COST CONSTRUCTIVE MODEL

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 is 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 1981 and is based on the study of 63 projects, which makes 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 in. It is measured in the units of time such as weeks, months.

This model of estimation is used for different types of Projects that are basically dived by Barry Boehm.Boehm's definition of organic, semidetached, and embedded systems are

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 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..

There are three types of models which are applied to different types of projects

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

1. Basic COCOMO :

These are basic formulae which are needed to be applied for calculating effort and time schedule

$$E(Effort) = a(KLOC)^b$$

$$Time = c(Effort)^d$$

$$Person\ Required = Effort/Time$$

The 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.

2. Intermediate COCOMO:

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 systems. 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 are:

- Required software reliability extent
- Size of the application database
- The complexity of the product
- Run-time performance constraints
- Memory constraints
- The volatility of the virtual machine environment
- Required turnabout time
- Analyst capability
- Software engineering capability
- Applications experience
- Virtual machine experience
- Programming language experience
- Use of software tools
- Application of software engineering methods
- Required development schedule

3. Detailed COCOMO:

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

COCOMO-II is the revised version of the original COCOMO (Constructive Cost Model) and is developed at University of Southern California. It is the model that allows one to estimate the cost, effort and schedule when planning a new software development activity.

It consists of three sub-models:

1. **End User Programming:**
   Application generators are used in this sub-model. End user write the code by using these application generators.
   Example – Spreadsheets, report generator, etc.
2. **Intermediate Sector :**

   (a). Application Generators and Composition Aids –
   This category will create largely prepackaged capabilities for user programming. Their product will have many reusable components. Typical firms operating in this sector are Microsoft, Lotus,Oracle, IBM, Borland, Novell.
   (b). Application Composition Sector –
   This category is too diversified and to be handled by prepackaged solutions. It includes GUI, Databases, domain specific components such as financial, medical or industrial process control packages.
   (c). System Integration –
   This category deals with large scale and highly embedded systems.

3. **Infrastructure Sector:**
   This category provides infrastructure for the software development like Operating System, Database Management System, User Interface Management System, Networking System, etc..

Stages of COCOMO II

1. **Stage-I:**
   It supports estimation of prototyping. For this it uses Application Composition Estimation Model. This model is used for the prototyping stage of application generator and system integration.
2. **Stage-II:**
   It supports estimation in the early design stage of the project, when we less

know about it. For this it uses Early Design Estimation Model. This model is used in early design stage of application generators, infrastructure, system integration.

3. **Stage-III:**
   It supports estimation in the post architecture stage of a project. For this it uses Post Architecture Estimation Model. This model is used after the completion of the detailed architecture of application generator, infrastructure, system integration.

Difference between COCOMOI and COCOMO II

COCOMO I is useful in the waterfall models of the software development cycle.	COCOMO II is useful in non-sequential, rapid development and reuse models of software.
It provides estimates of effort and schedule.	It provides estimates that represent one standard deviation around the most likely estimate.
This model is based upon the linear reuse formula.	This model is based upon the non linear reuse formula
This model is also based upon the assumption of reasonably stable requirements.	This model is also based upon reuse model which looks at effort needed to understand and estimate.
Effort equation's exponent is determined by 3 development modes.	Effort equation's exponent is determined by 5 scale factors.
Development begins with the requirements assigned to the software.	It follows a spiral type of development.
Number of submodels in COCOMO I is 3 and 15 cost drivers are assigned	In COCOMO II, Number of submodel are 4 and 17 cost drivers are assigned
Size of software stated in terms of	Size of software stated in terms of Object

Lines of code

points, function points and lines of code