the other two models and also the developers need to be sufficiently experienced and creative to develop such complex models.

All the above system types utilize different values of the constants used in Effort Calculations.

Types of Models: COCOMO consists of a hierarchy of three increasingly detailed and accurate forms. Any of the three forms can be adopted according to our requirements. These are types of COCOMO model:

1. Basic COCOMO Model
2. Intermediate COCOMO Model
3. Detailed COCOMO Model

The first level, **Basic COCOMO** can be used for quick and slightly rough calculations of Software Costs. Its accuracy is somewhat restricted due to the absence of sufficient factor considerations.

Intermediate COCOMO takes these Cost Drivers into account and **Detailed COCOMO** additionally accounts for the influence of individual project phases, i.e in case of Detailed it accounts for both these cost drivers and also calculations are performed phase wise henceforth producing a more accurate result. These two models are further discussed below.

Estimation of Effort:

Calculations –

1) Basic Model –

The above formula is used for the cost estimation of for the basic COCOMO model, and also is used in the subsequent models. The constant values a and b for the Basic Model for the different categories of system:

| SOFTWARE PROJECTS | A | B |
|-------------------|-----|------|
| ORGANIC | 2.4 | 1.05 |
| SEMI DETACHED | 3.0 | 1.12 |
| EMBEDDED | 3.6 | 1.20 |

- The effort is measured in Person-Months and as evident from the formula is dependent on Kilo-Lines of code. These formulas are used as such in the Basic Model calculations, as not much consideration of different factors such as reliability, expertise is taken into account, henceforth the estimate is rough.

Intermediate Model – The basic Cocomo model assumes that the effort is only a function of the number of lines of code and some constants evaluated according to the different software system. However, in reality, no system's effort and schedule can be solely calculated on the basis of Lines of Code. For that, various other factors such as reliability, experience, Capability. These factors are known as Cost Drivers and the Intermediate Model utilizes 15 such drivers for cost estimation. Classification of Cost Drivers and their attributes:

(i) **Product attributes –**

- Required software reliability extent
- Size of the application database
- The complexity of the product

(ii) **Hardware attributes –**

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

(iii) **Personnel attributes –**

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

(iv) **Project attributes –**

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

| Cost Drivers | Ratings | | | | | |
|-------------------------------|----------|------|---------|------|-----------|------------|
| | Very Low | Low | Nominal | High | Very High | Extra High |
| Product attributes | | | | | | |
| Required software reliability | 0.75 | 0.88 | 1.00 | 1.15 | 1.40 | |
| Size of application database | | 0.94 | 1.00 | 1.08 | 1.16 | |

| | | | | | | |
|---|------|------|------|------|------|------|
| Complexity of the product | 0.70 | 0.85 | 1.00 | 1.15 | 1.30 | 1.65 |
| Hardware attributes | | | | | | |
| Run-time performance constraints | | | 1.00 | 1.11 | 1.30 | 1.66 |
| Memory constraints | | | 1.00 | 1.06 | 1.21 | 1.56 |
| Volatility of the virtual machine environment | | 0.87 | 1.00 | 1.15 | 1.30 | |
| Required turnabout time | | 0.87 | 1.00 | 1.07 | 1.15 | |
| Personnel attributes | | | | | | |
| Analyst capability | 1.46 | 1.19 | 1.00 | 0.86 | 0.71 | |
| Applications experience | 1.29 | 1.13 | 1.00 | 0.91 | 0.82 | |
| Software engineer 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 | | | | | | |

| | | | | | | |
|---|------|------|------|------|------|--|
| Application of software engineering methods | 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 | |

The project manager is to rate these 15 different parameters for a particular project on a scale of one to three. Then, depending on these ratings, appropriate cost driver values are taken from the above table. These 15 values are then multiplied to calculate the EAF (Effort Adjustment Factor). The Intermediate COCOMO formula now takes the form:

$$E = (a(KLOC)^b) * EAF$$

The values of a and b in case of the intermediate model are as follows:

| SOFTWARE PROJECTS | A | B |
|-------------------|-----|------|
| ORGANIC | 2.4 | 1.05 |
| SEMI DETACHED | 3.0 | 1.12 |
| EMBEDDED | 3.6 | 1.20 |

Detailed Model – Detailed COCOMO incorporates all characteristics of the intermediate version with an assessment of the cost driver's impact on each step of the software engineering process. The detailed model uses different effort multipliers for each cost driver attribute. In detailed cocomo, the whole software is divided into different modules and then we apply COCOMO in different modules to estimate effort and then sum the effort. The Six phases of detailed COCOMO are: 1. Planning and requirements 2. System design 3. Detailed design 4. Module code and test 5. Integration and test 6. Cost Constructive model The effort is calculated as a function of program size and a set of cost drivers are given according to each phase of the software lifecycle.

The Six phases of detailed COCOMO are:

1. Planning and requirements
2. System design
3. Detailed design
4. Module code and test
5. Integration and test
6. Cost Constructive model

The effort is calculated as a function of program size and a set of cost drivers are given according to each phase of the software lifecycle.

Calculations:

| | A | b | c | d |
|----------------------|----------|----------|----------|----------|
| Organic | 2.4 | 1.05 | 2.5 | 0.38 |
| Semi-Detached | 3.0 | 1.12 | 2.5 | 0.35 |
| Embedded | 3.6 | 1.20 | 2.5 | 0.32 |

Formulae:

$$\text{Effort} = a(\text{KLOC})^b \text{ person-month}$$

$$\text{Development} = c(\text{KLOC})^d \text{ months}$$

$$\text{Average Staff Cycle} = \text{Effort} / \text{Development} \text{ persons}$$

$$\text{Productivity} = (\text{KLOC} / \text{Effort}) * 1000 \text{ no. of lines of code}$$

$$\text{No. of lines of code} = 16515 = 16.515 \text{ KLOC}$$

○ **Organic**

$$\text{Effort} = 2.4 * (16.515)^{1.05} = 45.601 \text{ person-month}$$

$$\text{Development} = 2.5 * (45.601)^{0.38} = 10.674 \text{ months}$$

$$\text{Average Staff Cycle} = 45.601 / 10.674 = 4.272 \text{ persons}$$

$$\text{Productivity} = (16.515 / 45.601) * 1000 = 362.163 \text{ no. of lines of code}$$

○ **Semi- Detached**

$$\text{Effort} = 3.0 * (16.515)^{1.12} = 69.365 \text{ person-month}$$

$$\text{Development} = 2.5 * (69.365)^{0.35} = 11.023 \text{ months}$$

$$\text{Average Staff Cycle} = 69.365 / 11.023 = 6.292 \text{ persons}$$

$$\text{Productivity} = (16.515 / 69.365) * 1000 = 238.088 \text{ no. of lines of code}$$

○ **Embedded**

$$\text{Effort} = 3.6 * (16.515)^{1.20} = 104.173 \text{ person-month}$$

$$\text{Development} = 2.5 * (104.173)^{0.32} = 11.056 \text{ months}$$

$$\text{Average Staff Cycle} = 104.173 / 11.056 = 9.422 \text{ persons}$$

$$\text{Productivity} = (16.515 / 104.173) * 1000 = 158.534 \text{ no. of lines of code}$$

| | Organic | Semi-Detached | Embedded |
|----------------------------|------------------------------|------------------------------|------------------------------|
| Effort | 45.601 person-month | 69.365 person-month | 104.173 person-month |
| Development | 10.674 months | 11.023 months | 11.056 months |
| Average Staff Cycle | 4.272 persons | 6.292 persons | 9.422 persons |
| Productivity | 362.163 no. of lines of code | 238.088 no. of lines of code | 158.534 no. of lines of code |