# Software Cost Estimation

For any new software project, it is necessary to know how much it will cost to develop and how much development time will it take. These estimates are needed before development is initiated, but how is this done? Several estimation procedures have been developed and are having the following attributes in common.

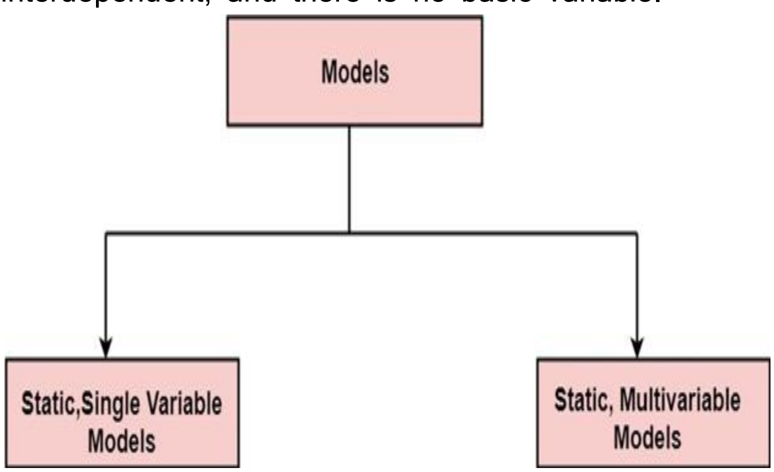
1. Project scope must be established in advanced.
2. Software metrics are used as a support from which evaluation is made.
3. The project is broken into small PCs which are estimated individually.
   To achieve true cost & schedule estimate, several option arise.
4. Delay estimation
5. Used symbol decomposition techniques to generate project cost and schedule estimates.
6. Acquire one or more automated estimation tools.

## Uses of Cost Estimation

1. During the planning stage, one needs to choose how many engineers are required for the project and to develop a schedule.
2. In monitoring the project's progress, one needs to access whether the project is progressing according to the procedure and takes corrective action, if necessary.

## Cost Estimation Models

A model may be static or dynamic. In a static model, a single variable is taken as a key element for calculating cost and time. In a dynamic model, all variable are interdependent, and there is no basic variable.



**Static, Single Variable Models:** When a model makes use of single variables to calculate desired values such as cost, time, efforts, etc. is said to be a single variable model. The most common equation is:

$\mathbf{C=aL^b}$

**Where** C = Costs

L= size

a and b are constants

The Software Engineering Laboratory established a model called SEL model, for estimating its software production. This model is an example of the static, single variable model.

$E=1.4L^{0.93}$

$DOC=30.4L^{0.90}$

$D=4.6L^{0.26}$

**Where** E= Efforts (Person Per Month)

DOC=Documentation (Number of Pages)

D = Duration (D, in months)

L = Number of Lines per code

**Static, Multivariable Models:** These models are based on method (1), they depend on several variables describing various aspects of the software development environment. In some model, several variables are needed to describe the software development process, and selected equation combined these variables to give the estimate of time & cost. These models are called multivariable models.

WALSTON and FELIX develop the models at IBM provide the following equation gives a relationship between lines of source code and effort:

$\mathbf{E=5.2L^{0.91}}$

In the same manner duration of development is given by

$\mathbf{D=4.1L^{0.36}}$

The productivity index uses 29 variables which are found to be highly correlated productivity as follows:

$$\mathbf{I} = \sum_{i=1}^{29} \mathbf{W_i}\,\mathbf{X_i}$$

Where $\mathbf{W_i}$ is the weight factor for the $\mathbf{i^{th}}$variable and $\mathbf{X_i=\{-1,0,+1\}}$ the estimator gives $\mathbf{X_i}$one of the values **-1, 0 or +1** depending on the variable decreases, has no effect or increases the productivity.

**Example:** Compare the Walston-Felix Model with the SEL model on a software development expected to involve 8 person-year of effort.

a. Calculate the number of lines of source code that can be produced.
b. Calculate the duration of the development.
c. Calculate the productivity in LOC/PY
d. Calculate the average manning

**Solution:**

The amount of manpower involved = 8PY=96persons-months

(a)Number of lines of source code can be obtained by reversing equation to give:

$$L = \left(\frac{E}{a}\right) 1/b$$

Then

L (SEL) = (96/1.4)1/0.93=94264 LOC

L (SEL) = (96/5.2)1/0.91=24632 LOC

(b)Duration in months can be calculated by means of equation

D (SEL) = 4.6 (L) 0.26

= 4.6 (94.264)0.26 = 15 months

D (W-F) = 4.1 $L^{0.36}$

= 4.1 (24.632)0.36 = 13 months

(c) Productivity is the lines of code produced per persons/month (year)

$$P\ (SEL) = \frac{94264}{8} = 11783\ \frac{LOC}{Person} - Years$$

$$P\ (Years) = \frac{24632}{8} = 3079\ \frac{LOC}{Person} - Years$$

(d)Average manning is the average number of persons required per month in the project

$$M\ (SEL) = \frac{96P-M}{15M} = 6.4 Persons$$

$$M\ (W\text{-}F) = \frac{96P-M}{13M} = 7.4 Persons$$

---

**Q. What is COCOMO? Explain COCOMO model in detail?**

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.

1. **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 -**

   4. **Basic Model - write formulae**

$$E = a_b (KLOC) b_{b},$$
$$D = C_b(E) d_b, \quad \text{and}$$
$$P = E/D$$

**where,**

**E is effort,**

**D is development time and**

**KLOC means kilo lines of code,**

**P is total number of persons require for the project.**

**The coefficients $a_b$ , $b_b$ , $c_b$ , $d_b$ for three modes are shown in table as:**

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,b,c and d for the Basic Model for the different categories of system:

| SOFTWARE PROJECTS | 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 |

The effort is measured in Person-Months and as evident from the formula is dependent on Kilo-Lines of code.

The development time is measured in Months.

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.

**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**

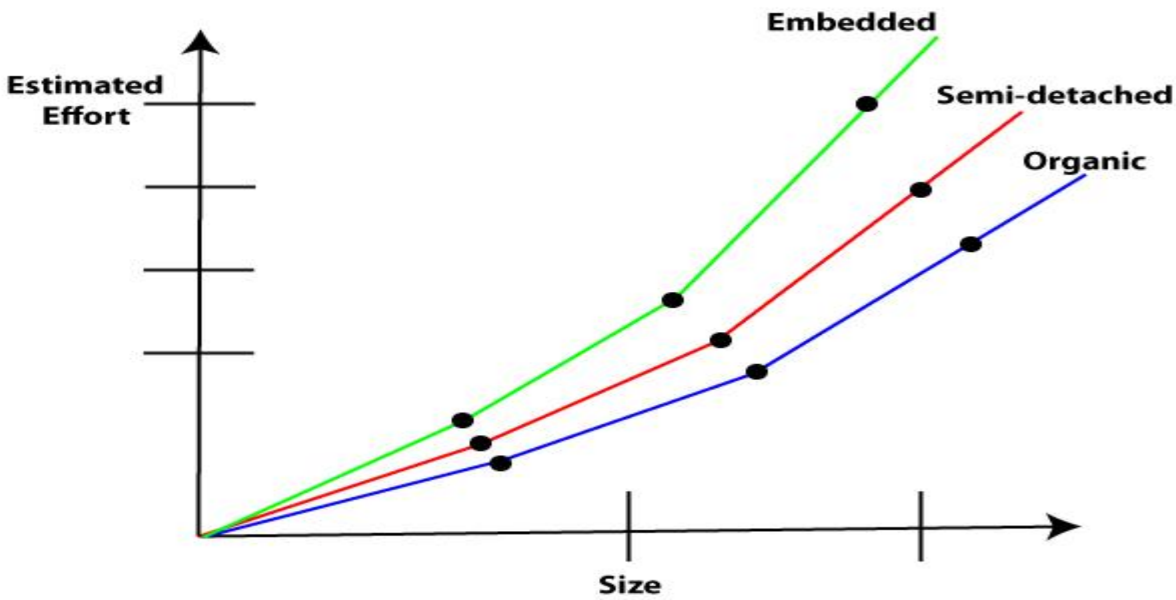
For the three classes of software products, the formulas for estimating the development time based on the effort are given below:

**Organic:** Tdev = 2.5(Effort) 0.38 Months

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

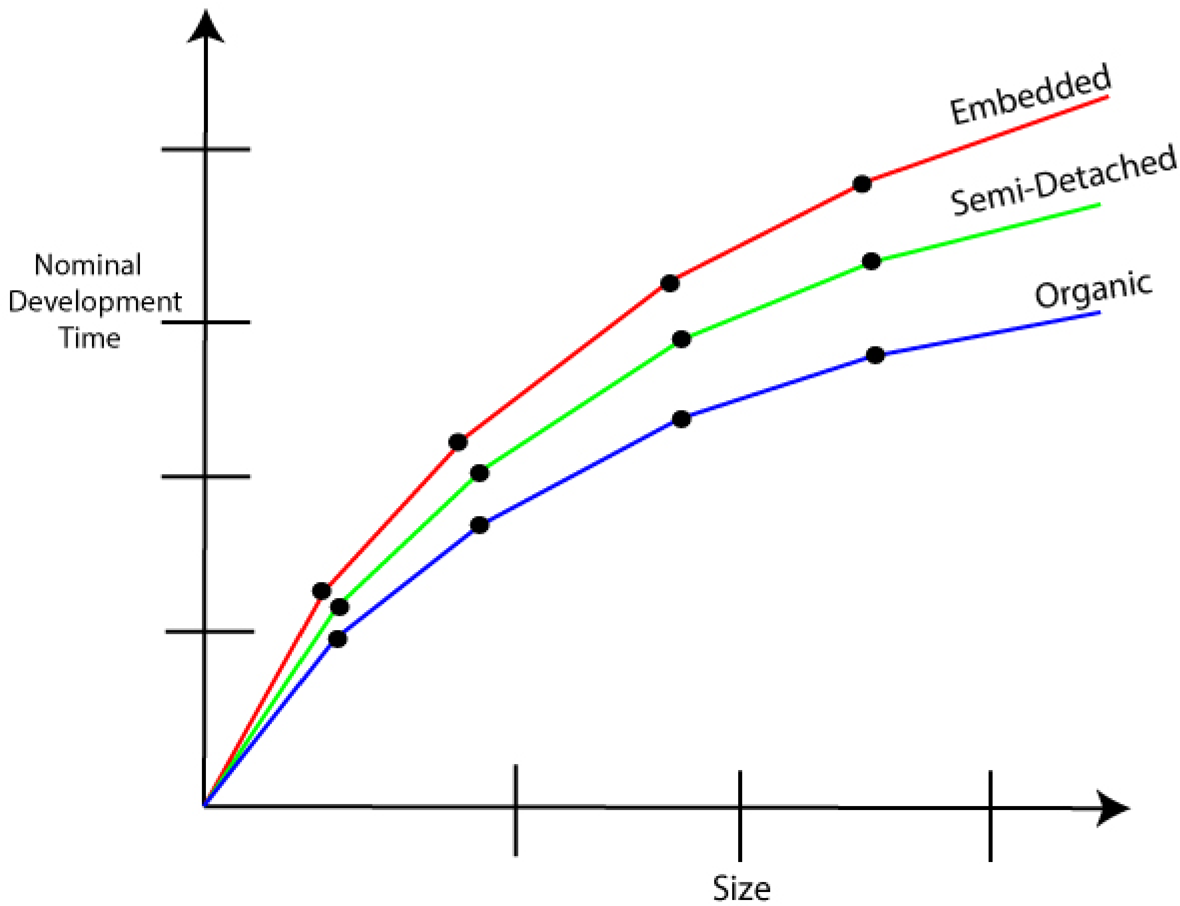
**Embedded:** Tdev = 2.5(Effort) 0.32 Months

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**

The development time versus the product size in KLOC is plotted in fig. From fig it can be observed that the development time is a sub linear function of the size of the product, i.e. when the size of the product increases by two times, the time to develop the product does not double but rises moderately. This can be explained by the fact that for larger products, a larger number of activities which can be carried out concurrently can be identified. The parallel activities can be carried out simultaneously by the engineers. This reduces the time to complete the project. Further, from fig, it can be observed that the development time is roughly the same for all three categories of products. For example, a 60 KLOC program can be developed in approximately 18 months, regardless of whether it is of organic, semidetached, or embedded type.




Development time versus 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 estimations 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 completes the project over a longer period of time than the estimated, then there is almost no decrease in the estimated cost value.

To illustrate the use of the COCOMO model, we apply the basic model to the CAD software example [described in SEPA, 5/e]. Using the LOC estimate and the coefficients noted in Table 5.1, we use the basic model to get:

$E = 2.4\ (KLOC)^{1.05}$

$= 2.4\ (33.2)^{1.05}$

= 95 person-months

This value is considerably higher than the estimates derived using LOC. Because the COCOMO model assumes considerably lower LOC/pm levels than those discussed in SEPA, 5/e, the results are not surprising. To be useful in the context of the example problem, the COCOMO model would have to be recalibrated to the local environment. To compute project duration , we use the effort estimate described above:

$D = 2.5\ E^{0.35}$

$= 2.5\ (95)^{0.35}$

= 12.3 months

The value for project duration enables the planner to determine a recommended number of people, *N,* for the project:

N = E/D

= 95/12.3

~ 8 people

In reality, the planner may decide to use only four people and extend the project duration accordingly.

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

**Solution:** The basic COCOMO equation takes the form:

$Effort=a_1*(KLOC)\ a_2$ PM

$Tdev=b_1*(efforts)b_2$ 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)Semidetached 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 semidetached mode is the most appropriate mode, keeping in view the size, schedule and experience of development time.

Hence E=3.0(200)1.12=1133.12PM

D=2.5(1133.12)0.35=29.3PM

Average staff size = E/D person-month= 1133.12/ 29.3 = 38.67

Productivity = KLOC/E = 200/1133.12 =0.1765 KLOC/PM

P = 176 LOC/PM

**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 | VERY LOW | LOW | NOMINAL | HIGH | VERY HIGH |
|---|---|---|---|---|---|
| **Product Attributes** | | | | | |
| 1.Required Software Reliability | 0.75 | 0.88 | 1.00 | 1.15 | 1.40 |
| 2.Size of Application Database | | 0.94 | 1.00 | 1.08 | 1.16 |
| 3.Complexity of The Product | 0.70 | 0.85 | 1.00 | 1.15 | 1.30 |
| **Hardware Attributes** | | | | | |
| 4.Runtime Performance Constraints | | | 1.00 | 1.11 | 1.30 |
| 5.Memory Constraints | | | 1.00 | 1.06 | 1.21 |
| 6.Volatility of the virtual machine environment | | 0.87 | 1.00 | 1.15 | 1.30 |
| 7.Required turnabout time | | 0.94 | 1.00 | 1.07 | 1.15 |
| **Personnel attributes** | | | | | |
| 8.Analyst capability | 1.46 | 1.19 | 1.00 | 0.86 | 0.71 |
| 9.Applications experience | 1.29 | 1.13 | 1.00 | 0.91 | 0.82 |
| 10Software engineer capability | 1.42 | 1.17 | 1.00 | 0.86 | 0.70 |
| 11.Virtual machine experience | 1.21 | 1.10 | 1.00 | 0.90 | |
| 12.Programming language experience | 1.14 | 1.07 | 1.00 | 0.95 | |
| **Project Attributes** | | | | | |
| 13.Application of software engineering | 1.24 | 1.10 | 1.00 | 0.91 | 0.82 |

| COST DRIVERS | VERY LOW | LOW | NOMINAL | HIGH | VERY HIGH |
|---|---|---|---|---|---|
| methods | | | | | |
| 14.Use of software tools | 1.24 | 1.10 | 1.00 | 0.91 | 0.83 |
| 15.Required development schedule | 1.23 | 1.08 | 1.00 | 1.04 | 1.10 |

1. 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:
2. $E = a(KLOC)^b\ (EAF)$
3. The values of a and b in case of the intermediate model are as follows:

| SOFTWARE PROJECTS | A | B |
|---|---|---|
| Organic | 3.2 | 1.05 |
| Semi Detached | 3.0 | 1.12 |
| Embeddedc | 2.8 | 1.20 |