COCOMO : Construcive Cost Model

* Developed by Barry ~~Bob~~ Boehm
* It is hierarchy of s/w cost
* It is cost estimation model

According to Boehm, software cost estimation should be done through three stages.

(1) Basic cocomo (2) Intermediate cocomo
(3) Complete cocomo

1. Basic cocomo Model

* It estimate s/w in rough & quick manner
* Mostly used for small-size software
* It has small size team
* Experience developer needed
* Deadlines are not strict

1. Effor → $a(KLOC)^b$
2. Development time → $c(Effort)^d$ months
3. Average Staff → $\frac{effort}{Development\ Time}$ person
4. Productivity → $\frac{KLOC}{Effort}$ KLOC/P·M

Question: Suppose that a project was estimated to be 400 KLOC. Calculate effort & time for each of three modes of development

1. Organic

$$\text{Effort} = a(\text{KLOC})^b \text{ person-month}$$
$$= 2.4(400)^{1.05}$$
$$= 1295 \text{ PM}$$

$$\text{Development Time} = c(\text{Effort})^d \text{ month}$$
$$= 2.5(1295)^{0.38}$$
$$= 38 \text{ Month.}$$

2- Semidetached

$$\text{Effo} = 3 \times (400)^{1.12} = 2462 \text{ PM}$$
$$\text{Development time} = 2.5 \times (2462)^{0.35}$$
$$= 38.4 \text{ months}$$

3- Embedded

$$\text{Effort} = 3.6 \times (400)^{1.2} = 4772 \text{ PM}$$
$$\text{Development Time} = 2.5 \times (4772)^{0.32}$$
$$= 38 \text{ Months}$$

	Organic	Semi Deatched	Embedded
· Size	2-50 KLOC	50-300 KLOC	300 & KLOC
· Team Size	Small size	Medium size	Large size
· Developer Experience	Experience developer	Avrage Experience ~~peo~~ developer	Very little previous experien-
· Environment	familar Environment	Less familar	Significant environment changes. (Almost new environment)
· Innovation	Little	Medium	Majou
· Deadline	Not tight	Medium	Tight
	Parallel System	Utility System Compiler	Air traffic Monitoring

mode	a	b	c	d
Organic	2.4	1.05	2.5	0.38
Semi-deatched	3.0	1.12	2.5	0.35
Embedded	3.6	1.20	2.5	0.32

Cocomo has three modes of development

Organic, Semi Deatched, Embedded

Intermediate Model

Q. A new project with estimated 400 KLOC embedded system has to developed. Project manager has a choice of hiring from 2 pools of developers: Very high capable (with application) with very little experience in programming language OR developers of low quality but a lot of programming language experience. Which is better experience in terms of 2 pools.

$$E = a\,(KLOC)^b \times EAF$$

$$E = 2.8 \times (400)^{1.20} \times EAF$$

Case I

$EAF = 0.82 \times 1.14 = 0.934$

$E = 2.8\,(400)^{1.20} \times 0.934$

$E = 3470\ PM$

Development Time

$D = 2.5\,(3470)^{0.32}$

$= 33.9\ Months$

Case II

$EAF = 1.29 \times 0.95 = 1.22$

$E = 2.8\,(400)^{1.20} \times 1.22$

$= 3412 \times 1.22$

$E = 4528\ PM$

$D.T = 2.5\,(4528)^{0.32}$

$D.T = 36.9\ M$

Cocomo The Intermediate Model

Difference between Basic & Intermediate model

1. Basic Model was quick but inaccurate & phase ~~inconsistent~~ inconsistent
2. Intermediate model includes a set of 15 additional predictors (cost Drivers)
3. It also take development environment into account during cost estimation.

Use of Cost Driver

Cost driver adjust nominal cost of project to actual project environment
↳ increasing accuracy of estimation

15 Cost Driver

I Product Attributes

a) Required S/w Reliability (REL)
b) Database Size (DATA)
c) Product Complexity (CPLX)

II Computer Attributes

a) Execution time Constraints (TIME)
b) Main Storage Constraint (STOR)
c) Virtual Machine Volatility (VIRT)
d) Computer Turnaround time (TURN)

III Personal Attributes

a) Analyst capability
b) Application Experience
c) Programmer capability
d) Virtual Machine Experience
e) Programming language Experience

IV Project Attributes

(a) Modern Programming Practices
(b) Use of S/w tools
(c) Required development schedule

Each Cost Driver is rated for the given project Environment

Very low — Low — Nominal — High — Very high — Extra high

Effort Adjustment Factor (EAF)

Calculated by multiplying all the values that have been obtained after categorizing each cost Driver

Equation for COCOMO

$$\text{Effort} = a_i (\text{KLOC})^{b_i} \times EAF$$

$$\text{Development Time} = c_i (\text{Effort})^{d_i}$$

Table 5.1: Multipliers of different cost drivers

Cost Drivers	Ratings					
	Very low	Low	Nominal	High	Very	Extra
Product Attributes						
RELY	0.75	0.88	1.00	1.15	1.40	–
DATA	–	0.94	1.00	1.08	1.16	–
CPLX	0.70	0.85	1.00	1.15	1.30	1.65
Computer Attributes						
TIME	–	–	1.00	1.11	1.30	1.66
STOR	–	–	1.00	1.06	1.21	1.56
VIRT	–	0.87	1.00	1.15	1.30	–
TURN	–	0.87	1.00	1.07	1.15	
Personnel Attributes						
ACAP	1.46	1.19	1.00	0.86	0.71	–
AEXP	1.29	1.13	1.00	0.91	0.82	–
PCAP	1.42	1.17	1.00	0.86	0.70	–
VEXP	1.21	1.10	1.00	0.90	–	–
LEXP	1.14	1.07	1.00	0.95	–	–
Project Attributes						
MODP	1.24	1.10	1.00	0.91	0.82	–
TOOL	1.24	1.10	1.00	0.91	0.83	–
SCED	1.23	1.08	1.00	1.04	1.10	–

The multiplying factors for all 15 cost drivers are multiplied to get the effort adjustment factor (EAF). Typically values for EAF ranges from 0.9 to 1.4.

The intermediate COCOMO equations take the form:

$$E = a_i(\text{KLOC})^{bi} \times \text{EAF}$$

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

Table 5.1: Multipliers of different cost drivers

Cost Drivers	Ratings					
	Very low	Low	Nominal	High	Very	Extra
Product Attributes						
RELY	0.75	0.88	1.00	1.15	1.40	–
DATA	–	0.94	1.00	1.08	1.16	–
CPLX	0.70	0.85	1.00	1.15	1.30	1.65
Computer Attributes						
TIME	–	–	1.00	1.11	1.30	1.66
STOR	–	–	1.00	1.06	1.21	1.56
VIRT	–	0.87	1.00	1.15	1.30	–
TURN	–	0.87	1.00	1.07	1.15	
Personnel Attributes						
ACAP	1.46	1.19	1.00	0.86	0.71	–
AEXP	1.29	1.13	1.00	0.91	0.82	–
PCAP	1.42	1.17	1.00	0.86	0.70	–
VEXP	1.21	1.10	1.00	0.90	–	–
LEXP	1.14	1.07	1.00	0.95	–	–
Project Attributes						
MODP	1.24	1.10	1.00	0.91	0.82	–
TOOL	1.24	1.10	1.00	0.91	0.83	–
SCED	1.23	1.08	1.00	1.04	1.10	–

The multiplying factors for all 15 cost drivers are multiplied to get the effort adjustment factor (EAF). Typically values for EAF ranges from 0.9 to 1.4.

The intermediate COCOMO equations take the form:

$$E = a_i(\text{KLOC})^{bi} \times \text{EAF}$$

Mode	a_i	b_i	c_i	d_i
Organic	3.2	1.05	2.5	0.38
Semi Detached	3.0	1.12	2.5	0.38
Embedded	2.8	1.20	2.5	0.32

COCOMO : Detailed Development Model

* It calculates the effect Cost Drivers on each phase of SDLC
* It uses phase sensitive effort multipliers for each cost driver to determine the amount of effort required to complete each phase of SDLC.

* It establishes Module Subsystem Hierarchy

The rating of cat driver is done at the level only where cost driver is most susceptible to variable.

* Adjustment factor (A)

$$A = 0.4(DD) + 0.3C + 0.3I$$

DD is Design Documentation
C is Code
I Integration Testing / Testing

* Size Equivalent $\boxed{\frac{(S \times A)}{100}}$