

Function Point Estimation / COCOMO



SWEN 256 – Software Process & Project Management

Function Point Process

- ✎ 1. Count # of business functions per category
 - Categories: outputs, inputs, db inquiries, files or data structures, and interfaces
- ✎ 2. Establish Complexity Factor for each and apply
 - Simple, Average, Complex
 - Set a weighting multiplier for each (0->15)
 - This results in the “unadjusted function-point total”
- ✎ 3. Compute an “influence multiplier” and apply
 - It ranges from 0.65 to 1.35; is based on 14 factors
- ✎ 4. Results in “function point total”
 - This can be used in comparative estimates

Function Point Estimation

- ☞ Covered last class
- ☞ Each function is divided into one of five categories, then multiplied by the appropriate number below

Function type	Low	Average	High
External inputs	× 3	× 4	× 6
External outputs	× 4	× 5	× 7
External queries	× 3	× 4	× 6
Internal logical files	× 7	× 10	× 15
External interface files	× 5	× 7	× 10

What is COCOMO

- ✎ COnstructive COst MOdel
- ✎ An algorithmic software cost estimation model
- ✎ Developed by Barry W. Boehm
- ✎ Uses a basic regression formula with parameters that are derived from historical project data and current project characteristics

Basic COCOMO



Basic COCOMO — Effort Applied

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

- Where
 - E is the Effort in staff months
 - a and b are coefficients to be determined
 - KLOC is thousands of lines of code

Basic COCOMO — More Constants

Mode	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

Basic COCOMO — The Modes

∞ Organic

- 2-50 KLOC, small, stable, little innovation

∞ Semi-detached

- 50-300 KLOC, medium-sized, average abilities, medium time-constraints

∞ Embedded

- > 300 KLOC, large project team, complex, innovative, severe constraints

Basic COCOMO — Effort Example

Suppose size is 200 KLOC

- Organic
 - $2.4(200)1.05 = 626$ staff-months
- Semi-Detached
 - $3.0(200)1.12 = 1,133$ staff-months
- Embedded
 - $3.6(200)1.20 = 2,077$ staff-months

Basic COCOMO — Development Time

$$TDEV = c(E)^d$$

- Where
 - TDEV is time for development
 - c and d are constants to be determined
 - E is the effort

Basic COCOMO — More Constants

Mode	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

Basic COCOMO — Example

Suppose size is 200 KLOC

- Organic
 - $E = 626$ staff months
 - $TDEV = 2.5(626)^{0.38} = 29$ months
- Semi-Detached
 - $E = 1,133$
 - $TDEV = 2.5(1133)^{0.35} = 29$ months
- Embedded
 - $E = 2077$
 - $TDEV = 2.5(2077)^{0.32} = 29$ months

Basic COCOMO — Average Staff Size

$$SS = \frac{E}{TDEV}$$

$$SS = \frac{\textit{Effort Applied}}{\textit{Development Time}}$$

$$SS = \frac{\textit{staff months}}{\textit{months}}$$

$$SS = \textit{staff}$$

Basic COCOMO — Productivity

$$P = \frac{Size}{E}$$

$$P = \frac{[KLOC]}{[staff\ months]}$$

$$P = \frac{KLOC}{staff\ month}$$

COCOMO Complete Example - Organic

Suppose an organic project has 7.5 KLOC,

- Effort $2.4(7.5)^{1.05} = 20$ staff-months
- Development time $2.5(20)^{0.38} = 8$ months
- Average staff $20 / 8 = 2.5$ staff
- Productivity $7,500 \text{ LOC} / 20 \text{ staff-months} = 375 \text{ LOC} / \text{staff-month}$

Item	Organic
Effort (staff-months)	20
Development Time	8
Average Staff	2.5
Productivity	375

COCOMO Complete Example - Embedded

Suppose an embedded project has 50 KLOC,

- Effort $3.6(50)1.20 = 394$ staff-months
- Development time $2.5(394)0.32 = 17$ months
- Average staff $394 / 17 = 23$ staff
- Productivity $50,000 \text{ LOC} / 394 \text{ staff-months} = 127 \text{ LOC} / \text{staff-month}$

Item	Embedded
Effort (staff-months)	394
Development Time	17
Average Staff	23
Productivity	127

Intermediate COCOMO



Intermediate COCOMO — Cost Drivers

Cost Drivers	Ratings					
	Very Low	Low	Nominal	High	Very High	Extra 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 the product	0.70	0.85	1.00	1.15	1.30	1.65
Hardware attributes						
Run-time performance constraints			1.00	1.11	1.30	1.66
Memory constraints			1.00	1.06	1.21	1.56
Volatility of the virtual machine environment		0.87	1.00	1.15	1.30	
Required turnabout time		0.87	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						
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	

Intermediate COCOMO — Effort



$$E = a(KLOC)^b \underbrace{\times C}_{\text{New}}$$

Where

- E is the effort
- a and b are constants (as before)
- KLOC is thousands of lines of code
- C is the effort adjustment factor

Intermediate COCOMO — Cost Drivers

Intermediate COCOMO introduces Cost Drivers

- They are used because
 - they are statistically significant to the cost of the project; and
 - they are *not* correlated to the project size (KLOC)
- To obtain multiplier
 - Determine each number using the grid (next slide)
 - Multiply them – the product is C

COCOMO Advantages

- Based on history
- Repeatable
- Unique adjustment factors
- Has different modes
- Works well on similar projects
- Highly calibrated
- Well-documented
- Easy to use

COCOMO Limitations

- Ignores requirements volatility
- Ignores documentation
- Ignores customer's "skill"
- Oversimplifies security
- Ignores software safety
- Ignores personnel turnover
- Ignores many hardware issues
- Personnel experience may be obsolete
- Must know the cost drivers
- **Must be able to predict project size**

“The models are just there to help, not to make the management decisions for you.”

--Barry Boehm

Questions/Discussion

