**Chapter 3 Exercise**

**Student Name:** Navachethan Murugeppa

**Student ID:** 40306253

**Course:** Software Project Management (SOEN-6481)

**3.2 Describe the COCOMO technique for deriving effort and cost estimates for software projects?**

**-** COCOMO (COnstruvtive Cost Model) is a technique used for estimating effort, cost, and schedules for a software project. It is widely used and most reliable as it works even when the previous projects data are unavailable, as it determines the estimation based on industry standard metrics along with lines of code, cost drivers and scale drivers, which makes it easy and reliable solution. Since the first demonstration of COCOMO model there have been several modifications made to adjust the requirements by different software organisations based on their kind of projects, but the core principle stays the same till date. COCOMO can be applied into 3 levels namely basic COCOMO, Intermediate COCOMO and Detailed COCOMO.

*Basic COCOMO:*

This is the simplest and original COCOMO method to calculate the effort and cost which basically determines the effort and cost based on a few empirical constant values and the Kilo lines of code (KLOC). The standard formula for effort is

Effort = 2.94 \* EAF \*(KLOC)E

Where, EAF – is the effort adjustment factor.

KLOC – Kilo Lines of code

E is the exponent derived from the scale drivers.

Duration = 3.67\*(effort)SE

Where SE is the schedule equation derived from the scale drivers.

*Intermediate COCOMO:*

With an improvisation on the basic COCOMO model for cost and effort estimation that solely depended on the KLOC and the cost, scale drivers the intermediate COCOMO model calculates with added parameters such as required reliability, application database size and application complexity which are termed as project attributes. Other attributes such as product attributes, hardware attributes, personnel attributes contribute to the calculation. With these parameters calculation is done based on assigning the weights or impact level of the attribute in the project with a scale ranging from very low to extra high having 6 divisions with values ranging from 0.7 to 1.6. The formula for intermediate COCOMO is :

E = a(KLOC)E

Where E and a are co-efficients whose values are determined based on the type of software project that includes organic, semi-detached, and embedded, each having defined co-efficients.

*Detailed COCOMO:*

Detailed COCOMO model is the calculations done on the bigger project that has clear information and requirements which is implemented in phases. In this the intermediate COCOMO formulae are used to calculate the effort and cost estimation considering each phase of the project which makes this approach a detailed COCOMO. Here the same procedure is followed as done previously in intermediate COCOMO calculations, but there the attributes are used for a whole project, but in detailed COCOMO calculations, cost drivers and attributes are analysed for each phase of the development making it more specific and detailed estimation.

To conclude the above solution talks about the evolution of COCOMO technique to calculate cost and effort estimates for a project. Although the above types are commonly used, there are many more variants of COCOMO models which were developed based on the organisational standards of the organisation that has adopted COCOMO. Some of the examples are COSYSMO and COCOMO II.

Example:

Consider A software development team is working on a **database management system** for a medium-sized business. The estimated **size** of the project is **100 KLOC (Kilo Lines of Code)**. The project falls under the **Semi-Detached** category, as it involves a mix of experienced and less-experienced developers.

Estimate the effort (in person-months) and development time (in months) using:

1. Basic COCOMO Model
2. Intermediate COCOMO Model
3. Detailed COCOMO Model

Solution:

The co-efficient values required for the COCOMO calculation for a semi-detached project are:

a = 3.0

b = 1.12

c = 2.5

d = 0.35

1. Basic COCOMO:

Effort calculation = a \* (KLOC)b

KLOC = 100

= 3.0 \* 1001.12

= **391.2 person months**

Cost estimation, T = c \* Ed

= 2.5 \* 391.20.35

= **46.3 months**

1. Intermediate COCOMO:

Effort estimation E = a\*(KLOC)b \* ∏(Effort Multipliers)

By assuming some of the effort multipliers as:

Reliability factor: 1.10

Complexity factor: 1.15

Experience level: 0.95

Development tools efficiency: 1.05

Storage Constraints: 1.10

Other cost drivers: 1.05

Total multipliers, TM = sum (all the above effort multipliers) = 1.10+1.15+0.95+1.05+1.10+1.05 = 1.43

Hence, total effort = 3.0 \* 1001.12 \* TM = 391.2 \* 1.43 = **559.4 person months**

Total time, T = 2.5 \* 559.40.35

= **53.5 months**

1. Detailed COCOMO:

Now consider that a project is divided into multiple phases and each phase contribute certain percentage of development. Each phase must be identified with required cost and scale drivers and intermediate cocomo analysis must be made for that phase. But for simplicity let’s consider that all the phases have same cost drivers with same value. The project is divided into four phases

Planning and requirements – 10%

Design – 20%

Coding and Unit Testing – 40%

Integration and Testing – 30%

The effort and time is estimated to all the above phases as per the following table:

|  |  |  |
| --- | --- | --- |
| **Phase** | **Effort estimation in**  **person months** | **Time estimation in**  **months** |
| Planning and requirements | 559.4 \* 0.1 = 55.94 | 6 months |
| Design | 559.4\*0.2 = 111.88 | 12 months |
| Coding and Unit Testing | 559.4\*0.4 = 223.76 | 22 months |
| Integration and Testing | 559.4\*0.3 = 167.82 | 13.5 months |
| **Total** | **559.4 PM** | **53.5 months** |

Note:

**Comparison of Constants Across Project Types**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Project Type** | **a (Effort Multiplier)** | **b (Effort Exponent)** | **c (Time Multiplier)** | **d (Time Exponent)** |
| **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 |