Department of Computer Engineering

Academic Term: First Term 2023-24

Class: T.E /Computer Sem – V / Software Engineering

Practical No:	**6**
Title:	**Estimating project cost using COCOMO Model**
Date of Performance:	**04-09-2023**
Roll No:	**9539, 9572**
Team Member:	**Crystal Fernandes, Sanika Rozario**

Sr. No	Performance Indicator	Excellent	Good	Below Average	Total Score
1	On time Completion & Submission (01)	01 (On Time)	NA	00 (Not on Time)	
2	Theory Understanding(02)	02(Correct)	NA	01 (Tried)	
3	Content Quality (03)	03(All used)	02 (Partial)	01 (rarely followed)	
4	Post Lab Questions (04)	04(done well)	3 (Partially Correct)	2(submitted)	

Signature of the Teacher:

Lab Experiment 05

Experiment Name: Estimating Project Cost Using COCOMO Model in Software Engineering

Objective: The objective of this lab experiment is to introduce students to the COCOMO (Constructive Cost Model) estimation technique for estimating software project cost and effort. Students will gain practical experience in using the COCOMO model to estimate the development effort, duration, and resources required for a sample software project.

Introduction: COCOMO is a widely used algorithmic cost estimation model in software engineering. It helps in quantifying the effort and resources needed for software development based on project size, complexity, and other factors.

Lab Experiment Overview:

1. Introduction to COCOMO Model: The lab session begins with an introduction to the COCOMO model, explaining the different versions (Basic, Intermediate, and Advanced) and their application in software cost estimation.
2. Defining the Sample Project: Students are provided with a sample software project along with its functional and non-functional requirements, complexity, and size metrics.
3. COCOMO Parameters: Students learn about the COCOMO model parameters, such as Effort Adjustment Factor (EAF), Scale Factors, and Cost Drivers, and how they influence the project's effort estimation.
4. Effort and Duration Estimation: Using the COCOMO model formula, students estimate the effort and duration required to complete the sample project based on the provided size and complexity metrics.
5. Resource Allocation: Students estimate the number of required resources, such as developers, testers, and project managers, based on the calculated effort and project duration.
6. Sensitivity Analysis: Students perform sensitivity analysis by varying the COCOMO parameters to observe their impact on the project cost estimation.
7. Conclusion and Reflection: Students discuss the significance of COCOMO in software project estimation and reflect on their experience in estimating project cost using the COCOMO model.

Learning Outcomes: By the end of this lab experiment, students are expected to:

· Understand the COCOMO model and its application in software cost estimation.
· Gain practical experience in using the COCOMO model to estimate effort, duration, and resources for a software project.
· Learn to consider various project factors and adjust COCOMO parameters for accurate cost estimation.
· Develop estimation skills for resource allocation and project planning.
· Appreciate the importance of data accuracy and project size metrics in project cost estimation.

Pre-Lab Preparations: Before the lab session, students should familiarize themselves with the COCOMO model, its parameters, and the cost estimation formula. They should also review the factors that influence the project's size and complexity.

Materials and Resources:

· Project brief and details for the sample software project
· COCOMO model guidelines and cost estimation formula
· Calculators or spreadsheet software for performing calculations

Conclusion: The lab experiment on estimating project cost using the COCOMO model provides students with practical insights into software cost estimation techniques. By applying the COCOMO model to a sample software project, students gain hands-on experience in assessing effort, duration, and resource requirements. The sensitivity analysis allows them to understand the impact of various factors on cost estimation. The lab experiment encourages students to use COCOMO in real-world scenarios, promoting informed decision-making in software project planning and resource allocation. Accurate cost estimation using COCOMO enhances project management and contributes to the successful execution of software engineering projects.

RESULTS:

Monte Carlo Risk Off
Auto Calculate Off

COCOMO II - Constructive Cost Model

Software Size Sizing Method Source Lines of Code

	SLOC	% Design Modified	% Code Modified	% Integration Required	Assessment and Assimilation (0% - 8%)	Software Understanding (0% - 50%)	Unfamiliarity (0-1)
New	60000						
Reused	0	0	0	0%	3%		
Modified	0	10%	10%	0%	3%	40%	0.4

Software Scale Drivers

Precedentedness	Nominal	Architecture / Risk Resolution	Extra High	Process Maturity	High
Development Flexibility	High	Team Cohesion	High		

Software Cost Drivers

Product

Required Software Reliability	High
Data Base Size	Nominal
Product Complexity	Nominal
Developed for Reusability	Low
Documentation Match to Lifecycle Needs	High

Personnel

Analyst Capability	Nominal
Programmer Capability	Low
Personnel Continuity	Nominal
Application Experience	High
Platform Experience	Nominal
Language and Toolset Experience	Low

Platform

Time Constraint	High
Storage Constraint	Nominal
Platform Volatility	High

Project

Use of Software Tools	High
Multisite Development	Extra High
Required Development Schedule	Nominal

Maintenance Off

Software Labor Rates

Cost per Person-Month (Dollars)

Calculate

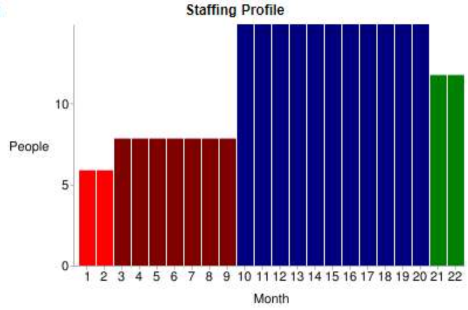
Results

Software Development (Elaboration and Construction)

Effort = 232.8 Person-months
Schedule = 19.0 Months
Cost = $0

Total Equivalent Size = 60000 SLOC
Effort Adjustment Factor (EAF) = 1.21

Staffing Profile



Acquisition Phase Distribution

Phase	Effort (Person-months)	Schedule (Months)	Average Staff	Cost (Dollars)
Inception	14.0	2.4	5.9	$0
Elaboration	55.9	7.1	7.8	$0
Construction	176.9	11.9	14.9	$0
Transition	27.9	2.4	11.7	$0

Software Effort Distribution for RUP/MBASE (Person-Months)

Phase/Activity	Inception	Elaboration	Construction	Transition
Management	2.0	6.7	17.7	3.9
Environment/CM	1.4	4.5	8.8	1.4
Requirements	5.3	10.1	14.2	1.1
Design	2.7	20.1	28.3	1.1
Implementation	1.1	7.3	60.2	5.3
Assessment	1.1	5.6	42.5	6.7
Deployment	0.4	1.7	5.3	8.4

Your output file is at http://softwarecost.org/tools/COCOMO/data/COCOMO_October_19_2023_15_00_37_215040.txt

Created by Ray Madachy at the Naval Postgraduate School. For more information contact him at rjmadach@nps.edu.

Post labs:

a) Analyse the COCOMO model and its different modes (Organic, Semi-detached, Embedded) to determine the most suitable mode for a specific project type:

- COCOMO Model Overview: Understand that the COCOMO (Constructive Cost Model) is a widely used software estimation model. It categorizes software projects into three modes: Organic, Semi-detached, and Embedded, based on various characteristics.
- Organic Mode: This mode is suitable for small to medium-sized projects. It assumes that the team is experienced, and requirements are well-understood. The process is well-structured, and there's not a significant need for innovation.
- Semi-detached Mode: For projects with some complexity but not extreme. It's a mid-point between organic and embedded. The team may have some experience, and requirements may be partially understood.
- Embedded Mode: Suitable for large, complex projects where innovation and new technologies are involved. The team may have limited experience, and requirements may be volatile.
- Selecting the Mode: Determine the most suitable mode by considering project size, complexity, team expertise, and the level of innovation required. A smaller, experienced team working on a project with well-defined requirements may fit the organic mode, while large, complex, innovative projects with less experienced teams may fit the embedded mode.

b) Apply the COCOMO model to estimate the project cost and effort required for a given software development project:

<u>1. Project Mode Selection:</u>

Considering the size and complexity, the project appears to be of medium size and complexity. It doesn't seem to be a small project (organic) or extremely large and complex (embedded). Hence, we can choose the Semi-detached mode.

<u>2. Gather Parameters:</u>

SLOC (Source Lines of Code): Based on your SRS, you mentioned a Software Size of 60,000 SLOC.

% Design Modified: As this is a new project, it's unlikely that design modifications are required. So, let's assume 0%.

% Code Modified: Similarly, for a new project, there shouldn't be any modifications. Let's assume 0%.

% Integration Required: As integration is mentioned, we'll assume that some level of integration is required. Let's use 0%.

Assessment and Assimilation (Software Understanding): For a new project, this can be relatively low. Let's assume 4%.

<u>3. Use COCOMO Formula:</u>

We'll use the basic COCOMO formula to calculate Effort and Schedule for the Semi-detached mode:

Effort (E) = a * (KLOC)^b * EAF

Schedule (T) = c * (E)^d

Where:

EAF (Effort Adjustment Factor) is a function of various cost drivers (e.g., personnel capabilities, development tools). For simplicity, we'll assume EAF as 1.0 (no adjustments).

a, b, c, and d are mode-specific constants. For Semi-detached mode, they are approximately a=3.0, b=1.12, c=2.5, and d=0.35.

c) Evaluate the factors influencing COCOMO estimates, such as project size, personnel capabilities, and development tools, and their implications on project planning and scheduling:

- Project Size: Larger projects typically require more effort, time, and cost. Smaller projects in the organic mode may have shorter schedules.
- Personnel Capabilities: Highly skilled teams may be more efficient, reducing effort and cost. Less experienced teams may need more resources.
- Development Tools: Efficient tools can increase productivity and reduce effort. However, tool integration and learning curve can affect the schedule.
- Implications: These factors have significant implications for project planning and scheduling. Understanding these factors helps in making realistic estimates and setting project expectations.