Software Project Management

Durga Prasad Mohapatra
Professor
CSE Deptt.
NIT Rourkela

Project Estimation Techniques cont…

- **Parametric models**
  - **COCOMO 81 and COCOMO II**

Parametric models for Size

1. Albrecht/IFPUG function points
2. Symons/Mark II function points
3. COSMIC function points
4. COCOMO 81 and COCOMO II

COCOMO

- COCOMO (CONSTRUCTIVE COST MODEL) -First published by Dr. Barry Boehm, 1981
- Several interactive cost estimation software
- packages available
- Derived from statistical regression of data from a base of 63 past projects (2000 - 512,000 DSIs)

COCOMO 81

- Based on industry productivity standards - database can be constantly updated
- Allows an organization to benchmark its software development productivity
- Basic model:

  $$\textbf{effort} = c \times \textbf{size}^{k}$$

  - c and k depend on the type of system: organic, semi-detached, embedded
  - Size is measured in 'kloc' i.e. Thousands of lines of code

COCOMO Mode & Model

- Three development environments (modes)
  - Organic Mode
  - Semidetached Mode
  - Embedded Mode
- Three increasingly complex models
  - Basic Model
  - Intermediate Model
  - Detailed Model

COCOMO Modes

- Organic Mode
  - Developed in familiar, stable environment
  - Product similar to previously developed product
  - <50,000 DSIs (ex: accounting system)
- Semidetached Mode
  - somewhere between Organic and Embedded (e.g. compilers, linkers etc.)
- Embedded Mode
  - new product requiring a great deal of innovation
  - inflexible constraints and interface requirements (ex: operating systems, real-time systems)

Modes

Feature	Organic	Semidetached	Embedded
Organizational understanding of product and objectives	Thorough	Considerable	General
Experience in working with related software systems	Extensive	Considerable	Moderate
Need for software conformance with pre-established requirements	Basic	Considerable	Full
Need for software conformance with external interface specifications	Basic	Considerable	Full

COCOMO Models

- Basic Model
  - Used for early rough, estimates of project cost, performance, and schedule
  - Accuracy: within a factor of 2 of actuals 60% of time
- Intermediate Model
  - Uses Effort Adjustment Factor (EAF) from 15 cost drivers
  - Doesn't account for 10 - 20 % of cost (training, maintenance, Quality, etc.)
  - Accuracy: within 20% of actuals 68% of time
- Detailed Model
  - Uses different Effort Multipliers for each phase of project (Most project managers use intermediate model)

Basic Effort Equation (COCOMO 81)

- Effort=c * (size)^k
  - c is a constant based on the developmental mode
    - organic = 2.4
    - semi = 3.0
    - embedded = 3.6
  - size = 1000s Source Lines of Code (KSLOC)
  - k is a constant for a given mode
    - organic = 1.05
    - semi = 1.12
    - embedded = 1.20

The COCOMO constants

System type	c	k
Organic (broadly, information systems)	**2.4**	**1.05**
Semi-detached (broadly utility apps)	**3.0**	**1.12**
Embedded (broadly, real-time)	**3.6**	**1.20**

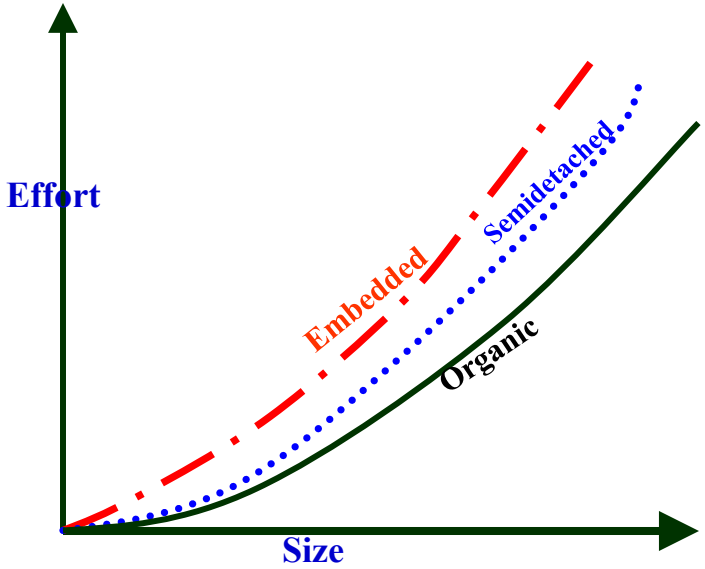
k exponentiation – 'to the power of…'
adds disproportionately more effort to the larger projects
takes account of bigger management overheads

Basic Model: Schedule Equation (COCOMO 81)

- Nominal Development time= $2.5*(\text{Effort})^{\text{exponent}}$
- 2.5 is constant for all modes
- Exponent based on mode
  - organic = 0.38
  - semi = 0.35
  - embedded = 0.32

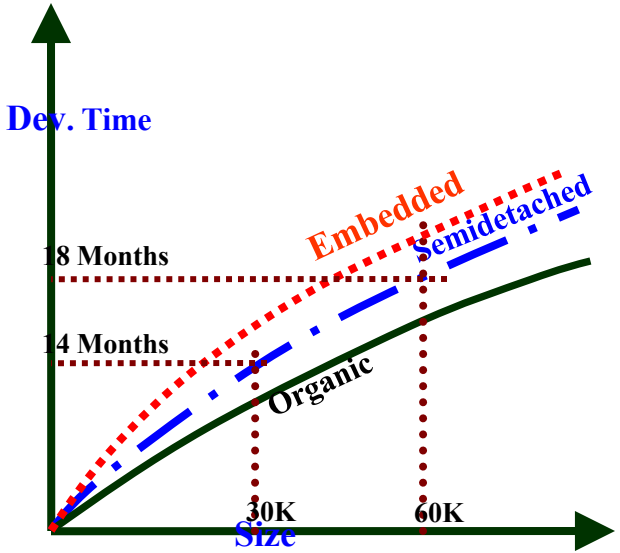
Basic COCOMO Model (CONT.)

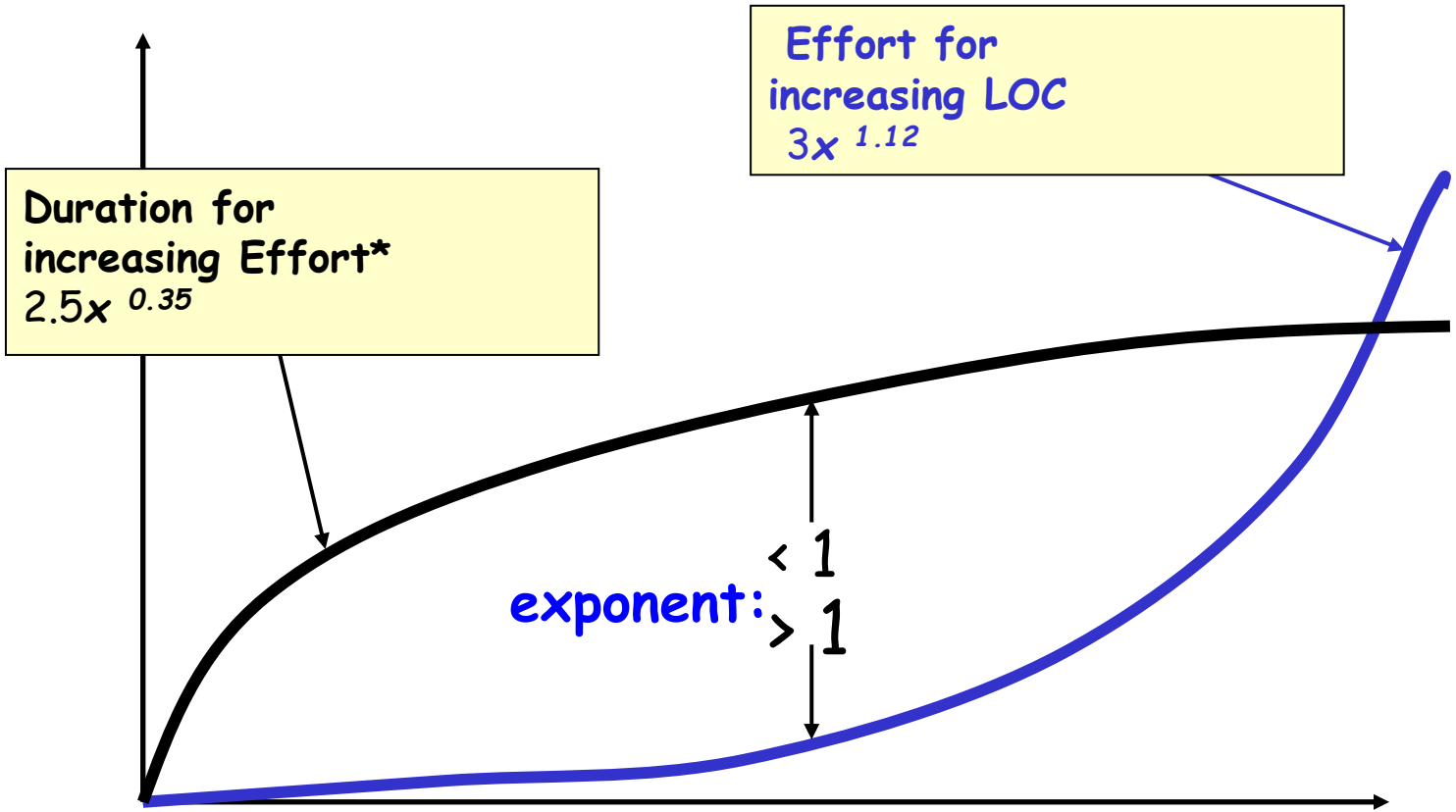
- Effort is somewhat super-linear in problem size.



Basic COCOMO Model cont …

- Development time
  - sublinear function of product size.
- When product size increases two times,
  - development time does not double.
- Time taken:
  - almost same for all the three product categories.




Effort for increasing LOC
3x 1.12
Duration for increasing Effort*
2.5x 0.35
exponent:
< 1
> 1

Basic COCOMO Model cont …

- Development time does not increase linearly with product size:
  - For larger products more parallel activities can be identified:
    - can be carried out simultaneously by a number of engineers.

Basic COCOMO Model cont …

- Development time is roughly the same for all the three categories of products:
  - For example, a 60 KLOC program can be developed in approximately 18 months
    - regardless of whether it is of organic, semi-detached, or embedded type.
  - There is more scope for parallel activities for system and application programs,
    - than utility programs.

Example - 1

The size of an organic software product has been estimated to be 32,000 lines of source code. Assume that average salary of a software developer is Rs. 15,000 per month. Determine the effort required to develop the software product, the nominal development time, and the cost to develop the product.

- Effort = $2.4*(32)^{1.05}$ = 91 PM
- Nominal development time = $2.5*(91)^{0.38}$ = 14 months
- Staff cost required to develop the product

  =91 x Rs. 15,000=Rs. 1,465,000

Example - 2

Suppose you are developing a software product in the organic mode. You have estimated the size of the product to be about 100,000 lines of code. Compute the nominal effort and the development time.

- Given that the size is 100 KLOC and the project is organic.
- Nominal effort=$2.4 \times 100^{1.05}=2.4 \times 125.893=302.1$ man-months
- Nominal development time=$2.5 \times (Effort)^{0.38}=2.5 \times 302.1^{0.38}=8.6$ months

Example - 3

Suppose that a certain software product for business application costs Rs. 50,000 to buy off-the-shelf and that its size is 40 KLOC. Assuming that in-house developers cost Rs. 6000 per PM (including overheads), would it be more cost-effective to buy the product or build it?

The product is for business application and can be classified as organic type. So,

- Nominal effort=$2.4 \times 40^{1.05}$=2.4 x 48.1=115.5 man-months
- In-house engineers cost Rs. 6000/-.
- So, the cost of development is 115.5 x 6000=Rs. 693,000/-
- But, purchasing the above S/W will cost Rs. 50,000.
- So, it is better to go for buying the product.

Example - 4

Suppose an organic project has 7.5 KLOC. Find the effort, development time, average staff required and productivity.

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

Example - 5

Suppose an embedded project has 50 KLOC. Find the effort, development time, average staff required and productivity.

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

Exercise

- A software package is required by a company to mine existing customer data to select prospective customers for a new launch.
  - The size is estimated to be 30KLOC.
  - Assume competent developers can be hired at Rs50,000/- per month.
  - However, commercial offering supporting almost all of the required features costs Rs. 100,000/-
- Should the company buy or build the product?

Buy/Build Decision

- The make/buy decision can be made based on the following conditions
  - Will the software product be available sooner than internally developed software?
  - Will the cost of acquisition plus the cost of customization be less than the cost of developing the software internally?
  - Will the cost of outside support (e.g., a maintenance contract) be less than the cost of internal support?

Summary

- Discussed fundamentals of Basic COCOMO
- Discussed various types of projects such as organic, semidetached and embedded
- Presented Cost and Effort estimation using Basic COCOMO
- Solved some examples on Cost and Effort estimation using Basic COCOMO

References :

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

Thank you