PROJECT ESTIMATION TECHNIQUES

- Estimation of various project parameters is an important project planning activity.
- The different parameters of a project that need to be estimated include—project size, effort required to complete the project, project duration, and cost.
  - Empirical estimation techniques
  - Heuristic techniques
  - Analytical estimation techniques

- Empirical Estimation Techniques
- Empirical estimation techniques are essentially based on making an educated guess of the project parameters.
- While using this technique, prior experience with the development of similar products is helpful.
- Heuristic Techniques
- Heuristic techniques assume that the relationships that exist among the different project parameters can be satisfactorily modelled using suitable mathematical expressions.

- Analytical Estimation Techniques
- Analytical estimation techniques derive the required results starting with certain basic assumptions regarding a project.
- Unlike empirical and heuristic techniques, analytical techniques do have certain scientific basis.
- It outperforms both empirical and heuristic techniques as far as estimating software maintenance efforts is concerned.

COCOMO - A HEURISTIC ESTIMATION TECHNIQUE

COCOMO Product Classes

- Roughly correspond to:
  - application, utility and system programs respectively.
    - Data processing and scientific programs are considered to be application programs.
    - Compilers, linkers, editors, etc., are utility programs.
    - Operating systems and real-time system programs, etc. are system programs.

COCOMO Model (CONT.)

- For each of the three product categories:
  - From size estimation (in KLOC), Boehm provides equations to predict:
    - project duration in months
    - effort in programmer-months
- Boehm obtained these equations:
  - examined historical data collected from a large number of actual projects.

Person-month

- Popular unit for effort measurement
- Person-month (PM) is considered to be an appropriate unit for measuring effort
- because developers are typically assigned to a project for a certain number of months
- One person month is the effort an individual can typically put in a month

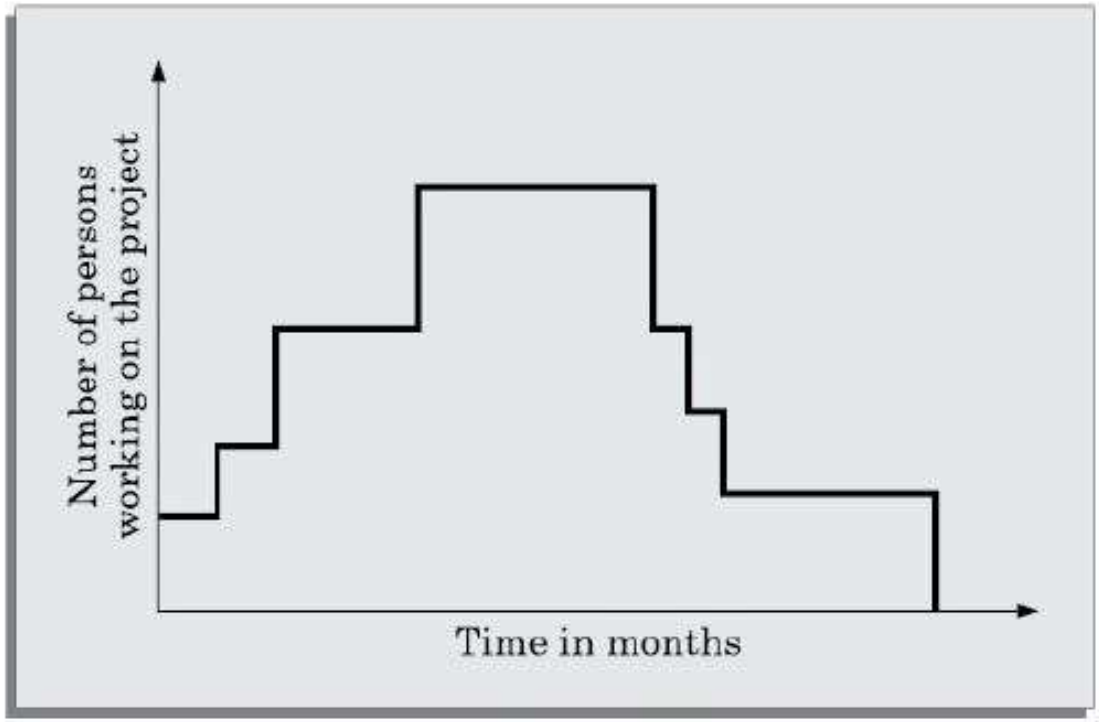


Figure 3.3: Person-month curve.

- The plot in Figure 3.3 shows that different number of personnel may work at different points in the project development
- The number of personnel working on the project usually increases or decreases by an integral number, resulting in the sharp edges in the plot

COCOMO Model (CONT.)

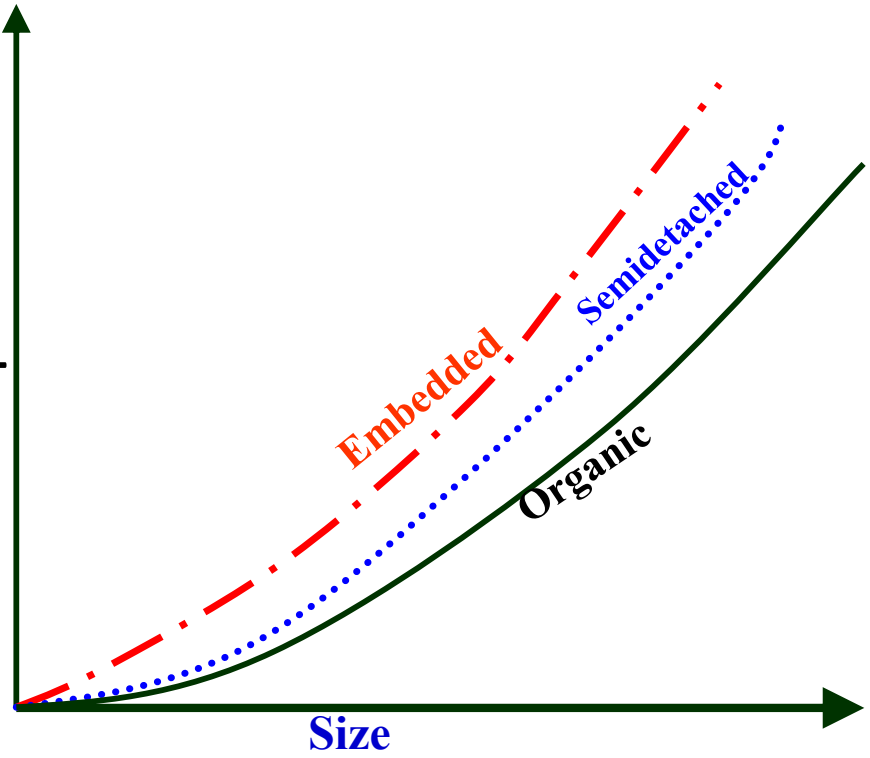
- Software cost estimation is done through three stages:
  - Basic COCOMO,
  - Intermediate COCOMO,
  - Complete COCOMO.

COCOMO Model (CONT.)

- The basic COCOMO estimation model is given by expressions of the following
- forms:
- Effort = a1 × (KLOC)a2 PM
- Tdev = b1 × (Effort)b2 months
- where,
- KLOC is the estimated size of the software product expressed in Kilo
- Lines Of Code.
- a1, a2, b1, b2 are constants for each category of software product.
- Tdev is the estimated time to develop the software, expressed in months.
- Effort is the total effort required to develop the software product, expressed in person- months (PMs).

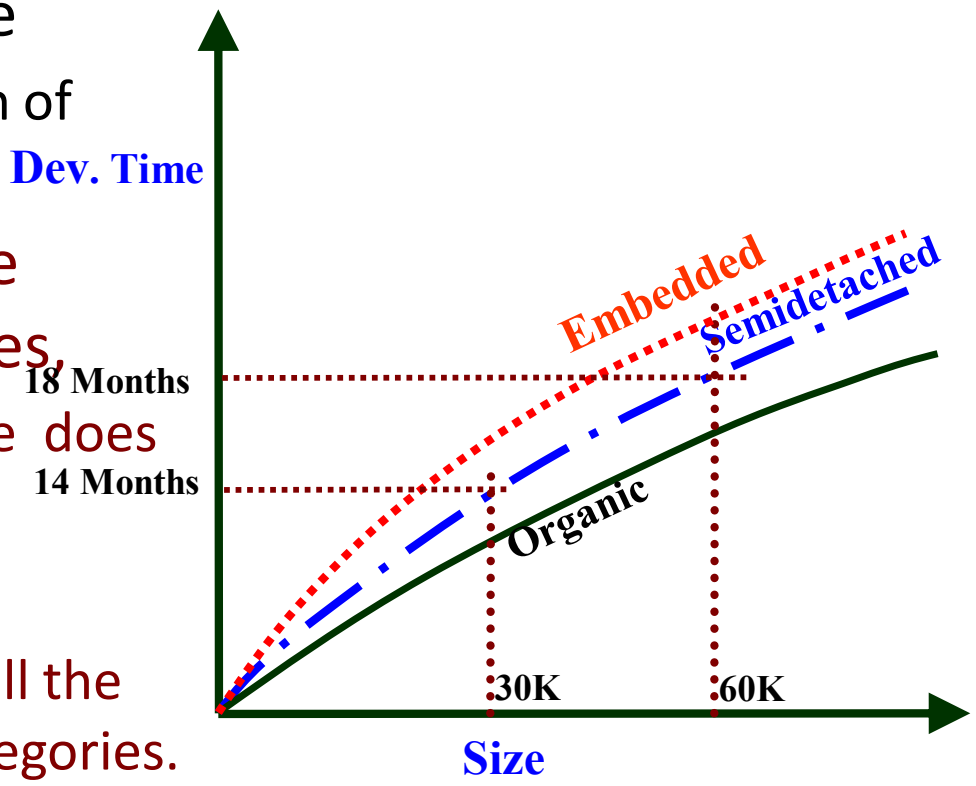
Basic COCOMO Model (CONT.)

- 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.



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.

Basic COCOMO Model (CONT.)

Example

Assume that the size of an organic type software product has been estimated to be 32,000 lines of source code. Assume that the average salary of a software developer is Rs. 25,000 per month. Determine the effort required to develop the software product, the nominal development time, and the cost to develop the product.

Intermediate COCOMO Model

- In order to obtain an accurate estimation of the effort and project duration, the effect of all relevant parameters must be taken into account.
- **Product:** The characteristics of the product that are considered include the inherent complexity of the product, reliability requirements of the product, etc.
- **Computer:** Characteristics of the computer that are considered include the execution speed required, storage space required, etc.
- **Personnel:** The attributes of development personnel that are considered include the experience level of personnel, their programming capability, analysis capability, etc.
- **Development environment:** Development environment attributes capture the development facilities available to the developers. An important parameter that is considered is the sophistication of the automation (CASE) tools used for software 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.

3.Complete 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 components are estimated separately:
  - summed up to give the overall cost of the system.