COCOMO Estimation for Library Management System

You will use the COCOMO 81 Intermediate Model to estimate the effort and timeline for building the Library Management System. This exercise will help you understand software cost estimation and project planning.

LEARNING OBJECTIVES

1. **Understand how software size affects project effort**
2. **Learn about cost drivers and their impact**
3. **Estimate person-months and project duration**
4. **Compare estimation with reality**
5. **Make informed project planning decisions**

BEFORE YOU START

What is COCOMO?

COCOMO = COnstructive COst MOdel

- **A mathematical model to estimate software development effort**
- **Created by Barry Boehm in 1981**
- **Used by thousands of companies worldwide**

Key Terms:

- **SLOC: Source Lines of Code**
- **Person-Month: One person working for one month**
- **EAF: Effort Adjustment Factor (multiplier based on project conditions)**
- **Organic/Semi-Detached/Embedded: Project complexity levels**

STEP-BY-STEP INSTRUCTIONS

PART 1: ACCESS THE COCOMO CALCULATOR

Option A: Online Calculator

1. **Go to: http://softwarecost.org/tools/COCOMO81/index.php**
2. **Look for "COCOMO 81 Intermediate Model"**

PART 2: GATHER PROJECT INFORMATION

For our Library Management System, we need:

1. Software Size (SLOC)

What is SLOC? Count of all lines in your source code.

Our estimate:

- **Basic system: 10k – 15k SLOC**
- **Medium system: 15k – 25k SLOC**
- **Advanced system: 25k – 40k SLOC**

Your task: Choose a number between 10k and 40k

- **Example: 12k SLOC (for basic system)**
- **Write down: I choose _____ SLOC**

2. Development Mode
Three options:

Mode	When to Choose	Example Projects
Organic	Small team, familiar environment	Simple website, internal tools
Semi-Detached	Mixed experience, some unknowns	E-commerce site, mobile app
Embedded	Strict constraints, real-time	Air traffic control, medical devices

For Library System:

- **Most student projects: Organic or Semi-Detached**
- **If adding complex features: Semi-Detached**
- **Your choice: □ Organic □ Semi-Detached □ Embedded**

EXERCISE: FILL COST DRIVERS
You will rate 15 factors from Very Low (VL) to Extra High (XH).
Product Attributes (3 factors)
1. Required Reliability
How critical are bugs?

- **Very Low (0.75): Simple internal tool, bugs are okay**
- **Nominal (1.00): Standard business application**
- **High (1.15): Important system, moderate cost of failure**
- **Very High (1.40): High cost of failure**

Library System: Users can't check out books if system crashes. Choose: High (1.15)

2. Database Size
How big is the database compared to code?

- **Low (0.94): Small database**
- **Nominal (1.00): Medium database**
- **High (1.08): Large database**
- **Very High (1.16): Very large database**

Library System: Books + members + transactions. Choose: High (1.08)

3. Product Complexity
How complex is the software?

- **Low (0.85): Simple data entry**
- **Nominal (1.00): Standard business system**
- **High (1.15): Complex logic**
- **Very High (1.30): Highly complex**
- **Extra High (1.65): Extremely complex**

Library System: Search, checkout, returns, fines, reports. Choose: High (1.15)

Computer Attributes (4 factors)
4. Execution Time Constraint
How much performance is needed?

- **Nominal (1.00): Normal response time**
- **High (1.11): Faster than normal**
- **Very High (1.30): Real-time constraints**
- **Extra High (1.66): Extreme performance**

Library System: Should be fast but not critical. Choose: Nominal (1.00)

5. Main Storage Constraint
Memory limitations?
- **Nominal (1.00): Standard memory**
- **High (1.06): Some limitations**
- **Very High (1.21): Severe limitations**
- **Extra High (1.56): Extreme limitations**

Library System: Standard computers. Choose: Nominal (1.00)

6. Virtual Machine Volatility
How stable is the platform?
- **Low (0.87): Very stable**
- **Nominal (1.00): Some changes**
- **High (1.15): Frequent changes**
- **Very High (1.30): Constant changes**

Library System: Using stable technologies. Choose: Low (0.87)

7. Computer Turnaround Time
How fast can you test/run code?
- **Low (0.87): Fast turnaround**
- **Nominal (1.00): Normal**
- **High (1.07): Slow**
- **Very High (1.15): Very slow**

Library System: Local development, fast testing. Choose: Low (0.87)

Personnel Attributes (5 factors)
8. Analyst Capability
Experience of analysts?
- **Very Low (1.46): No experience**
- **Low (1.19): Some experience**
- **Nominal (1.00): Average**
- **High (0.86): Experienced**
- **Very High (0.71): Expert**

For students: Choose: Low (1.19) or Nominal (1.00)

9. Applications Experience
Experience with similar projects?
- **Very Low (1.29): None**
- **Low (1.13): Some**
- **Nominal (1.00): Average**
- **High (0.91): Experienced**
- **Very High (0.82): Expert**

For students: This is your first LMS. Choose: Very Low (1.29)

10. Programmer Capability
Programming skills?
- **Very Low (1.42): Beginners**
- **Low (1.17): Below average**
- **Nominal (1.00): Average**
- **High (0.86): Good**

- **Very High (0.70): Excellent**

For students: Choose based on your skills: Low (1.17) to Nominal (1.00)

11. Virtual Machine Experience
Experience with development tools?
- **Very Low (1.21): None**
- **Low (1.10): Some**
- **Nominal (1.00): Average**
- **High (0.90): Experienced**

For students: Using standard tools. Choose: Low (1.10)

12. Programming Language Experience
Experience with chosen language?
- **Very Low (1.14): None**
- **Low (1.07): Some**
- **Nominal (1.00): Average**
- **High (0.95): Experienced**

For students: Java/Python/JavaScript experience. Choose: Nominal (1.00)

Project Attributes (3 factors)
13. Modern Programming Practices
Using good development practices?
- **Very Low (1.24): No practices**
- **Low (1.10): Some practices**
- **Nominal (1.00): Standard**
- **High (0.91): Good practices**
- **Very High (0.82): Excellent practices**

For students: Learning as you go. Choose: Low (1.10)

14. Use of Software Tools
Quality of development tools?
- **Very Low (1.24): Basic tools**
- **Low (1.10): Some tools**
- **Nominal (1.00): Standard IDE**
- **High (0.91): Good tools**
- **Very High (0.83): Excellent tools**

For students: Standard IDE. Choose: Nominal (1.00)

15. Required Development Schedule
How compressed is the schedule?
- **Very Low (1.23): Very relaxed**
- **Low (1.08): Some pressure**
- **Nominal (1.00): Standard**
- **High (1.04): Tight**
- **Very High (1.10): Very tight**

For students: Semester project deadline. Choose: High (1.04)

USING THE ONLINE CALCULATOR {Library Management System}

Step 1: Enter Basic Information

Software Product Size: [Enter your SLOC, e.g., 12000]

Software Development Mode: [Choose Organic/Semi-Detached/Embedded]

Step 2: Select All Cost Driver Ratings

Click on each dropdown and select your rating from:

- **Very Low (VL)**
- **Low (L)**
- **Nominal (N)**
- **High (H)**
- **Very High (VH)**
- **Extra High (XH)**

Step 3: Calculate

Click "Calculate" or "Estimate" button.

Step 4: Record Results

You'll see:

Effort = _____ Person-Months

Schedule = _____ Months

Effort Adjustment Factor (EAF) = _____

Write down your results:

EXERCISE WORKSHEET [Event Management Platform]

Parameter	Your Choice	Value/Multiplier
SLOC	_____	_____
Mode	_____	**a= , *b*= , *c*= , *d*=**
Reliability	_____	_____
Database Size	_____	_____
Complexity	_____	_____
Execution Time	_____	_____
Storage	_____	_____
VM Volatility	_____	_____
Turnaround Time	_____	_____
Analyst Capability	_____	_____
Applications Exp	_____	_____
Programmer Capability	_____	_____
VM Experience	_____	_____
Language Exp	_____	_____
Programming Practices	_____	_____
Software Tools	_____	_____
Schedule Pressure	_____	_____

Results:

- **EAF = _____**
- **Effort = _____ person-months**
- **Schedule = _____ months**
- **Team Size = _____ people Effort/schedule**

ANALYSIS QUESTIONS

Answer these after calculation:

1. **How many person-months did you get? _____**
    - **Is this more or less than you expected?**
2. **How long will the project take? _____ months**
    - **Can this be done in one semester (4 months)?**
3. **What's the ideal team size? _____ people**
    - **Do you have enough people?**
4. **Which cost driver increased effort the most? _____**
    - **Why do you think this factor is so important?**
5. **Which cost driver decreased effort the most? _____**
    - **How can you improve this factor?**
6. **What would happen if you halved the SLOC (made the project smaller)?**
    - **New effort: _____**
    - **New schedule: _____**
7. **What if you improved team experience (change "Programmer Capability" to High)?**
    - **New effort: _____**
    - **Percentage change: _____%**