**Introduction**

* **COnstructive COst MOdel** was introduced by **Dr. Barry Boehm’s** textbook **Software Engineering Economics**.
* This model is now generally called **“COCOMO 81”**.it refers to a group of models and is used to estimate the development efforts which are involved in a project.
* COCOMO is based upon the estimation of lines of code in a system and the time.COCOMO has also considered the aspects like project attributes, hardware, assessment of produce, etc.
* This provides transparency to the model which allows software managers to understand why the model gives the estimates it does. Moreover, the baseline **COCOMO originally underlies a waterfall model lifecycle.**
* The developments are done on multiple COCOMO models in parallel for cost estimates that cover a broader scope that exceeds the boundaries of traditional software development are discussed.
* The COCOMO model has basically two parameters like effort calculation and development time to define the quality of any software product:

Efforts calculation: Efforts can be calculated by the number of persons required to complete the task successfully. It is calculated in the unit person-month.

**Development time**: the time that is required to complete the task.it is calculated in the unit of time like months, weeks, and days. It depends on the effort calculation, if the number of persons is greater then definitely the development time is low.

Software projects under COCOMO model strategies are classified into 3 categories, **organic, semi-detached, and embedded.**

**Organic**: A software project is said to be an organic type if-

* Project is small and simple.
* Project team is small with prior experience.
* The  problem is well understood and has been solved in the past.
* Requirements of projects are not rigid, such a mode example is payroll processing system.

**Semi-Detached** **Mode:**  A software project is said to be a Semi-Detached type if-

* Project has complexity.
* Project team requires more experience,better guidance and creativity.
* The  project has an intermediate size and has mixed rigid requirements such a mode example is a transaction processing system which has fixed requirements.
* It also includes the elements of organic mode and embedded mode.
* Few such projects are- Database Management System(DBMS), new unknown operating system, difficult inventory management system.

**Embedded Mode:**A software project is said to be an Embedded mode type if-

* A software project has fixed requirements of resources .
* Product is developed within very tight constraints.
* A software project requiring the highest level of complexity, creativity, and experience requirement fall under this category.
* Such mode software requires a larger team size than the other two models.

**Types of COCOMO Models**

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

* Basic COCOMO Model
* Intermediate COCOMO Model
* Detailed COCOMO Model
* **Basic COCOMO Model**
* Basic COCOMO Model: The first level, Basic COCOMO can be used for quick and slightly rough
* calculations of Software Costs. This is because the model solely considers based on lines of source code together with constant values obtained from software project types rather than other factors which have major influences on the Software development process as a whole.
* It requires to calculate the efforts which are required to develop in three modes of development that are organic mode, semi-detached mode, and embedded mode.  
  The basic COCOMO estimation model is given by the following expressions:
* E = ax (KLOC)b  
  D = c x (Effort)d  
  P = effort/time
* **Where,**
* E is effort applied in person-months.  
  D is development time in months.  
  P is the total no. of persons required to accomplish the project.
* The constant values a,b,c, and d for the Basic Model for the different categories of the system

| **Software Project** | **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 |

* 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.
* **Example –** Consider a software project using semi-detached mode with 300 Kloc .find out effort estimation, development time, and person estimation.
* **Solution –**
* **Effort (E) =** a\*(KLOC)b = 3.0\*(300)1.12 = 1784.42 PM  
  **Development Time (D) =** c*(E)d = 2.5*(1784.42)0.35 = 34.35 Months(M)  
  **Person Required (P) = E/D =** 1784.42/34.35 = 51.9481 Persons ~52 Persons

### Intermediate COCOMO Model

Intermediate COCOMO model is an extension of the Basic COCOMO model which includes a set of cost drivers into account in order to enhance more accuracy to the cost estimation model as a result. The estimation model makes use of a set of “cost driver attributes” to compute the cost of the software. The estimated effort and scheduled time are given by the relationship:

**Effort (E) =** a\*(KLOC)b \*EAF PM  
**D =** c x (Effort)d

**Where,**

**E =** Total effort required for the project in Person-Months (MM).  
**KLOC =** The size of the code for the project in Kilo lines of code.  
**a, b =** The constant parameters for the software project.

**EAF =** It is an Effort Adjustment Factor, which is calculated by multiplying the parameter values of different cost driver parameters. For ideal, the value is 1.

#### Classification of Cost Drivers and their attributes:

**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

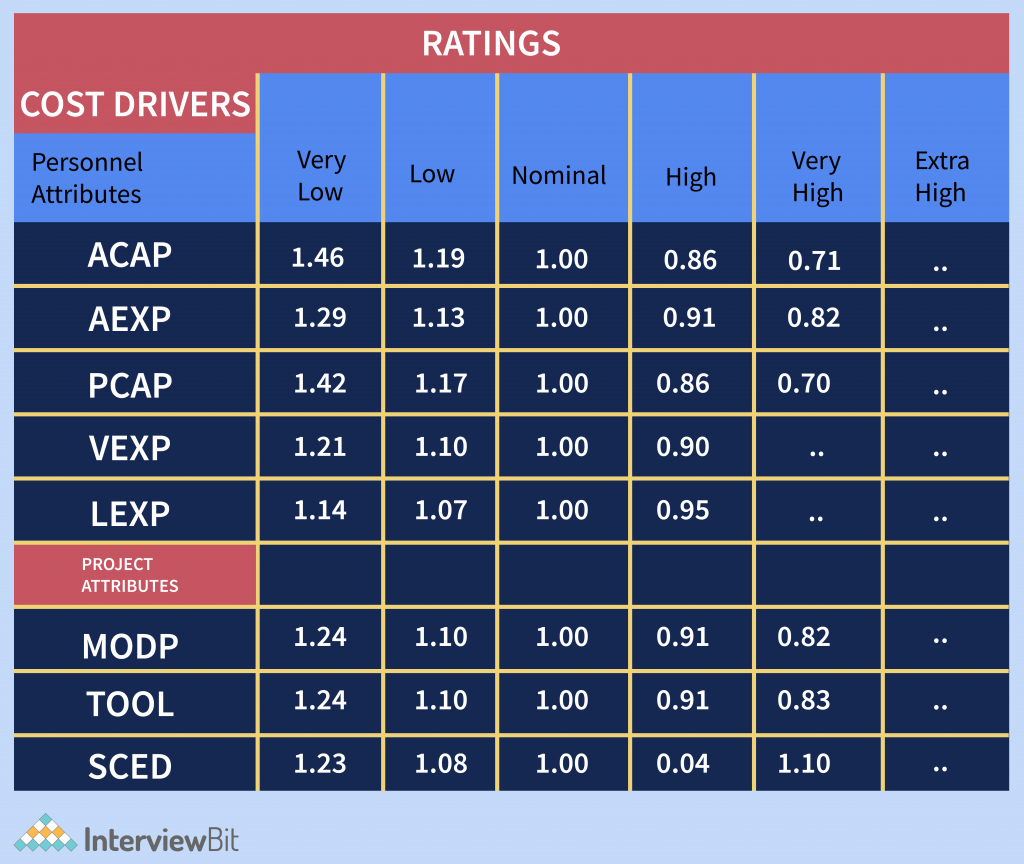
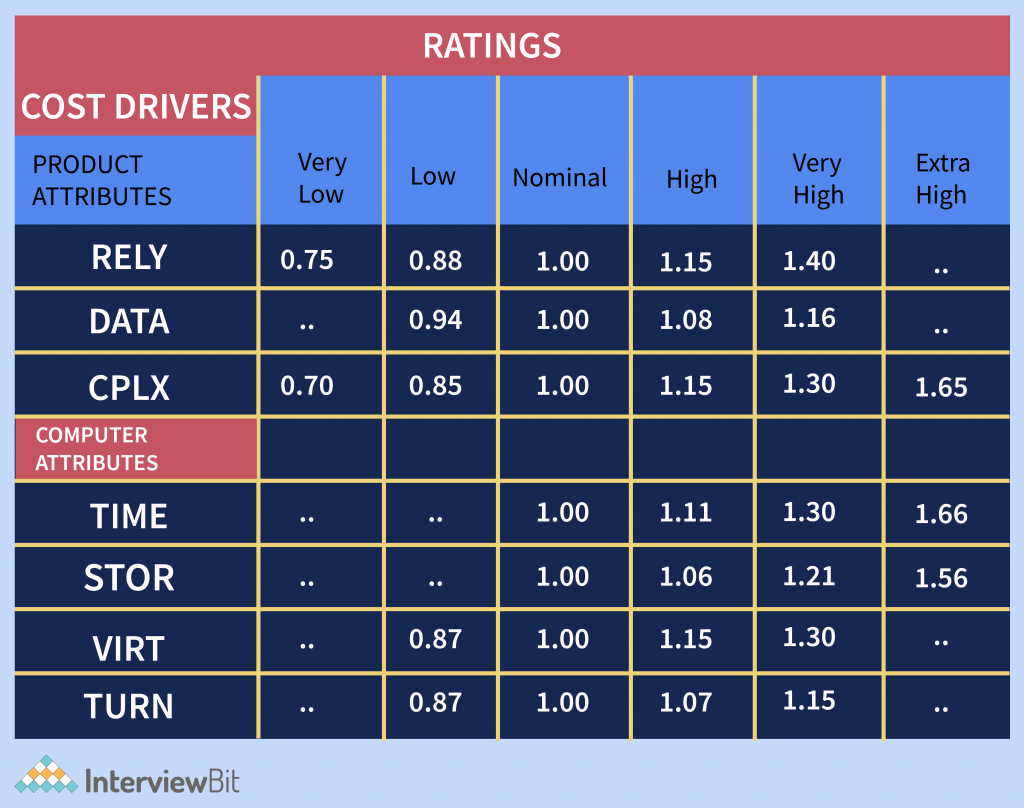
**Personal attributes –**

* 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

**The cost drivers are divided into four categories**



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

| **Software Project** | **A** | **B** | **C** | **D** |
| --- | --- | --- | --- | --- |
| Organic | 3.2 | 1.05 | 2.5 | 0.38 |
| Semi-Detached | 3.0 | 1.12 | 2.5 | 0.35 |
| Embedded | 2.8 | 1.20 | 2.5 | 0.32 |

**Example:** For a given project was estimated with a size of 300 KLOC. Calculate the Effort, Scheduled time for development by considering the developer having high application experience and very low experience in programming.

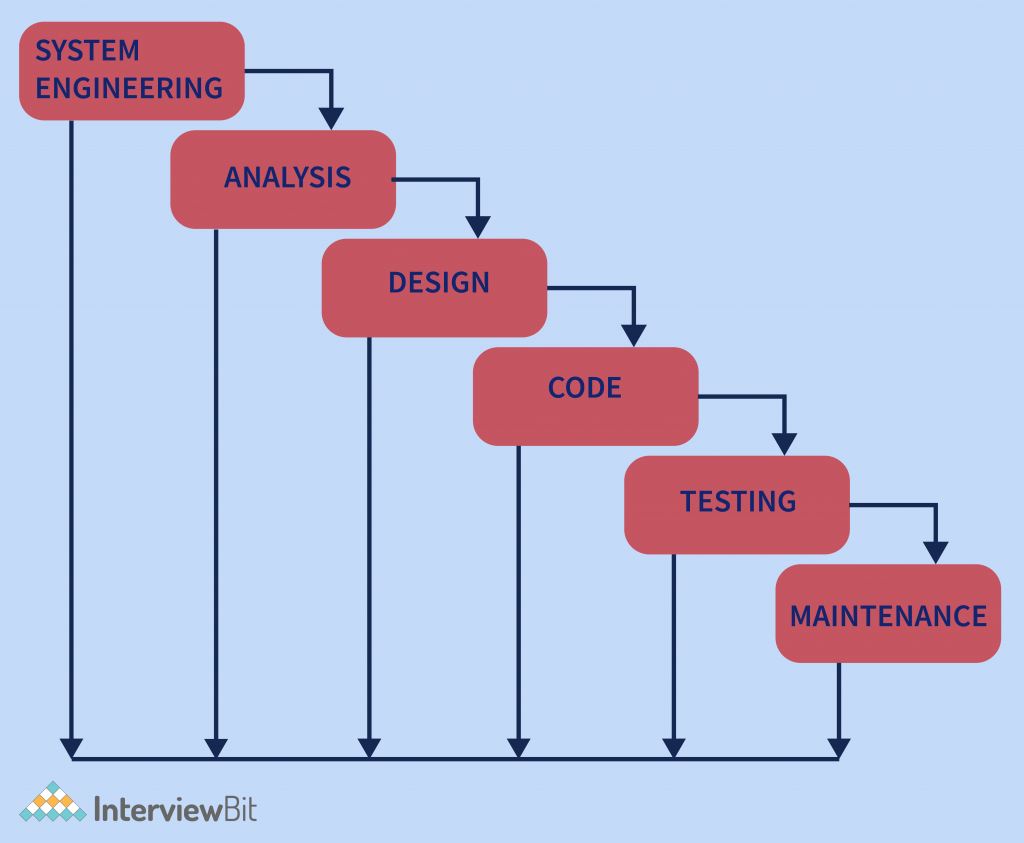
**Solution** –

Given the estimated size of the project is: 300 KLOC  
Developer having highly application experience: 0.82 (as per above table)  
Developer having very low experience in programming: 1.14(as per above table)

EAF = 0.821.14 = 0.9348 Effort (E) = a(KLOC)b EAF = 3.0(300)1.12 0.9348 = 1668.07 MM  
Scheduled Time (D) = c(E)d = 2.5\*(1668.07)0.35 = 33.55 Months(M)

### Detailed/Advanced COCOMO model

The model incorporates all qualities of both Basic COCOMO and Intermediate COCOMO strategies on each software engineering process. The model accounts for the influence of the individual development phase (analysis, design, etc.) of the project.it uses multipliers for each phase of the project .it is a complex mode.it includes more factors that influence the software projects and give more accurate estimations. In this, the whole software is divided into different modules and then we apply COCOMO in different modules to estimate effort and then sum the effort. For instance, detailed COCOMO will perform cost estimation in each development phase of the Waterfall Model-



The Six phases of detailed COCOMO are:

* Planning and requirements
* [System design](https://www.scaler.com/courses/system-design/?utm_source=ib&utm_medium=blog-inlink&utm_campaign=ib-inlinks-auto)
* Detailed design
* Module code and test
* Integration and test
* Cost Constructive model

In the Detailed COCOMO Model, the cost of each subsystem is estimated separately. This approach reduces the margin of error in the final estimate.

**Example:** A distributed Management Information System (MIS) product for an organization having offices at several places across the country can have the following sub-components:

* Database part
* Graphical User Interface (GUI) part
* Communication part

Of these, the communication part can be considered as Embedded software. The database part could be Semi-detached software, and the GUI part Organic software. The costs for these three components can be estimated separately, and summed up to give the overall cost of the system.

## Advantages of COCOMO Model

* COCOMO is transparent, one can see how it works unlike other models such as SLIM
* Drivers are particularly helpful to the estimator to understand the impact of different factors that affect project costs.
* COCOMO Provides ideas about historical projects.
* The COCOMO model is easy to estimate the total cost of the project.
* The drivers are very helpful to understand the impact of the different factors that affect project crises.

## Limitations of COCOMO Model

* It is hard to estimate KDSI early on in the project when most effort estimates are required.
* KDSI, actually, is not a size measure, it is a length measurement.
* Extremely vulnerable to misclassification of the development mode.
* Success depends largely on tuning the model to the needs of the organization, using historical data which is not always available.
* It limits the accuracy of software costs.
* It also ignores customer skills, cooperation, and knowledge.