Software Project Management

Durga Prasad Mohapatra
Professor
CSE Deptt.
NIT Rourkela

Project Estimation Techniques cont …

- **Intermediate COCOMO**
- **Complete COCOMO**

Intermediate COCOMO

- Basic COCOMO model assumes
  - effort and development time depend on product size alone.
- However, several parameters affect effort and development time, such as:
  - Reliability requirements
  - Availability of CASE tools and modern facilities to the developers
  - Size of data to be handled

Intermediate COCOMO

- For accurate estimation,
  - the effect of all relevant parameters must be considered:
  - Intermediate COCOMO model recognizes this fact:
    - refines the initial estimate obtained by the basic COCOMO by using a set of 15 cost drivers (multipliers).

Intermediate COCOMO cont …

For example,

- If modern programming practices are used,
  - initial estimates are scaled downwards.

- If there are stringent reliability requirements on the product,
  - initial estimate is scaled upwards.

Intermediate COCOMO cont…

- Rate the different parameters, may be on a scale of one to three or three-point scale:
  - Depending on these ratings,
    - multiply cost driver values with the estimate obtained using the basic COCOMO.
- In some cases, value of the parameters (cost drivers) may be less than 1, e.g. if modern programming practices are being used.

Intermediate COCOMO cont …

- Takes basic COCOMO as starting point
- Identifies the attributes such as personnel, product, computer and project attributes, which affect the cost and development time.
- Multiplies basic cost by attribute multipliers which may increase or decrease the costs.

Attributes

Personnel attributes

- Analyst capability (ACAP)
- Virtual machine experience (VEXP)
- Programmer capability (PCAP)
- Programming language experience (LEXP)
- Application experience (AEXP)

Product attributes

- Reliability requirement (RELY)
- Database size (DATA)
- Product complexity (CPLX)

More Attributes

Computer attributes

- Execution time constraints (TIME)
- Storage constraints (STOR)
- Virtual machine volatility (VIRT)
- Computer turnaround time (TURN)

Project attributes

- Modern programming practices (MODP)
- Software tools (TOOL)
- Required development schedule (SCED)

COCOMO effort multipliers (cost drivers)

- Each of the 15 attributes receives a rating on a six-point scale that ranges from "very low" to "extra high" (in importance or value).
- The product of all effort multipliers results in an *effort adjustment factor (EAF)*.
- Typical values for EAF range from 0.9 to 1.4.

COCOMO – cost drivers

Cost Driver	Very low	Low	Nominal	High	Very High	Extra High
Required reliability	0.75	0.88	1.0	1.15	1.40	
Database size		0.94	1.0	1.08	1.16	
Product complexity	0.70	0.85	1.0	1.15	1.30	
Execution time constraint			1.0	1.11	1.30	
Memory constraint			1.0	1.06	1.21	
Virtual machine volatility		0.87	1.0	1.15	1.30	
Computer turnaround time		0.87	1.0	1.07	1.15	
Analyst capability	1.46	1.19	1.0	0.86	0.71	
Application experience	1.29	1.13	1.0	0.91	0.82	
Programmer capability	1.42	1.17	1.0	0.86	0.70	
Virtual machine experience	1.21	1.10	1.0	0.90		
Programming language experience	1.14	1.07	1.0	0.95		
Modern programming practices	1.24	1.10	1.0	0.91	0.82	
Use of software tools	1.24	1.10	1.0	0.91	0.83	
Development schedule	1.23	1.08	1.0	1.04	1.10	

Intermediate Model Effort Equation (COCOMO 81)

- Effort=EAF*c*(size)^k
  - EAF (effort adjustment factor) is the product of effort multipliers corresponding to each cost driver rating
  - c is a constant based on the developmental mode
    - organic = 3.2
    - semi = 3.0
    - embedded = 2.8
  - size = 1000s Delivered Source Instruction (KDSI)
  - k is a constant for the given mode

Intermediate Model
Effort Equation (COCOMO 81) cont …

- The development time calculation uses **effort** in the same way as in the Basic COCOMO, i.e.

  Nominal Development time= $2.5*(\text{Effort})^{\text{exponent}}$

  where, 2.5 is constant for all modes, and

  exponent is based on mode

  organic = 0.38

  semidetached = 0.35

  embedded = 0.32

Using COCOMO development effort multipliers

An example: for analyst capability:

- Assess capability as very low, low, nominal, *high* or very high
- Extract multiplier:

very low	1.46
low	1.19
nominal	1.00
high	*0.80*
very high	0.71

- Adjust nominal estimate e.g. 32.6 x 0.80 = 26.8 staff months

Example 1

- Determine effort, duration, staffing level for the following scenario:
  - Estimated size 10,000 LOC = 10 KLOC.
  - Small project, familiar development
  - Analyst capability: Low
  - Application experience: Low
  - programmer capability: low
  - Programmer experience: High

Example 1 cont …

- Need to produce 10,000 LOC = 10 KLOC.
- Since a small project and familiar development, use organic model:
  - Effort $=3.2(10)^{1.05} = 35.86$ Person-Months
  - Development-time $= 2.5(35.86)^{.38} = 9.74$ Months
  - Average Staff = 35.86 PM/9.74 Months = $3.68 \cong 4$ People

Example 1 cont…

- Now, the attribute multipliers will be as follows:
  - Analyst capability - 1.19 (low)
  - Application experience - 1.13 (low)
  - programmer capability - 1.17 (low)
  - programming experience - 0.95 (high)
- So, Adjustment factor = 1.19*1.13*1.17*0.95 = 1.49
- Effort = 35.86*1.49 = 53 Person-months
- Development time = $2.5*(53)^{.38}$ = 11.3 Months
- Average Staff = 53PM/11.3M=4.69 (approx. 5) People

Example 2

- Suppose the project to be developed is a flight control system (mission critical) with 319,000 DSI in embedded mode.
- Reliability must be very high (RELY=1.40). So we can calculate:

  Effort = $1.40*3.6*(319)^{1.20}$ = 5093 PM(approx.)
  Duration = $2.5*(5093)^{0.32}$ = 38.4 months (approx.)
  Average Staffing = 5093 PM/38.4 months = 133 People (approx.)
  This calculation requires correction by replacing 3.6 with 2.8.

Example 3

- An embedded software system on microcomputer hardware to be developed.
- Basic COCOMO predicts a 45 person-month effort requirement
- Attributes = RELY (1.15), STOR (1.21), TIME (1.10), TOOL (1.10)
- Intermediate COCOMO predicts

  45 * 1.15 * 1.21 * 1.10 *1.10 = 76 person-months.
- Assume total cost of person month = Rs. 50,000.
- Total cost = 76 * 50,000 = Rs.38,00, 000

Shortcoming of basic and intermediate COCOMO models

- For better accuracy:
  - COCOMO has to be calibrated to an organizations' environment.
- Very sensitive to parameter change:
  - Over a person-year difference in a 10 KLOC project with minor adjustments
- Broad brush model that can generate significant errors

Shortcoming of basic and intermediate COCOMO models cont …

- Software reuse
- Application generation programs
- Object-oriented approaches
- Application engineering (reuse, applications translation)
- Need for rapid development

Shortcoming of basic and intermediate COCOMO models

- Both models:
  - consider a software product as a single homogeneous entity:
  - However, most large systems are made up of several smaller sub-systems.
    - Some sub-systems may be considered as organic type, some may be considered embedded, etc.
    - for some the reliability requirements may be high, and so on.

Complete COCOMO

- Overcomes some of the limitations of Basic and Intermediate COCOMO.
- Cost of each sub-system is estimated separately.
- Costs of the sub-systems are added to obtain total cost.
- Reduces the margin of error in the final estimate.

Complete COCOMO Example

- A Management Information System (MIS) for an organization having offices at several places across the country:
  - Database part (semi-detached)
  - Graphical User Interface (GUI) part (organic)
  - Communication part (embedded)
- Costs of the individual components are estimated separately:
  - summed up to give the overall cost of the system.

Summary

- Discussed the fundamentals of Intermediate COCOMO.
- Presented the different cost drivers (multipliers).
- Explained Cost and Effort estimation using Intermediate COCOMO.
- Shown the limitations of Basic and Intermediate COCOMO.
- Solved some examples on Cost and Effort estimation using Intermediate COCOMO.
- Discussed fundamentals of Complete (Detailed) COCOMO.

References :

1. B. Hughes, M. Cotterell, R. Mall, *Software Project Management*, Sixth Edition, McGraw Hill Education (India) Pvt. Ltd., 2018.
2. R. Mall, *Fundamentals of Software Engineering*, Fifth Edition, PHI Learning Pvt. Ltd., 2018.

Thank you