Name : G. Nithish

Reg No : 19BCS0012

Course : Software Engineering

Explain Basic COCOMO Model and Intermediate COCOMO Model with eg for each.

Ans:

According to Boehm, Software Cost estimation should be done through three stages:

i) Basic COCOMO Model

ii) Intermediate COCOMO model

iii) Detailed Model.

Now we will discuss first 2 model's in detail with example.

COCOMO Model:-

→ Boehm Proposed COCOMO (Constructive Cost Estimation Model) in 1981.

→ COCOMO is one of the most generally used software estimation model in the world. COCOMO predicts the efforts and schedule of a software product based on the size of the software.

The necessary steps in this model are:

1) Get an initial estimate of the developed effort from evaluation of thousands of delivered lines of source code (KDLOC).

2) Determine the set of 15 multiplying factors from various attributes of the project.

3) Calculate the effort estimate by multiplying the ~~result~~ initial estimate

with all multypling factors.

To determine the initial effort E_i in Person-months the equation used is of the type.

$$E_i = a^*(KDloc)^b$$

the values of the constant "a" and "b" are depends on the project type:

In COCOMO the projects are categorized into three types:-

1. Organic
2. Semidetached
3. Embedded.

Organic:

↳ A development of project can be treated as Organic. If the project deals with developing a well understood application program, the size of the

development team is reasonably small, and the team members are experienced in developing similar methods of Projects.

Eg:→ Simple inventory management System:

Project

Semidetached:

→ A project treated with semidetached type if the development consist of a mixer of experienced and inexperienced staff.

→ Team members may have finite experience in related system but may be unfamiliar with some aspect of the order being developed.

Eg:→ Developing a new operating System.

Project

Embeddod:-

→ A Projed treated as embedded type, If the software being developed is strongly coupled to complex hardware, or if stringent regulations on the operational method exist.

For eg:-> ATM ----//

Progect.

The values of Constant's 'a' and 'b'

Software Projects	a	b
Organic	2.4	1.05
Semi Detached	3.0	1.12
Embedded	3.6	1.20

Example for Basic COCOMO Model.

Calculate COCOMO effort, TDEV, average staffing and Productivity for an organic project that is estimated to be 39,800 lines of code.

solution :-

(KLOC) or (KDLOC) ⇒ $\frac{39800}{1000}$ ⇒ 39.8 Kloc

Project type ⇒ organic

Effort :-

$E_i \Rightarrow a(KLoC)^b$

$\Rightarrow 2.4*(39.8)^{1.05}$

E ⇒ 114.8 Programmer month

Time for development ⇒ $c(E)^d$

The values of constant's c & d are based on project type.

Project type	c	d
Organic	2.5	0.38
Semi Detached	2.5	0.35
Embedded	2.5	0.32

$TDEV = C(E)^d$

$\Rightarrow 2.5*(114.8)^{0.38}$

TDEV ⇒ 15.16 month

Average staffing SS ⇒ E/TDEV

SS ⇒ 114.8/15.16

SS = 7.57 programers.

Productivity ⇒ $\frac{\text{Original Code Size}}{\text{Effort} * 20}$ day/permonth

⇒ $\frac{39800}{114.8 * 20}$ ⇒ 17.3 LOC/Programer day

Intermediate Model:-

↦ The basic cocomo model considers that the effort is only a function of the no. of lines of code and some constants calculated according to the various software systems.

↦ The intermediate cocomo model recognizes these fact and refines the intial estimates obtained through the basic cocomo model by using set of 15 cost drives on Valiay attributes of software engineering.

Classification of cost Derivers and their attributes:-

① Product attributes:

* Required software reliability extent.

- Size of the application database
- The complexity of the product.

iii) Hardware Attributes:-

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

iii) Personal 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					
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 Product	0.70	0.85	1.00	1.15	1.30
Hardware Attributes					
Runtime performance Constraints			1.00	1.11	1.30
Memory Constraints			1.00	1.06	1.21
Volatility of the virtual machine environment		0.87	1.00	1.15	1.30
Required turnabout time		0.94	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	very low	low	Nominal	High	very High
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 values of 'a' and 'b' in case of intermediate

Software Projects	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~~ Example :-

A new Project with estimated 400 KLOC embedded system has to be developed, Project manager has a choice of hiring from two pools of developers: with very high application experience and very little experience in the programming language being used for developers of very low application experience but a lot of experience with the programming language. What is the impact of hiring

all developers from one or the other pool.

Solution:

This is the case of embedded mode.

Hence:

$$E = a_i (KLOC)^{b_i} * EAF$$

$$D = c_i (E)^{d_i}$$

Case 1: Developers are with very high Application experience and very little experience in the programming language being used.

$$EAF = 0.82 * 1.14 = 0.9348$$

$$E = 2.8(200)^{1.20} * 0.9348.$$

$$E = 3470 \text{ Programmer\_month}$$

Duration

$$D = 2.5(3470)^{0.32} = 33.9 \text{ Month}$$

Case 2: Developers of very low applicatio experience but a lot of experience with the programming language.

$$EAF = 1.29 * 0.95 = 1.22$$

$$E = 2.8 (400)^{1.20} * 1.22$$

$$\boxed{E = 4528 \text{ Programmer-month}}$$

Duration (or) Time

$$D = 2.5 (4528)^{0.32}$$

$$\boxed{D = 36.9 \text{ Month}}$$

Case 2: requires more effort and time. Hence, low quality application experience but a lot of programming language experience could not match with the very high application experience and very little programming language experience.