responsibility of compiling the results and p[illegible]

3.6 COCOMO—A [illegible] TECHNIQUE

COCOMO (COnstructive COst estimation MOdel) was proposed by Boehm, 1981. Boehm postulated that any software development project can be classified into any one of the following three categories based on the development complexity: organic, semidetached, and embedded. In order to classify a product into the identified categories, Boehm requires us to consider not only the characteristics of the product but also those of the development team and development environment. Roughly speaking, the three product classes correspond to application, utility and system programs, respectively. Normally, data processing programs[1] are considered to be application programs. Compilers, linkers, etc. are utility programs. Operating systems and real-time system programs, etc. are system programs. System programs interact directly with the hardware and typically involve meeting timing constraints and concurrent processing.

Brooks, 1975 states that utility programs are roughly three times as difficult to write as application programs, and system programs are roughly three times as difficult as utility programs. Thus, according to Brooks, the relative levels of product development complexity for the three categories (application, utility and system programs) of products are 1:3:9.

Boehm's [1981] definitions of organic, semidetached, and embedded systems are elaborated as follows:

1. **Organic:** We can consider a development project to be of organic type, if the project deals with developing a well-understood application program, the size of the development

[1] A data processing program is one which processes large volumes of data using a simple algorithm. An example of a data processing application is a payroll software. A payroll software computes the salaries of the employees and prints cheques for them. In a payroll software, the algorithm for pay computation is fairly simple. The only complexity that arises while developing such a software product arises from the fact that the pay computation has to be done for a large number of employees.

team is reasonably small, and the team members are experienced in developing similar types of projects.

2. **Semidetached:** A development project can be considered to be of semidetached type, if the development team consists of a mixture of experienced and inexperienced staff. Team members may have limited experience on related systems but may be unfamiliar with some aspects of the system being developed.

3. **Embedded:** A development project is considered to be of embedded type, if the software being developed is strongly coupled to complex hardware, or if stringent regulations on the operational procedures exist.

Observe that Boehm in addition to considering the characteristics of the product being developed, considers the characteristics of the team members in deciding the category of the development project. Thus, a simple data processing program may be classified as semidetached if the team members are inexperienced in the development of similar products. For the three product categories, Boehm provides different sets of expressions to predict the effort (in units of person-months) and development time from the size estimation given in KLOC (Kilo Lines of Source Code). One person-month is the effort an individual can typically put in a month. This effort estimate takes into account the productivity losses that may occur due to lost time such as holidays, weekly offs, coffee breaks, etc.

Note that effort estimation is expressed in units of person-months (PM). 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. The person-month unit indicates the work done by one person working on the project for one month. It should be carefully noted that an effort estimation of 100 PM does not imply that 100 persons should work for 1 month. Neither nor does it imply that 1 person should be employed for 100 months. The effort estimation simply denotes the area under the person-month curve (see Figure 3.3) for the project. The plot in Figure 3.3 shows that different number of personnel may work at

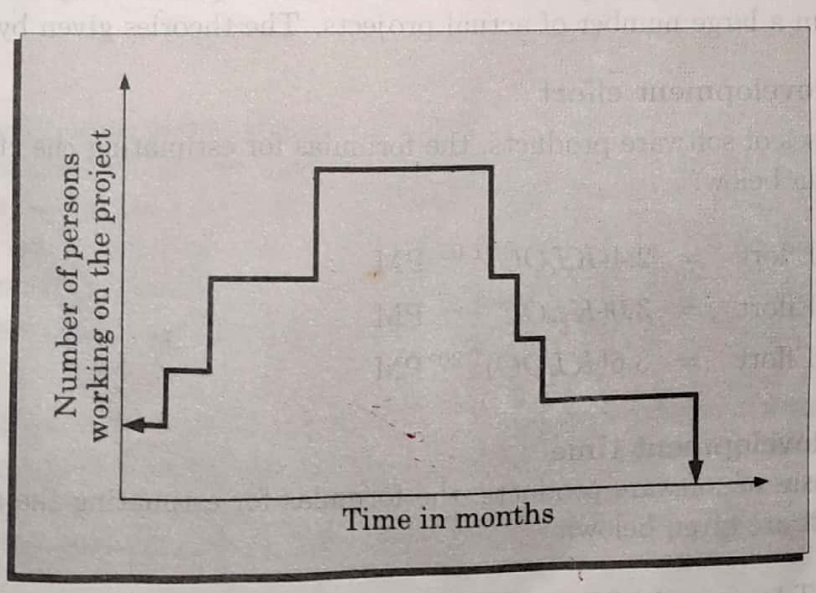

Figure 3.3: Person-month curve.

different point in the project development, as is typical in a practical industry scenario. The number of personnel working on the project usually increases and decreases by an integral

number, resulting in the sharp edges in the plot. We shall elaborate in Section 3.8 how the number of persons to work at any time on the product development is determined.

According to Boehm, software cost estimation should be done through three stages: basic COCOMO, intermediate COCOMO, and complete COCOMO. We discuss these stages as follows:

3.6.1 Basic COCOMO Model

The **basic COCOMO model** gives an approximate estimate of the project parameters. The basic COCOMO estimation model is given by the following expressions:

$$\text{Effort} = a_1 \times (KLOC)^{a_2} \text{ PM}$$

$$\text{Tdev} = b_1 \times (Effort)^{b_2} \text{ Months}$$

Where

(a) KLOC is the estimated size of the software product expressed in Kilo Lines of Code,

(b) a_1, a_2, b_1, b_2 are constants for each category of software products,

(c) Tdev is the estimated time to develop the software, expressed in months,

(d) Effort is the total effort required to develop the software product, expressed in person months (PMs).

According to Boehm, every line of source text should be calculated as one LOC irrespective of the actual number of instructions on that line. Thus, if a single instruction spans several lines (say n lines), it is considered to be nLOC. The values of a_1, a_2, b_1, b_2 for different categories of products as given by Boehm [1981]. He derived the above expressions by examining historical data collected from a large number of actual projects. The theories given by Boehm were:

Estimation of development effort

For the three classes of software products, the formulas for estimating the effort based on the code size are shown below:

Organic : Effort $= 2.4(KLOC)^{1.05}$ PM

Semidetached : Effort $= 3.0(KLOC)^{1.12}$ PM

Embedded : Effort $= 3.6(KLOC)^{1.20}$ PM

Estimation of development time

For the three classes of software products, the formulas for estimating the development time based on the effort are given below:

Organic : Tdev $= 2.5(Effort)^{0.38}$ Months

Semidetached : Tdev $= 2.5(Effort)^{0.35}$ Months

Embedded : Tdev $= 2.5(Effort)^{0.32}$ Months