



# COCOMO MODEL

**Estimating Efforts and Cost**

**Group-5**

# cocomo model

## Constructive Cost Model

An algorithm software cost estimation model developed by **Barry Boehm** published in 1981

### Applies on 3 classes of software projects :

- **Organic project**

- Small team
- Good experience
- Less rigid requirements

Example : Inventory management system

- **Semi-detached**

- Medium sized team
- Mixed experience
- Mixed rigidity

Example : DB design & OS development

- **Embedded project**

- Combination of organic & semi-detached

Example : Banking software or Traffic control software

# What are the types of COCOMO?

## 1. Basic cocomo:

- a. Static model
- b. Single-valued
- c. Computes development effort as function of size expressed in estimated LOC
- d. Estimates effort roughly
- e. Estimation accuracy less

## 1. Intermediate:

- a. Computes software development effort as a function of program size
- b. Cost drivers includes product, hardware, personnel, project attributes

# What are the types of COCOMO?

## Cost Drivers:

1. Product attributes
2. Hardware attributes
3. Personal attributes
4. Project attributes

## 3. Detailed Model :

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

# Number of user inputs

Those items provided by the user that describe individual application-oriented data (such as screen)

For our project we get 10 Inputs by counting

# Number of user outputs

Each user output that provides application-oriented information to the user is counted. (such as reports and messages, rather than the individual components of these)

For our project we get 6 Outputs by counting

# Number Of User Inquiries

An inquiry is defined as an on-line input that results in the generation of some immediate software response in the form of an on-line output. Each distinct inquiry is counted. For example : List of hostels .

# Number of files

Each logical master file. It's mainly Database . For example :

1. User Perspective
2. Room Offering
3. Payment
4. Supplier



# Number of external interfaces

All machine readable interfaces (e.g., data files on tape or disk) that are used to transmit information to another system are counted. For example :

**1.Google API**

**2.FB API**

**3.BKASH**

**4.DBBL**

# Step 1:

## Information Domain Values

Measurement Parameter	Count		Simple <input type="radio"/>	Average <input checked="" type="radio"/>	Complex <input type="radio"/>		Total
Number of user inputs	<input type="text" value="10"/>	X	3	4	6	=	<input type="text" value="40.00"/>
Number of user outputs	<input type="text" value="6"/>	X	4	5	7	=	<input type="text" value="30.00"/>
Number of user inquiries	<input type="text" value="8"/>	X	3	4	6	=	<input type="text" value="32.00"/>
Number of files	<input type="text" value="4"/>	X	7	10	15	=	<input type="text" value="40.00"/>
Number of external interfaces	<input type="text" value="4"/>	X	5	7	10	=	<input type="text" value="28.00"/>
Count=Total							<input type="text" value="170.00"/>

Count Total

# Step 2

## Complexity Weighting Factors

// heading of the second table Rate each factor on a scale of 0 to 5:

(0 = No influence, 1 = Incidental, 2 = Moderate, 3 = Average, 4 = Significant, 5 = Essential):

Question	0	1	2	3	4	5
1. Does the system require reliable backup and recovery?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
2. Are data communications required?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
3. Are there distributed processing functions?	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
4. Is performance critical?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
5. Will the system run in an existing, heavily utilized operational environment?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
6. Does the system require on-line data entry?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
7. Does the on-line data entry require the input transaction to be built over multiple screens or operations?	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
8. Are the master file updated on-line?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
9. Are the inputs, outputs, files, or inquiries complex?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
10. Is the internal processing complex?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
11. In the code designed to be reusable?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
12. Are conversion and installation included in the design?	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
13. Is the system designed for multiple installations in different organizations?	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
14. Is the application designed to facilitate change and ease of use by the user?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
<b>Total</b>						
40.00						

Show Total of weighting Factor

## Step 3: Calculating LOC

- The end user should select a programming language.
- Once the programming language is selected, then the end user can calculate the Line Of Code (LOC).

Programming Language	LOC/FP (average)	Select
Assembly Language	320	<input type="radio"/>
C	128	<input type="radio"/>
COBOL	105	<input type="radio"/>
Fortran	105	<input type="radio"/>
Pascal	90	<input type="radio"/>
Ada	70	<input type="radio"/>
Object-Oriented Languages	30	<input checked="" type="radio"/>
Fourth Generation Languages (4GLs)	20	<input type="radio"/>
Code Generators	15	<input type="radio"/>
Spreadsheets	6	<input type="radio"/>
Graphical Languages (icons)	4	<input type="radio"/>

**LOC/FP:**

# Step 4: Calculate the Effort & Duration

**Organic Model:** Small, simple software projects where a small but experienced team works to set of less rigid requirements.

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

$$D = 2.5 * (E)^{0.38}$$

**Semi-Detached Model:** An intermediate software model where teams with mixed experience must meet a mix of rigid and less than rigid requirements.

$$E = 3.0 * (KLOC)^{1.12}$$

$$D = 2.5 * (E)^{0.35}$$

**Embedded Model:** A software project which will must be developed within tight hardware, software & operational constraints.

$$E = 3.6 * (KLOC)^{1.20}$$

$$D = 2.5 * (E)^{0.32}$$

## Step 4 :

Software Project	$a_b$	$b_b$	$c_b$	$d_b$	Select
Organic	2.4	1.05	2.5	0.38	<input type="radio"/>
Semi-detached	3.0	1.12	2.5	0.35	<input checked="" type="radio"/>
Embedded	3.6	1.20	2.5	0.32	<input type="radio"/>

Calculate Effort and Duration

$$\text{Effort (E)} = a_b(\text{KLOC})^{b_b} = 19.65 \quad \text{Duration (D)} = c_b(E)^{d_b} = 7.09$$

Reset Data

Average staff size= (E/D)= (19.65/7.09) = **2.77** ~ 3.00 person-month.

If the salary of a developer = **30,000** BDT

Total cost including overhead will be = (30,000\*3)\*(2)\*(7.09)= **12,76,200** BDT

# Advantages Of COCOMO Model

- Easily Understandable.
- More predictable & accurate.
- The Drivers are very supportive to understand the impact on different factors that affect project costs.
- Accounts for various factors that affect cost of the project.

# Disadvantages of COCOMO Model

- Ignores Documentation & requirements.
- Dependent on the amount of time spent in each phase.
- Ignores skills ,co-operation, knowledge & parameters.
- Hardware requirements are denied.
- Personal turnover levels aren't used.



# Thank You