Project: Stock Recording System

CSE 5325 – Fall 2024

Project Management

Module: COCOMO

Deliverable: COCOMO Estimate Report

Version: [1.0] **Date: [11/07/2024]**

Prepared by:

Raviteja Avutapalli

Mav Id:1002163445

TABLE OF CONTENTS

1. Introduction

The Stock Recording System (SRS) Project is a comprehensive initiative designed with a secure, efficient platform for tracking and managing stock trading activities. The project aims to deliver a user-friendly system accessible via both a web application and an Android mobile application. The platform's core functionalities include user registration, real-time stock watchlist updates, transaction recording, capital gain and loss calculations, and notifications for significant stock events. These capabilities will allow users to monitor their investments, record transactions, and track performance over time.

The primary objective of this document is to estimate the cost, time, and resources required to develop the SRS project using the COCOMO II (Constructive Cost Model II) framework.

COCOMO II is a widely recognized software cost estimation model that helps project managers and developers predict the required effort, development time, and overall cost based on software size and project-specific attributes. COCOMO II is an industry-standard tool for predicting software development costs based on software size, project-specific attributes, and productivity factors. By analyzing five Scale Drivers and seventeen Cost Drivers, this report provides a comprehensive cost estimate based on the project’s unique requirements and complexity.

The final cost estimate includes human resources, non-human resource costs, overhead, and a 100% profit margin, offering a realistic budget for the client. The document’s structured approach ensures clear insights into the project’s timeline, cost drivers, and potential risks.

Key factors covered include Source Lines of Code (SLOC), estimated based on projected functionality, and various Scale and Cost Drivers that affect productivity and resource needs. This structured estimation process enables an accurate prediction of project effort, time, and financial requirements, critical for planning and resource allocation.

This document presents the cost estimation of SRS using COCOMO II. The cost estimation relies on factors such as Source Lines of Code,5 Scale Drivers and 17 Cost Drivers.

The estimation is carried out using the System Star tool, which makes use of the previously specified scale and cost drivers. This tool provides insight into the project's effort, duration, cost, and productivity.

The **Constructive Cost Model** (**COCOMO**) is a procedural software cost estimation model developed by Barry W. Boehm. The model parameters are derived from fitting a regression formula using data from historical projects (63 projects for COCOMO 81 and 163 projects for COCOMO II).

Types of Projects in the COCOMO Model

The COCOMO model classifies software projects into three types based on complexity, team size, and development environment:

1. **Organic**: Small teams work on familiar, well-understood problems with nominal experience requirements.
2. **Semi-detached**: Projects with moderate team size and complexity, requiring more experience and creativity than organic projects. Examples include compilers and certain embedded systems.
3. **Embedded**: The most complex projects with large teams and high experience levels, demanding substantial creativity and expertise.

The Intermediate Cocomo formula now takes the form:

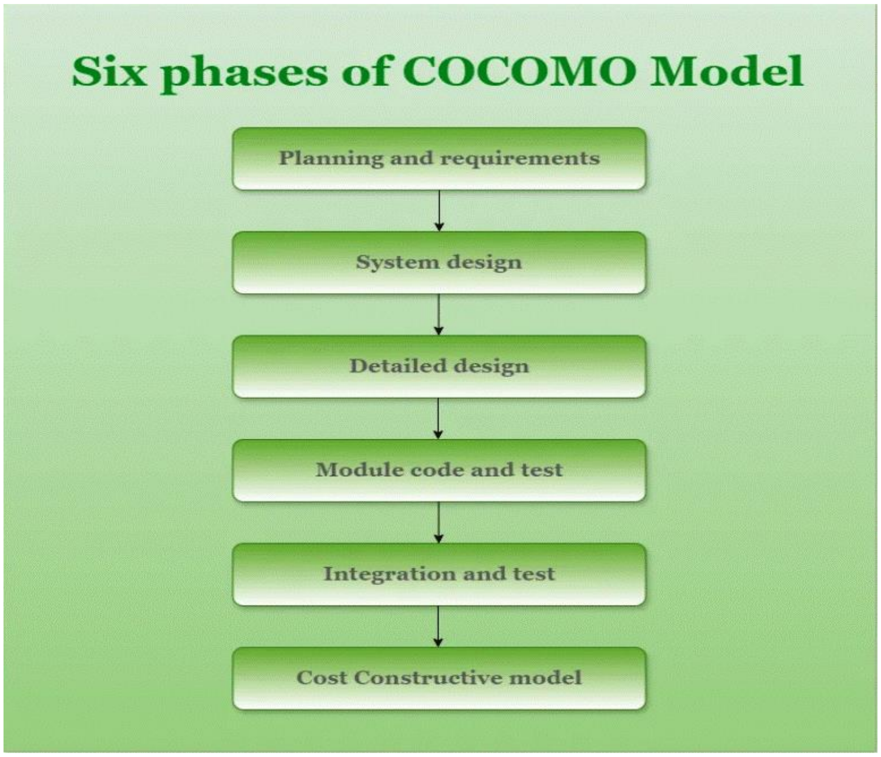
$E = a_i(\text{KLoC})^{b_i}(\text{EAF})$

where E is the effort applied in person-months, **KLoC** is the estimated number of thousands of delivered lines of code for the project, and **EAF** is the factor calculated above. The coefficient a_i and the exponent b_i are given in the next table.

Software project	**a_i**	**b_i**	**c_i**
Organic	3.2	1.05	0.38
Semi-detached	3.0	1.12	0.35
Embedded	2.8	1.20	0.32

The Development time D and the most effective number of Persons P calculation uses E in the same way as in the Basic COCOMO:

$D = 2.5\ E^{c_i}$



Detailed COCOMO Model

The Detailed COCOMO model builds on the Intermediate COCOMO by analyzing the impact of each cost driver at every software development phase, providing granular estimates. This model divides the project into separate modules, allowing for a more precise effort estimation by aggregating effort across all modules.

Phases of Detailed COCOMO

1. **Planning & Requirements**: This phase sets the foundation by defining the project's scope, objectives, and constraints, ensuring clarity from the start. It includes preparing a detailed project plan outlining schedules, resources, and milestones.

2. **System Design**: In this stage, the software's high-level architecture is developed, defining major components, their interactions, and data flow to form a blueprint for implementation.

3. **Detailed Design**: Each system component is specified in detail, covering data structures, algorithms, and interfaces to guide the development phase accurately.

4. **Module Coding & Testing**: Here, the code is written and tested for each module according to the detailed specifications, translating design into a functional form.

5. **Integration & Testing**: Individual modules are integrated, and the combined system is tested to ensure that components work cohesively as a complete solution.

6. **Cost Constructive Model**: This model quantifies the cost and effort needed for a project, making it useful for budgeting and time planning.

Importance of COCOMO

1. **Cost Estimation**: COCOMO provides a structured approach to estimating costs and effort, which is vital for financial planning and resource management.

2. **Resource & Project Management**: By accounting for project size, complexity, and team experience, COCOMO helps allocate resources effectively and plan realistic project timelines.

3. **Benchmarking**: COCOMO allows projects to be compared against industry standards, assisting organizations in assessing and improving their software development processes.

4. **Decision Support**: It offers quantitative insights for decision-making, particularly regarding project scope, priorities, and resource distribution.

Types of COCOMO Models

1. **Basic COCOMO**: This model offers a quick estimation method based solely on project size in KLOC, making it suitable for straightforward projects.

2. **Intermediate COCOMO**: This model refines estimates by considering additional factors, or "cost drivers," like system reliability and team experience.

3. **Detailed COCOMO**: Extending beyond Basic and Intermediate, this model considers comprehensive project factors such as team skills and software complexity, providing the most accurate estimation.

Applications of COCOMO

1. **NASA**: COCOMO was used in planning the software for the Space Shuttle, helping NASA allocate resources and schedule accurately based on project size and complexity.

2. **Businesses**: Large corporations use COCOMO to plan the development of complex software systems, facilitating resource allocation and budgeting.

3. **Commercial Software & Academic Research**: COCOMO supports cost and time estimation for new products or academic prototypes, helping organizations make decisions on investment and development timelines.

Advantages

1.COCOMO provides a systematic way to estimate project costs and efforts, helping identify the highest-impact factors on project success.

2.It also assists in evaluating project feasibility by forecasting the cost and time required at different development stages.

Limitations

The model assumes that project size is the main cost determinant, which may overlook unique team factors that significantly affect outcomes. COCOMO's reliance on general assumptions and averages means estimates may lack precision for highly customized projects.

Best Practices

1. **Customize Inputs**: Adjust COCOMO’s parameters to reflect specific project needs and conditions for more accurate estimations.

2. **Use Historical Data**: Improve accuracy by using past project data to refine parameters, aligning COCOMO outputs with real-world outcomes.

3. **Validate Estimates**: Regularly compare estimates with actual project results, adjusting the model to improve future predictions.

4. **Combine with Other Methods**: Pair COCOMO with other estimation techniques, like expert judgment or bottom-up estimation, to enhance accuracy and mitigate bias.

2. Estimating Factors

2.1 Source of Lines of Code

The number of lines of code that were delivered as part of this project is as follows. A justification for the overall quantity of LOC is given.

SLOC | Source Lines of Code	Value Chosen:5000
Justification: The SRS project involves multiple components (registration, watchlist, transaction records, notifications) that necessitate an optimized codebase for both the website and Android app. Based on similar financial software projects, the estimated codebase size is approximately 5,000 lines of highly optimized code.	

2.2 Scale Drivers

The following is the list of scale drivers, the values applicable to this project and a justification for each value chosen:

PREC | Precedentedness	Value Chosen: **Largely Familiar**
Justification: This driver measures the team's familiarity with the project type and its requirements. For the SRS project, the team has moderate experience with financial systems and similar applications, allowing them to leverage past knowledge and practices. Hence, **PREC is rated as "Largely Familiar,"** as the team's experience with comparable systems facilitates understanding and reduces the learning curve.	

FLEX | Development Flexibility	Value Chosen: **General Conformity**
Justification: FLEX assesses how strictly the project requirements are defined and the extent to which they may evolve. The SRS project will follow an agile methodology, allowing the team to incorporate client feedback throughout the development process. Although requirements are somewhat flexible, they still adhere to certain fixed standards due to regulatory and security needs. Thus, **FLEX is rated as "General Conformity,"** indicating a balance between client-driven adaptability and fixed project goals.	

RESL | Architecture / Risk Resolution	Value Chosen: **Mostly**
Justification: RESL evaluates the level of planning for potential risks and architectural challenges The SRS project involves handling sensitive financial data, which introduces security and integrity risks. To manage these effectively, the team will emphasize robust architecture and comprehensive risk management strategies early in the project. As a result, **RESL is rated as "Moderately Flexible,"** reflecting a proactive but balanced approach to managing complexity and risk.	

TEAM | Team Cohesion	Value Chosen: **Largely Cooperative**
Justification: TEAM considers the team’s ability to work well together, which is essential for effective collaboration and productivity. For the SRS project, the team has previously collaborated on similar projects, fostering strong communication and trust. Frequent client interactions further support cohesion and alignment. **TEAM is rated as "Largely Cooperative,"** acknowledging that existing team synergy will support efficient coordination and a collaborative environment.	

PMAT | Process Maturity	Value Chosen: **SEI CMM Level 4**
Justification: The team's experience with React and Android development, combined with a proven process for client satisfaction, reflects a strong foundation. The organization's extensive portfolio and meticulous process adherence align with SEI CMM Level 4, indicating a mature and well-defined development approach.	

2.3 Cost Drivers

Cost Drivers are 17 specific factors in the COCOMO II model that influence the time and effort needed for a project. These drivers are grouped into categories based on personnel expertise, product requirements, platform constraints, and project management. Below is a breakdown of each Cost Driver with justifications for the SRS project:

PERSONNEL:

<table>
<tr><td><b>ACAP | Analyst Capability</b></td><td>Value Chosen: <b>High</b></td></tr>
<tr><td colspan="2">Justification: The project manager and analysts have strong skills in requirements gathering and data analysis, ensuring high-quality specifications for the financial features of the SRS project. Their expertise will help avoid misunderstandings and rework, streamlining development.</td></tr>
</table>

<table>
<tr><td><b>APEX | Applications Experience</b></td><td>Value Chosen: <b>High</b></td></tr>
<tr><td colspan="2">Justification: The team has extensive experience with financial applications, giving them a solid understanding of the specific requirements and challenges in the stock trading domain. This reduces the learning curve and potential issues during development.</td></tr>
</table>

<table>
<tr><td><b>PCAP | Programmer Capability</b></td><td>Value Chosen: <b>High</b></td></tr>
<tr><td colspan="2">Justification: The developers possess strong programming skills and are proficient in the languages and frameworks needed for web and Android applications. This expertise will contribute to efficient, high-quality coding with fewer errors.</td></tr>
</table>

<table>
<tr><td><b>PLEX | Platform Experience</b></td><td>Value Chosen: <b>Very High</b></td></tr>
<tr><td colspan="2">Justification: The development team is well-versed in the target platforms, including web development and Android, as well as necessary tools and environments. This familiarity enhances productivity, allowing the team to leverage platform-specific features effectively.</td></tr>
</table>

<table>
<tr><td><b>LTEX | Language and Tool Experience</b></td><td>Value Chosen: <b>Nominal</b></td></tr>
<tr><td colspan="2">Justification: While the team has general experience with necessary tools and languages, some aspects of the technology stack may require minor training or additional support, especially for specific security and encryption libraries.</td></tr>
</table>

PCON | Personnel Continuity	Value Chosen: **Very High**
Justification: Personnel turnover is expected to be minimal due to the team’s commitment and project duration. This stability ensures continuity in knowledge and reduces the time required for new team members to adjust.	

PLATFORM:

TIME | Execution Time Constraint	Value Chosen: **High**
Justification: Real-time stock updates and quick response times are critical for user experience in the SRS. The platform must process and display data with minimal delay, particularly for stock watchlists and notifications.	

STOR | Main Storage Constraint	Value Chosen: **Nominal**
Justification: Storage requirements are within typical constraints for a financial application, with data primarily focused on transaction logs and user portfolios. Optimizations will ensure efficient storage, keeping this factor at a manageable level.	

PVOL | Platform Volatility	Value Chosen: **Low**
Justification: The platform (web and Android) is relatively stable, with minimal anticipated changes in underlying technology. This stability allows developers to focus on functional requirements without frequent adjustments to accommodate platform updates.	

PRODUCT:

RELY | Required Reliability	Value Chosen: **High**
Justification: Given that SRS handles sensitive financial data, high reliability is essential. Any malfunction or data inconsistency could lead to financial loss for clients. Therefore, stringent testing and robust error handling are prioritized.	

DATA | Database Size	Value Chosen: **High**
Justification: The SRS project requires a large database to store extensive information, including user transactions, portfolio data, and historical trade records. This high data requirement influences database design and performance, adding complexity. The rating is calculated as D/P, which is the ratio of database size to source lines of code (SLOC). Because of the high number of users, the database is huge, with 5000 SLOC. As a result, a high value was chosen.	

CPLX | Product Complexity	Value Chosen: **Nominal**
Justification: The complexity of SRS is moderate, as it combines standard features for financial tracking with added functionalities for notifications, real-time updates, and security. While complex, these features are manageable for the experienced team.	

RUSE | Required Reusability	Value Chosen: **Nominal**
Justification: Some components, such as authentication and notifications, may be reused in future applications. However, the project mainly focuses on custom financial functionalities specific to SRS, so reusability requirements are moderate. While some new components will be required, this existing foundation allows for a "Nominal" development complexity rating.	

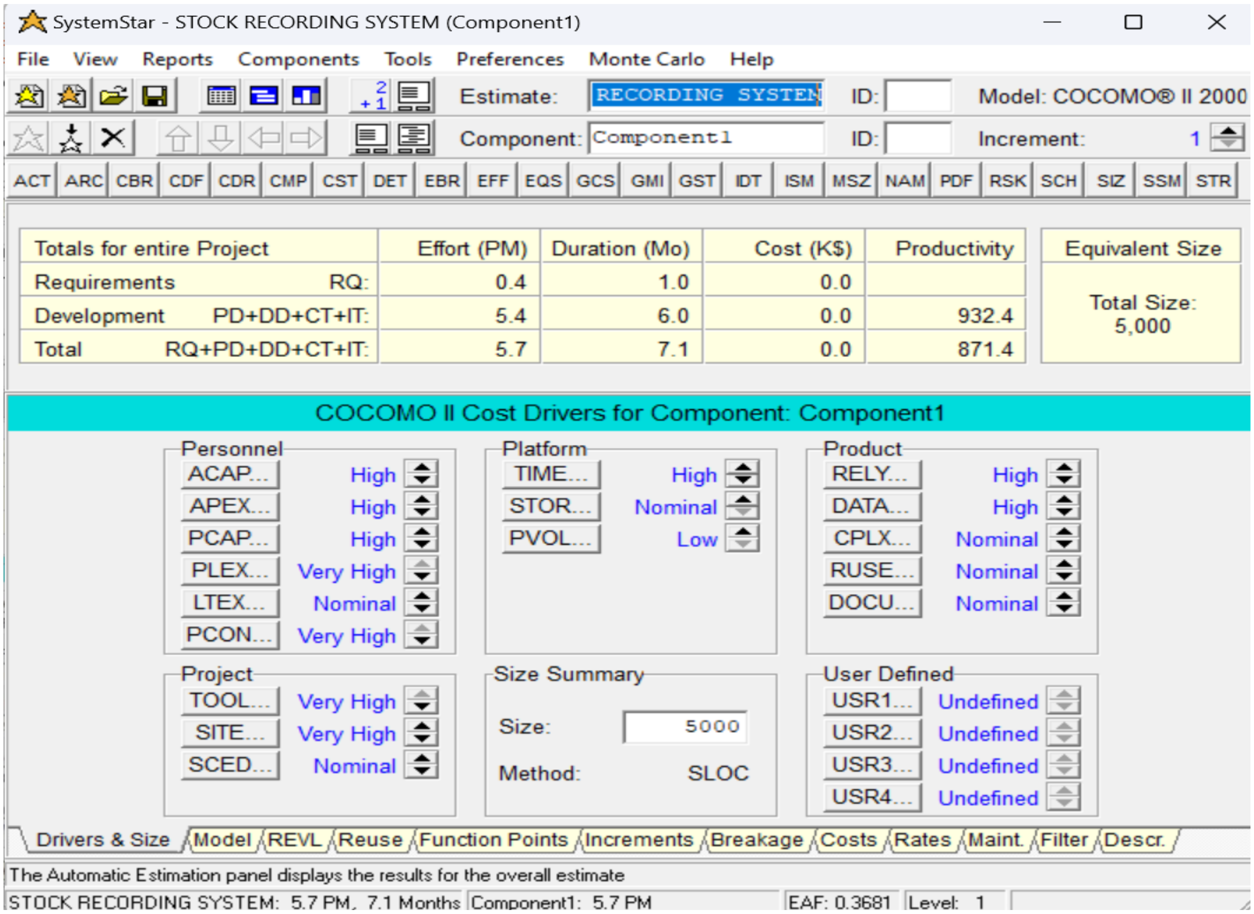
DOCU | Documentation match to life cycle needs	Value Chosen: **Nominal**
Justification: The project documentation is considered "Nominal" as it comprehensively covers all system functionalities and meets the project lifecycle needs. The project requires thorough documentation for client handover and maintenance. The documentation will cover system architecture, API details, and user guidelines, ensuring clear instructions for future updates or maintenance.	

PROJECT:

<table>
<tr><td><b>TOOL | Use of Software Tools</b></td><td>Value Chosen: <b>Very High</b></td></tr>
<tr><td colspan="2">Justification: Mature, integrated development tools (Very High) boost efficiency and reuse (prior project experience). The project will leverage mature, integrated development environments (IDEs), version control, testing frameworks, and security tools to enhance productivity and ensure quality. These tools are vital for a project of this scope and complexity, especially given the financial domain’s rigorous requirements.</td></tr>
</table>

<table>
<tr><td><b>SITE | Multisite Development</b></td><td>Value Chosen: <b>Very High</b></td></tr>
<tr><td colspan="2">Justification: The Multiside development involves managing a network of websites under one central platform . While the project may be developed across multiple sites or involve remote team members, robust collaboration tools and clear communication protocols are in place to ensure effective teamwork. The project manager’s oversight will help coordinate tasks across locations.</td></tr>
</table>

<table>
<tr><td><b>SCED | Development Schedule</b></td><td>Value Chosen: <b>Nominal</b></td></tr>
<tr><td colspan="2">Justification: The development schedule is considered "Nominal" due to its flexibility and alignment with project requirements. The development schedule is aligned with the project’s complexity and deliverables. While the timeline is structured to meet client deadlines, it allows sufficient flexibility to manage minor delays or challenges without significant impact.</td></tr>
</table>


SystemStar - STOCK RECORDING SYSTEM (Component1)
File View Reports Components Tools Preferences Monte Carlo Help
ACT ARC CBR CDF CDR CMP CST DET EBR EFF EQS GCS GMI GST IDT ISM MSZ NAM PDF RSK SCH SIZ SSM STR
Drivers & Size / Model / REVL / Reuse / Function Points / Increments / Breakage / Costs / Rates / Maint. / Filter / Descr.

Estimate: RECORDING SYSTEM ID: Model: COCOMO® II 2000

Component: Component1 ID: Increment: 1

Totals for entire Project		Effort (PM)	Duration (Mo)	Cost (K$)	Productivity	Equivalent Size
Requirements	RQ:	0.4	1.0	0.0		Total Size: 5,000
Development	PD+DD+CT+IT:	5.4	6.0	0.0	932.4	
Total	RQ+PD+DD+CT+IT:	5.7	7.1	0.0	871.4	

COCOMO II Cost Drivers for Component: Component1

Personnel
ACAP... High
APEX... High
PCAP... High
PLEX... Very High
LTEX... Nominal
PCON... Very High

Platform
TIME... High
STOR... Nominal
PVOL... Low

Product
RELY... High
DATA... High
CPLX... Nominal
RUSE... Nominal
DOCU... Nominal

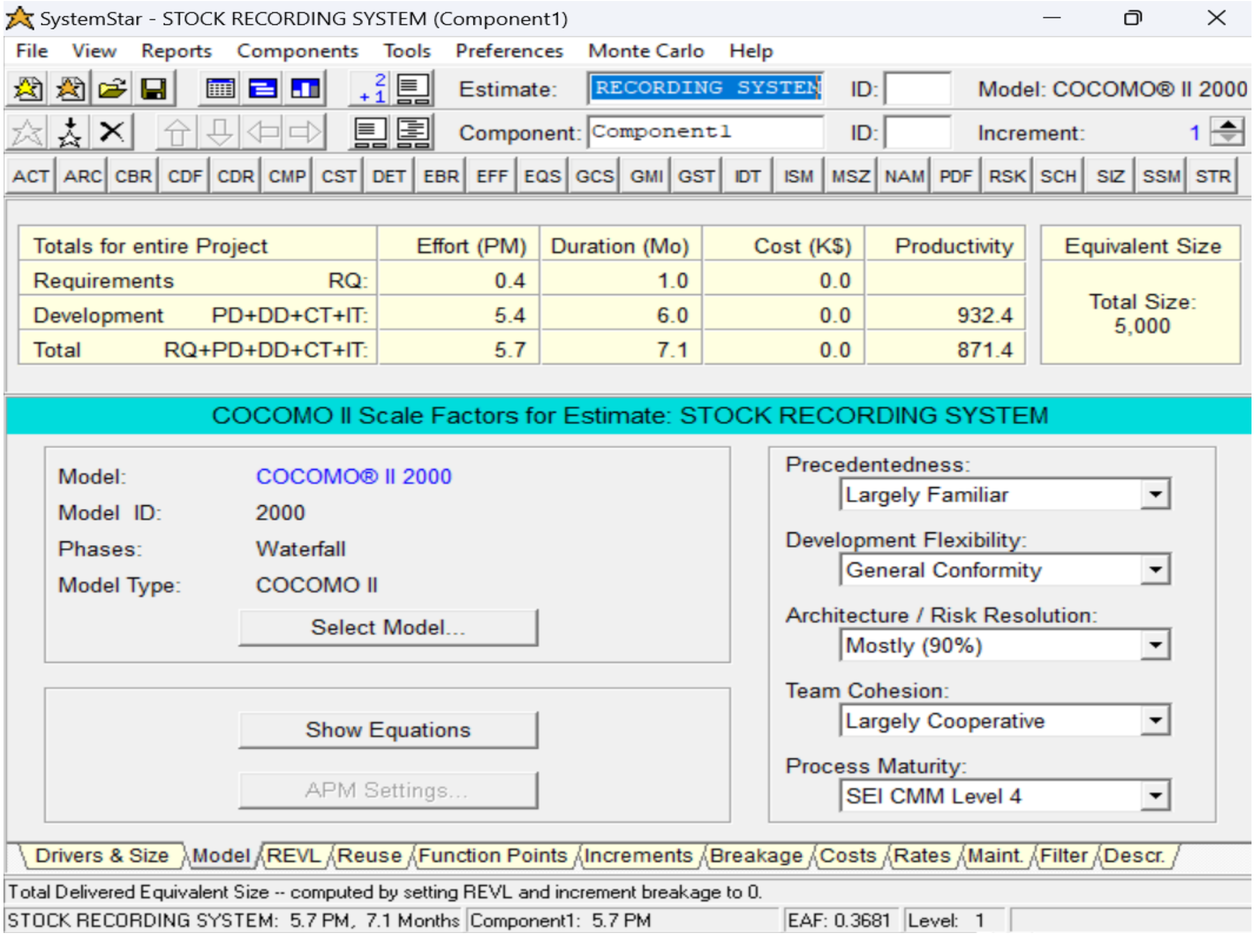
Project
TOOL... Very High
SITE... Very High
SCED... Nominal

Size Summary
Size: 5000
Method: SLOC

User Defined
USR1... Undefined
USR2... Undefined
USR3... Undefined
USR4... Undefined

The Automatic Estimation panel displays the results for the overall estimate

STOCK RECORDING SYSTEM: 5.7 PM, 7.1 Months Component1: 5.7 PM EAF: 0.3681 Level: 1


SystemStar - STOCK RECORDING SYSTEM (Component1)
File View Reports Components Tools Preferences Monte Carlo Help
ACT ARC CBR CDF CDR CMP CST DET EBR EFF EQS GCS GMI GST IDT ISM MSZ NAM PDF RSK SCH SIZ SSM STR
Select Model...
Show Equations
APM Settings...
Drivers & Size / Model / REVL / Reuse / Function Points / Increments / Breakage / Costs / Rates / Maint. / Filter / Descr.

Estimate: RECORDING SYSTEM ID: Model: COCOMO® II 2000

Component: Component1 ID: Increment: 1

Totals for entire Project		Effort (PM)	Duration (Mo)	Cost (K$)	Productivity	Equivalent Size
Requirements	RQ:	0.4	1.0	0.0		Total Size: 5,000
Development	PD+DD+CT+IT:	5.4	6.0	0.0	932.4	
Total	RQ+PD+DD+CT+IT:	5.7	7.1	0.0	871.4	

COCOMO II Scale Factors for Estimate: STOCK RECORDING SYSTEM

Model: COCOMO® II 2000
Model ID: 2000
Phases: Waterfall
Model Type: COCOMO II

Precedentedness: Largely Familiar
Development Flexibility: General Conformity
Architecture / Risk Resolution: Mostly (90%)
Team Cohesion: Largely Cooperative
Process Maturity: SEI CMM Level 4

Total Delivered Equivalent Size -- computed by setting REVL and increment breakage to 0.

STOCK RECORDING SYSTEM: 5.7 PM, 7.1 Months Component1: 5.7 PM EAF: 0.3681 Level: 1

3 Project Final Timeline and Cost Structure

· PREVIOUS COST, WORK AND DURATION

Work	Durations
Project Initiation Phase	**7 days**
Requirements	**7 days**
System Design Phase	**10 days**
Development	**21 days**
Testing	**15 days**
Deployment, Documentation and Training	**7 days**
Maintenance and Support Setup	**3 days**
Total Duration	**70 days**

Estimated Human Resource Cost

RESOURCE	Monthly	3 Months
Developer-1	$5000	$15000
Developer-2	$5000	$15000
Developer-3	$5000	$15000
Developer-4 (Tester 1)	$5000	$15000
Developer-5(Tester 2)	$5000	$15000
Project Manager	$10,000	$30,000
	Total	**$105,000**

Estimated time duration of each employee

EMPLOYEE TYPE	Start Date	End Date	Total Duration
Developer-1	09/03/2024	12/03/2024	345 Hours
Developer-2	09/03/2024	12/03/2024	345 Hours
Developer-3	09/03/2024	12/03/2024	360 Hours
Developer-4	09/03/2024	12/03/2024	336 Hours
Developer-5	09/03/2024	12/03/2024	324 Hours
Project Manager	09/03/2024	12/03/2024	384 Hours

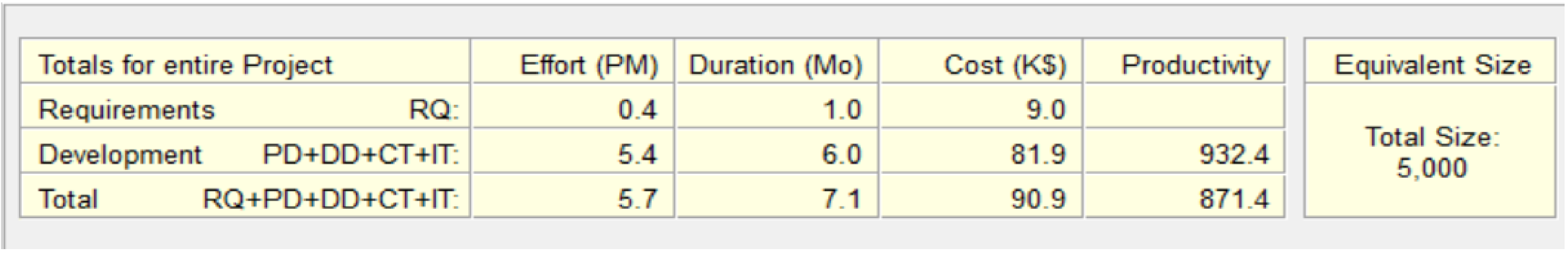
Estimated Non-Human Resource cost

NON HUMAN RESOURCES TYPE	ESTIMATED COST
Software	$2,000.00
Working Systems	$12,000.00
Database Setup	$10,000.00
Health Insurance	$7,000.00
Cost of Utilities	$8,000.00
Cost of Infrastructure(Building Cost)	$53,000.00
Travel	$2,000.00
Miscellaneous charges	$1,000.00

New Schedule : COCOMO ESTIMATES Duration, Cost

Totals for entire project	Effort (PM)	Duration (Months)	Cost(K$)
Requirements	0,4	1,0	9,0
Development	5,4	6,0	81,9
Total	5,7	7,1	83,6

The project timeline has been extended from 3 months to 7.1 months. This adjustment is primarily due to cost-related factors, which can vary based on elements such as taxes and possible salary changes for the project team. Although other aspects like code volume, platform stability, team expertise, and software tools affect development duration, cost was the primary reason for extending the timeline.



Totals for entire Project		Effort (PM)	Duration (Mo)	Cost (K$)	Productivity	Equivalent Size
Requirements	RQ:	0.4	1.0	9.0		Total Size: 5,000
Development	PD+DD+CT+IT:	5.4	6.0	81.9	932.4	
Total	RQ+PD+DD+CT+IT:	5.7	7.1	90.9	871.4	

COCOMO estimated costs (Human Resources)

- There are 5 developers and a program manager. The monthly salary of the developers are $5000/month and project Manager is $10,000/month, respectively.

· **Non-Human Resources**

Non-human resources required like software, working systems ,database setup, utility cost ,health insurance etc. the average cost of such resources results to be $95K.

- Cost of product = $95k + $90.9K **= $ 185.9K**

· **Profit**

To get 100% profit

- 100% of total cost + total cost = 2*total cost
    - Final cost = 2 x $185.6K = **$371.8K**

· **Final Cost**

The final cost of would be **$371.8K**

COCOMO Estimated Project Costs

The COCOMO model offers a structured cost estimate for the Stock Recording System (SRS) project, broken down into Human Resources, Non-Human Resources, and Profit.

Human Resources Costs

The project team consists of five developers and one project manager. Each developer earns $5,000 per month, totaling $15,000 over the three-month project, while the project manager receives $10,000 monthly, totaling $30,000. This results in a total **human resources cost of $105,000** over the project duration.

Non-Human Resources Costs

In addition to labor, the project requires resources like software, database setup, utilities, and health insurance. These non-human resources are estimated to cost **$95,000**. When combined with human resources, the **total product development cost is $185,900**.

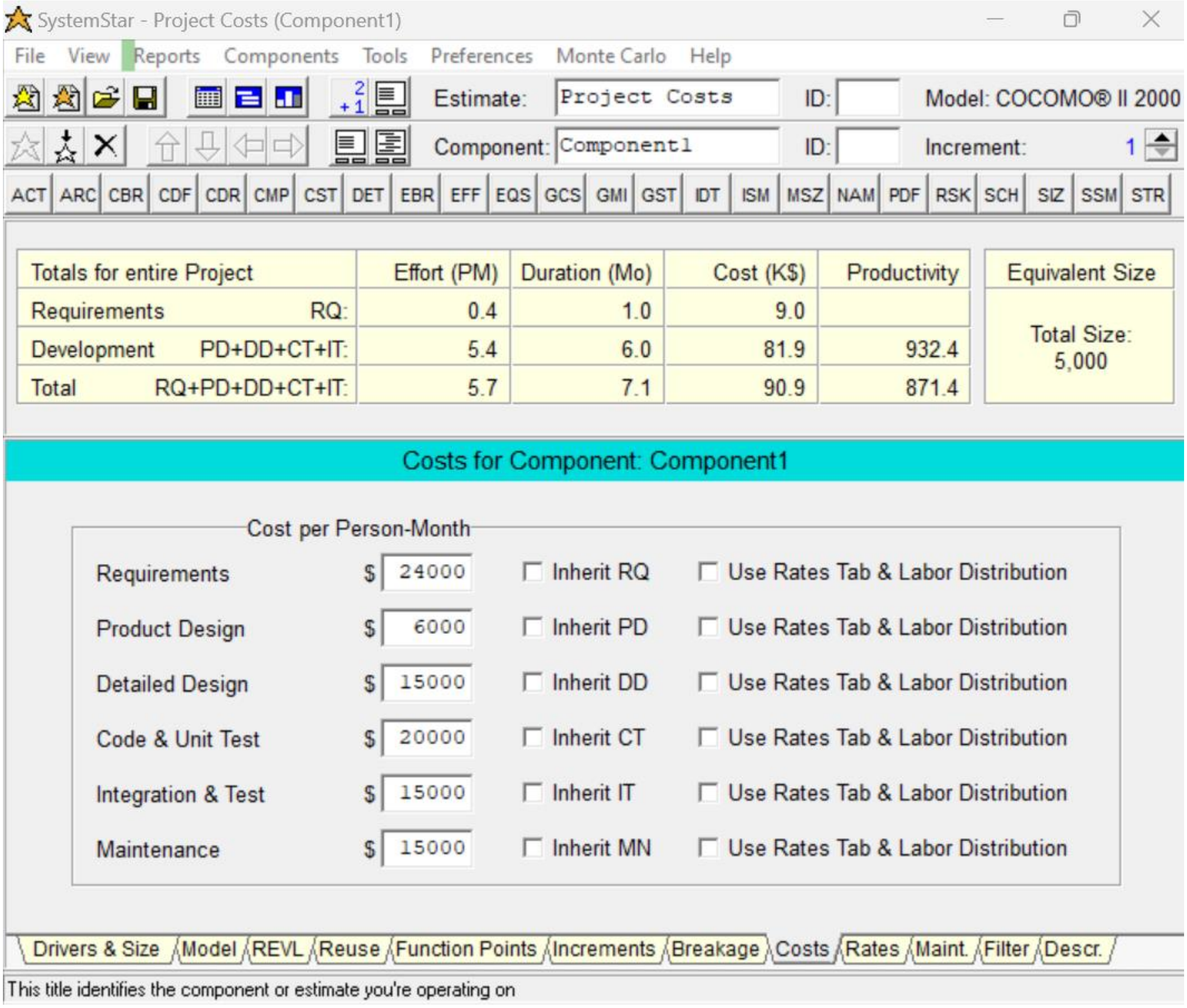
Profit Calculation

To meet business goals, a 100% profit margin is applied to the total cost, effectively doubling it:

Final Cost = 2×185,900=371,800

Final Project Cost Estimate

The final cost of the project, inclusive of all resources and profit, is **$371.8K**. This comprehensive estimate ensures all expenses are covered and aligns with profitability targets, providing a solid financial plan for project completion and client billing.



Estimate: Project Costs ID:

Model: COCOMO® II 2000

Component: Component1 ID:

Increment: 1

Totals for entire Project		Effort (PM)	Duration (Mo)	Cost (K$)	Productivity	Equivalent Size
Requirements	RQ:	0.4	1.0	9.0		Total Size: 5,000
Development	PD+DD+CT+IT:	5.4	6.0	81.9	932.4	
Total	RQ+PD+DD+CT+IT:	5.7	7.1	90.9	871.4	

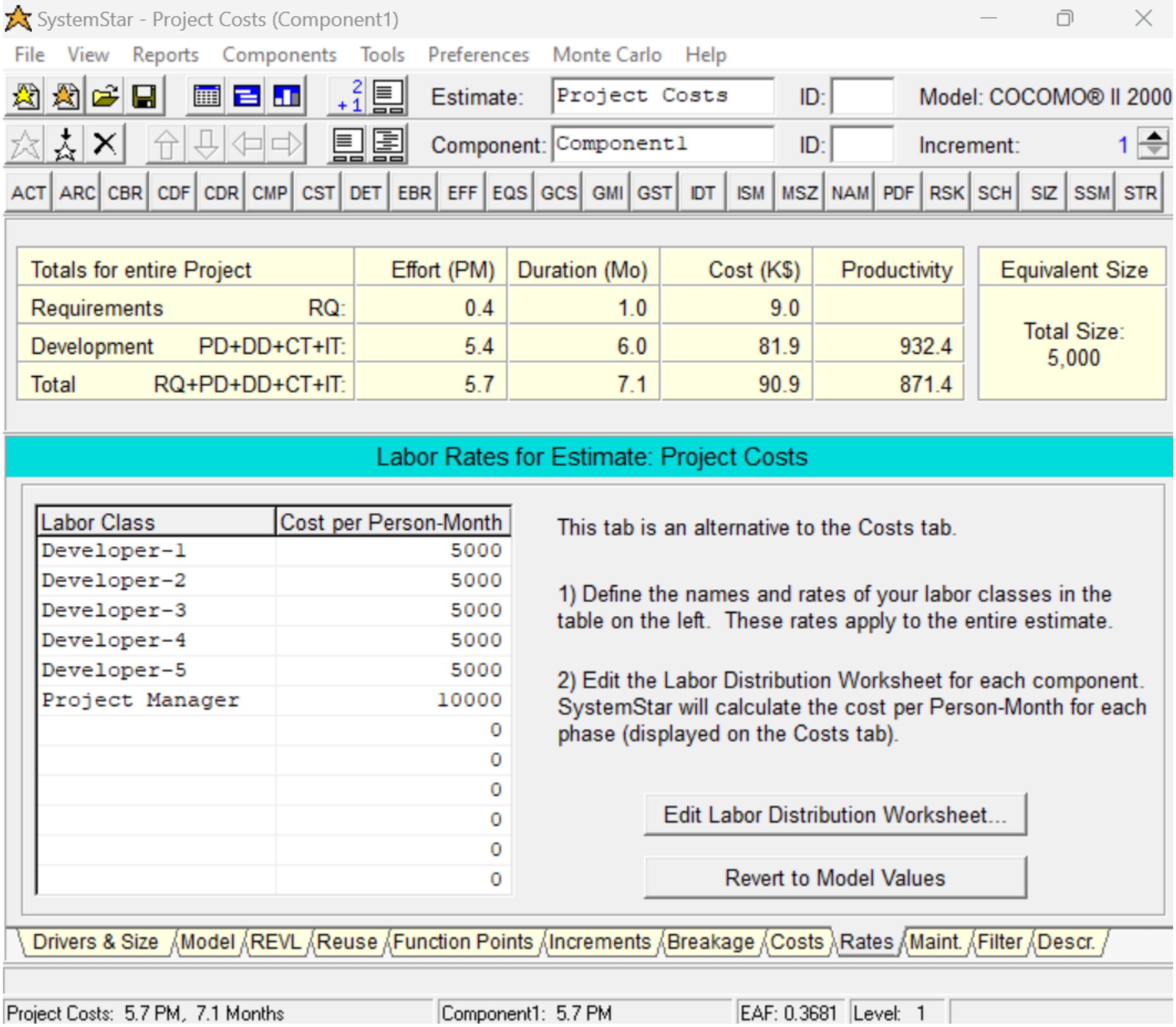
Costs for Component: Component1

Cost per Person-Month

Requirements	$ 24000	□ Inherit RQ	□ Use Rates Tab & Labor Distribution
Product Design	$ 6000	□ Inherit PD	□ Use Rates Tab & Labor Distribution
Detailed Design	$ 15000	□ Inherit DD	□ Use Rates Tab & Labor Distribution
Code & Unit Test	$ 20000	□ Inherit CT	□ Use Rates Tab & Labor Distribution
Integration & Test	$ 15000	□ Inherit IT	□ Use Rates Tab & Labor Distribution
Maintenance	$ 15000	□ Inherit MN	□ Use Rates Tab & Labor Distribution

This title identifies the component or estimate you're operating on

[13]


SystemStar - Project Costs (Component1)
File View Reports Components Tools Preferences Monte Carlo Help
ACT ARC CBR CDF CDR CMP CST DET EBR EFF EQS GCS GMI GST IDT ISM MSZ NAM PDF RSK SCH SIZ SSM STR

Estimate: Project Costs ID:

Model: COCOMO® II 2000

Component: Component1 ID:

Increment: 1

Totals for entire Project	Effort (PM)	Duration (Mo)	Cost (K$)	Productivity	Equivalent Size
Requirements RQ:	0.4	1.0	9.0		Total Size: 5,000
Development PD+DD+CT+IT:	5.4	6.0	81.9	932.4	
Total RQ+PD+DD+CT+IT:	5.7	7.1	90.9	871.4	

Labor Rates for Estimate: Project Costs

Labor Class	Cost per Person-Month
Developer-1	5000
Developer-2	5000
Developer-3	5000
Developer-4	5000
Developer-5	5000
Project Manager	10000
	0
	0
	0
	0
	0
	0

This tab is an alternative to the Costs tab.

1) Define the names and rates of your labor classes in the table on the left. These rates apply to the entire estimate.

2) Edit the Labor Distribution Worksheet for each component. SystemStar will calculate the cost per Person-Month for each phase (displayed on the Costs tab).

Edit Labor Distribution Worksheet...
Revert to Model Values
Drivers & Size Model REVL Reuse Function Points Increments Breakage Costs Rates Maint. Filter Descr.

Project Costs: 5.7 PM, 7.1 Months

Component1: 5.7 PM

EAF: 0.3681

Level: 1

4. Conclusion and Recommendations

1. **Understanding the Variance in Estimates**

   Initial Estimate (3 months): Originally, the project was estimated to take only 3 months. This shorter estimate may have been based on simple metrics or assumptions, like task duration, without factoring in complexities that affect the timeline and costs.

   COCOMO Estimate (7.1 months): The COCOMO model (Constructive Cost Model) is a sophisticated estimation tool used specifically for software projects. It incorporates a detailed analysis of several critical factors that impact the effort, time, and cost needed. Key components in COCOMO's calculations include:

1. **Source Lines of Code (SLOC)**: The number of lines of code expected for the project is a primary factor since larger codebases require more time to develop and test.

2. **Scale Factors**: Five factors (like project size and complexity) that can increase project effort depending on how large or complicated the project is.

3. **Cost Drivers**: Seventeen cost drivers, such as team experience, tool efficiency, and software reliability, provide finer adjustments to the estimate by considering project-specific challenges.

   Because COCOMO considers these detailed factors, its estimates are typically more realistic but often longer, as its better accounts for the complexities and challenges that simpler models might overlook.

2. **Proposed Solution: Two-Phase Development**

   Given the longer timeline indicated by COCOMO, the idea is to split the project into two main phases, rather than developing everything at once. This strategy allows us to get part of the project (the website) up and running earlier while still working towards the full project completion (website and app).

   Here's how each phase is structured:

Phase 1: Website Development

1. **Goal**: Build and launch the website first, which gives clients something to start using sooner.

2. **Rationale**: The team has strong experience in web development, which means they can work efficiently and with high quality on the website. Completing this part early also gives clients an early product to interact with and provide feedback on, which is valuable for refining both the website and the future app.

3. **Benefits**:

   i. Reduces the overall time clients must wait to see project results.

   ii. Creates a foundation where feedback from website users can help improve the Android app's design and functionality.

Phase 2: Android App Development

1. **Goal**: Focus on building the Android app after the website is released.

2. **Training Component**: While the team is working on the website, they will also undergo training in Android development to build their skills in this area. This staggered approach prepares the developers to smoothly transition to app development once the website is complete.

3. **Benefits**:

    i. Ensures the team is ready and skilled for mobile app development.

    ii. Allows adjustments to the app's design based on feedback from the website, leading to a more refined product.

4. Advantages of the Phased Approach

This phased approach has several important advantages:

1. **Faster Delivery of Part of the Project**: Since the website is completed first, clients don't have to wait 7.1 months to see the initial product. They can start using the website earlier, which also creates a positive impression by showing progress.

2. **Optimized Use of Skills and Resources**: By focusing on the website first, the team can immediately leverage their strengths in web development. This targeted use of expertise keeps productivity high and results efficient.

3. **Early Client Feedback**: Launching the website first gives clients a product they can test and use, gathering valuable user feedback. This feedback can guide changes or improvements for the Android app, making it more aligned with user needs when it's eventually launched.

4. **Skill Building for Future Needs**: The Android training ensures that the team has the skills needed for the app phase, enhancing their capabilities and improving the quality of the app development.

Improvements Made (from Assignment 2 to Assignment 3)

Between Assignment 2 and Assignment 3, significant refinements were introduced to enhance the estimation and planning for the Stock Recording System (SRS) project:

1. **Improved Cost Estimation**
   In Assignment 2, cost estimates were primarily based on resource allocation and task duration assumptions, without accounting for broader aspects like overhead, profit, and detailed effort multipliers. With the adoption of the COCOMO II model in Assignment 3, we achieved a more comprehensive cost breakdown, incorporating both operational and personnel costs. This resulted in a precise and dependable cost structure, aiding effective project planning.

2. **Inclusion of Scale and Cost Drivers**
   The original Waterfall plan focused on sequential task breakdowns and resource assignments. In Assignment 3, using COCOMO II introduced essential factors such as team cohesion, product complexity, platform familiarity, and tool use. These scale and cost drivers helped capture qualitative aspects, such as team skills, tool efficiency, and risk levels, providing a well-rounded view of project effort that identified potential challenges overlooked in the initial estimates.

3. **Enhanced Risk and Resource Management**
   Assignment 3 emphasized a more proactive approach to risk and resource management by including scale drivers and effort multipliers that address project volatility, team stability, and complexity. This approach led to better risk mitigation strategies, improving the project's ability to adapt to evolving requirements and technical uncertainties.

4. **Detailed Financial Planning with Overhead and Profit Margins**
   Unlike Assignment 2, which focused mainly on direct expenses, Assignment 3 introduced overhead costs (50%) and a profit margin (100%), providing a comprehensive view of financial needs. This addition ensured all operational and profitability elements were included, enhancing transparency in client billing and improving the overall budget planning.

Impact of COCOMO Analysis on the Project

The application of the COCOMO II model transformed the estimation, planning, and cost structure of the SRS project, with several key impacts:

1. **Realistic Cost and Effort Estimates**
   COCOMO II estimated a total project cost of $371.8K, aligning well with the overall project scope and timeline. This refined allocation of effort across team members enabled precise workload planning, reducing risks of over- or under-committing resources during development phases.

2. **Identification of Key Cost Drivers**
   COCOMO II analysis highlighted critical cost drivers, including Required Reliability (RELY), Platform Volatility (PVOL), and Tool Use (TOOL). For instance, the high platform volatility indicated a need for thorough integration and testing, while strong tool use, such as with Jira and Slack, reflected high productivity and collaboration standards. These insights helped prioritize resources and pinpoint areas needing additional focus.

3. **Comprehensive Understanding of Project Complexity**
   By examining scale drivers like Precedentedness and Team Cohesion, we gained valuable insights into project complexity and team dynamics. This reinforced the importance of structured collaboration, staged learning, and consistent methodology to ensure cohesion and flexibility, ultimately improving project outcomes.

4. **Enhanced Financial Planning**
   Including overhead and profit margins provided a complete perspective on project costs, ensuring client billing was accurate and all operational expenses were covered. The final charge estimate of **$371.8K** accounted for tools, licenses, and contingencies, ensuring project profitability.

Project Shortcomings and Their Impact
While many improvements were made, certain limitations remain that could affect project success if unaddressed:

1. **Limited Android Development Expertise**

   i. *Issue*: Limited experience with Android development was noted in Programmer Capability (PCAP) and Platform Experience (PLEX) ratings.

   ii. *Impact*: This gap may slow development and lead to rework. Mitigation options include additional training, mentorship, or hiring experienced mobile developers.

2. **High Platform Volatility (PVOL)**

   i. *Issue*: Dependency on external APIs and real-time data adds volatility.

   ii. *Impact*: API changes or data unavailability may cause delays and bugs. Strategies like frequent API testing, error handling, and fallback mechanisms can help maintain stability.

3. **Strict Development Schedule (SCED)**

   i. *Issue*: The project’s tight timeline allows minimal flexibility.

   ii. *Impact*: Rushed development cycles could reduce testing time, impacting quality. Focusing on high-priority features and minimum viable functionality can help manage this risk.

4. **Product Complexity (CPLX)**

   i. *Issue*: Nominal complexity ratings suggest integration and security challenges.
   ii. *Impact*: Mismanaged complexity could impact usability and introduce technical debt. Using Waterfall methods like task mapping and staged testing can break complex tasks into manageable parts.

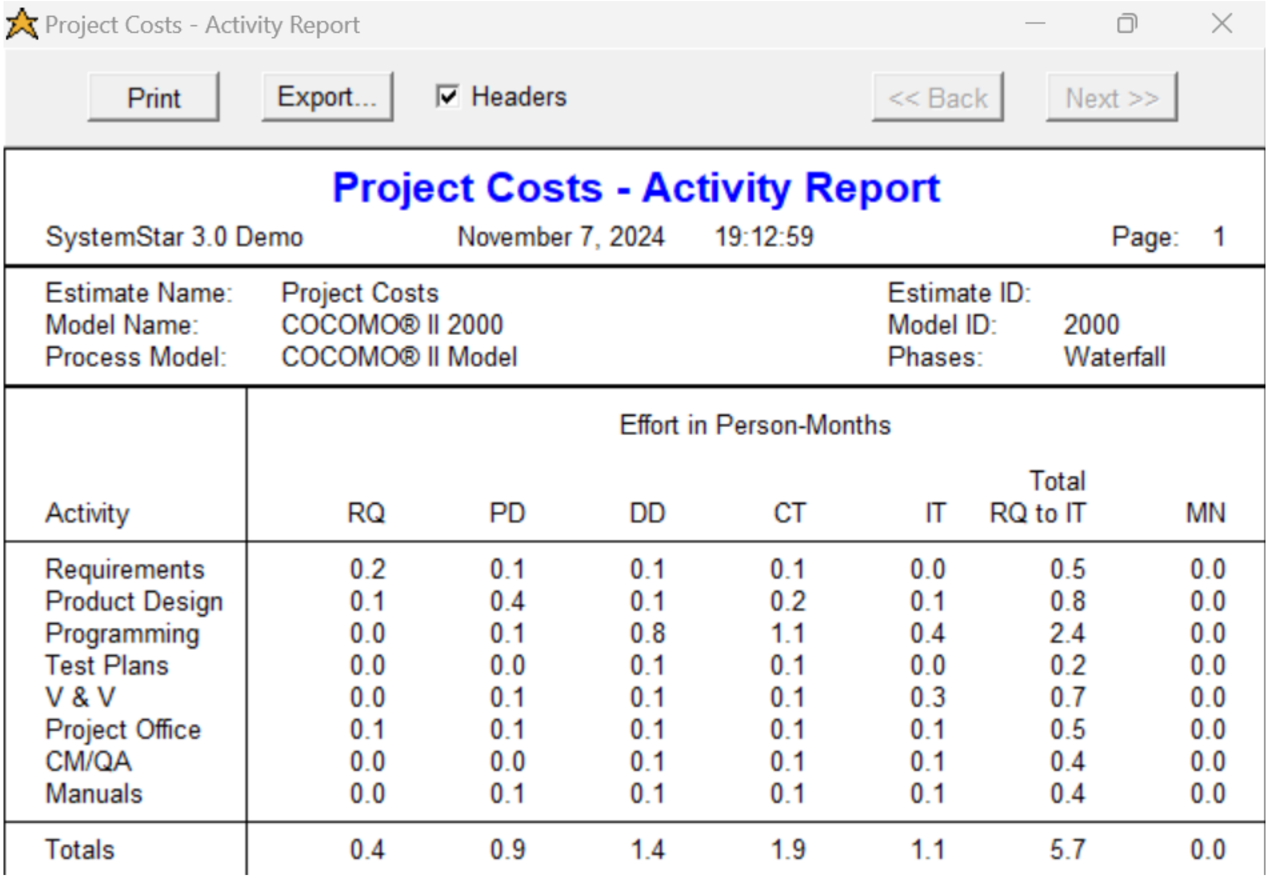
Recap and Recommendations

The COCOMO II analysis revealed additional cost, effort, and risk considerations not addressed in the original Waterfall plan, such as overhead, profit margins, team cohesion, and platform volatility. These insights provided a clearer view of project risks and resources, resulting in a more precise estimation of the project's scope and budget.

Recommendation: With refined cost structures, realistic effort estimates, and enhanced risk management, we recommend continuing the project. However, it's crucial to address skill gaps, manage platform volatility, and focus on core features to meet deadlines and achieve successful project delivery.

Appendices

Additional documents, print screens of COCOMO reports, references.



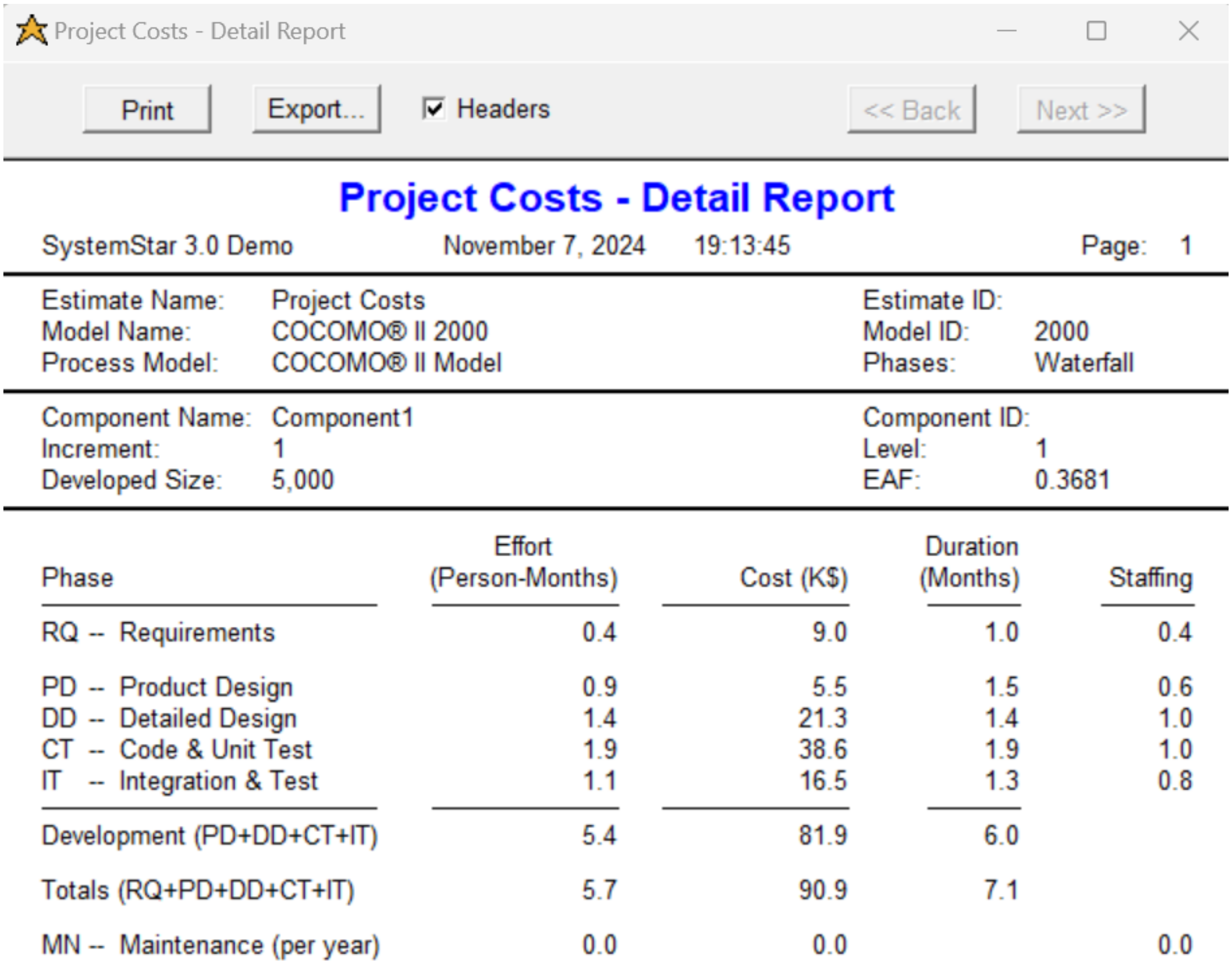
Project Costs - Activity Report

SystemStar 3.0 Demo November 7, 2024 19:12:59 Page: 1

Estimate Name: Project Costs
Model Name: COCOMO® II 2000
Process Model: COCOMO® II Model
Estimate ID:
Model ID: 2000
Phases: Waterfall

Effort in Person-Months

Activity	RQ	PD	DD	CT	IT	Total RQ to IT	MN
Requirements	0.2	0.1	0.1	0.1	0.0	0.5	0.0
Product Design	0.1	0.4	0.1	0.2	0.1	0.8	0.0
Programming	0.0	0.1	0.8	1.1	0.4	2.4	0.0
Test Plans	0.0	0.0	0.1	0.1	0.0	0.2	0.0
V & V	0.0	0.1	0.1	0.1	0.3	0.7	0.0
Project Office	0.1	0.1	0.1	0.1	0.1	0.5	0.0
CM/QA	0.0	0.0	0.1	0.1	0.1	0.4	0.0
Manuals	0.0	0.1	0.1	0.1	0.1	0.4	0.0
Totals	0.4	0.9	1.4	1.9	1.1	5.7	0.0



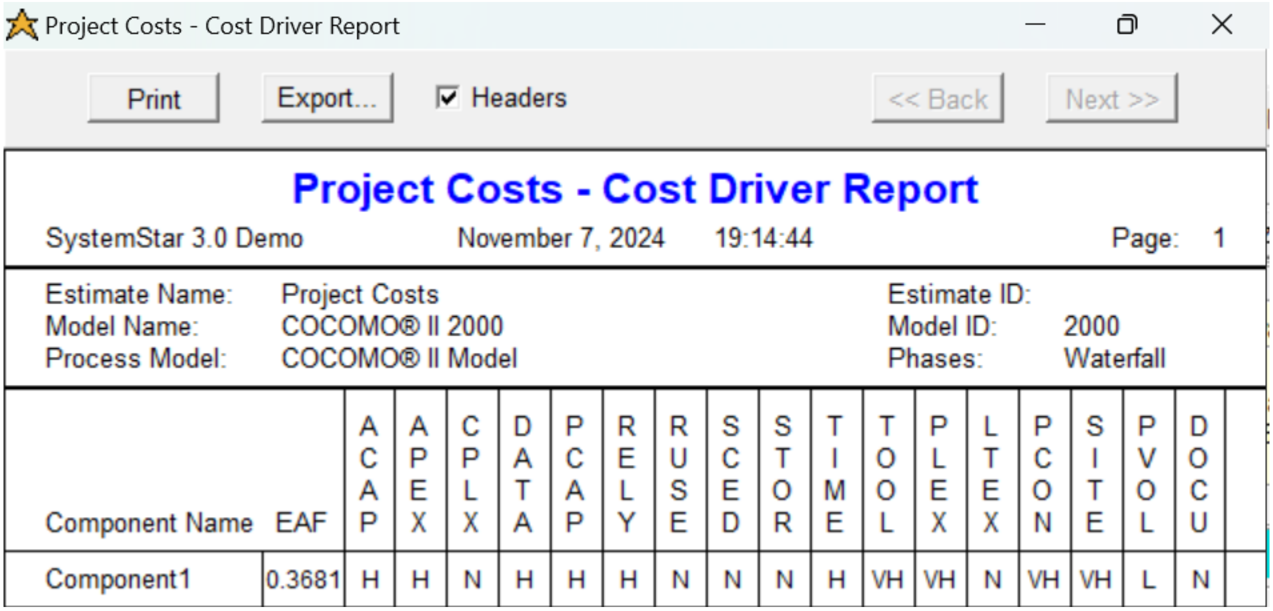
Project Costs - Detail Report

SystemStar 3.0 Demo   November 7, 2024   19:13:45   Page: 1

Estimate Name:	Project Costs	Estimate ID:	
Model Name:	COCOMO® II 2000	Model ID:	2000
Process Model:	COCOMO® II Model	Phases:	Waterfall

Component Name:	Component1	Component ID:	
Increment:	1	Level:	1
Developed Size:	5,000	EAF:	0.3681

Phase	Effort (Person-Months)	Cost (K$)	Duration (Months)	Staffing
RQ -- Requirements	0.4	9.0	1.0	0.4
PD -- Product Design	0.9	5.5	1.5	0.6
DD -- Detailed Design	1.4	21.3	1.4	1.0
CT -- Code & Unit Test	1.9	38.6	1.9	1.0
IT -- Integration & Test	1.1	16.5	1.3	0.8
Development (PD+DD+CT+IT)	5.4	81.9	6.0	
Totals (RQ+PD+DD+CT+IT)	5.7	90.9	7.1	
MN -- Maintenance (per year)	0.0	0.0		0.0



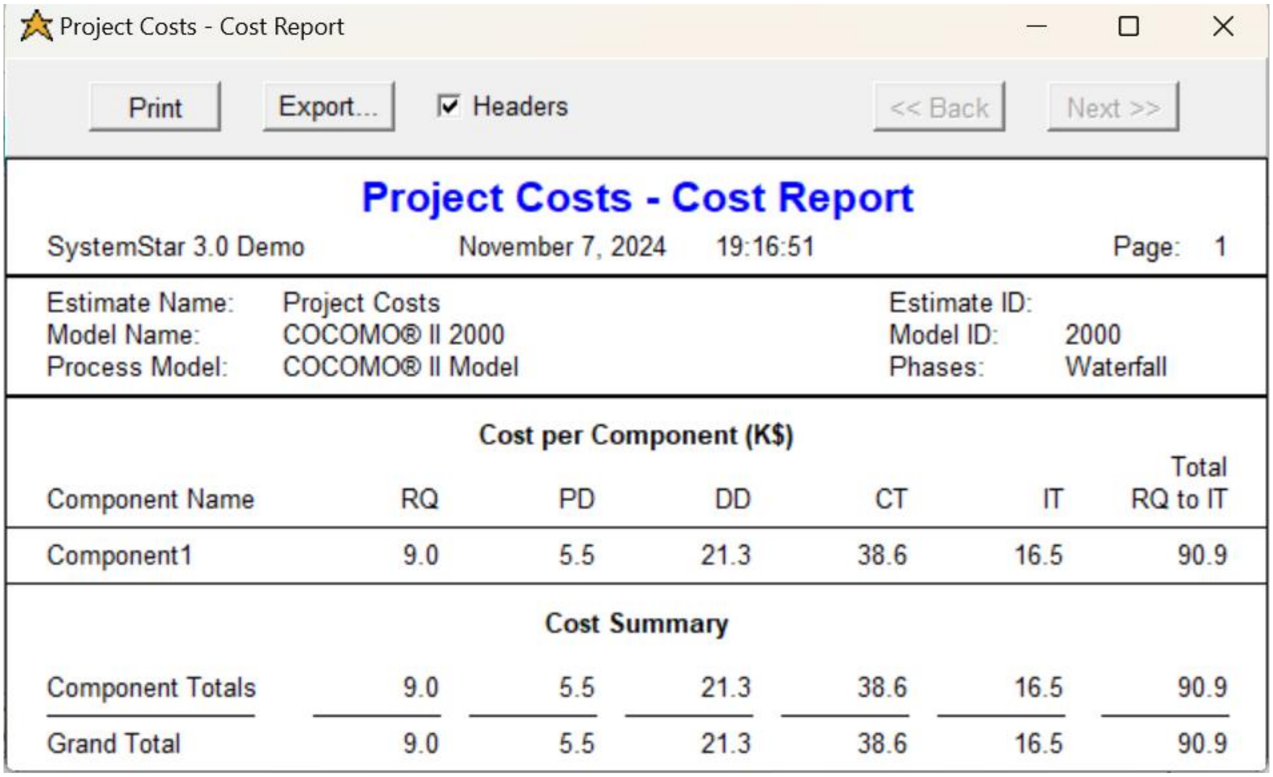
Project Costs - Cost Driver Report

SystemStar 3.0 Demo   November 7, 2024   19:14:44   Page: 1

Estimate Name:	Project Costs	Estimate ID:	
Model Name:	COCOMO® II 2000	Model ID:	2000
Process Model:	COCOMO® II Model	Phases:	Waterfall

Component Name	EAF	ACAP	APEX	CPLX	DATA	PCAP	RELY	RUSE	SCED	STOR	TIME	TOOL	PLEX	LTEX	PCON	SITE	PVOL	DOCU
Component1	0.3681	H	H	N	H	H	H	N	N	N	H	VH	VH	N	VH	VH	L	N

[21]



Project Costs - Cost Report

SystemStar 3.0 Demo   November 7, 2024   19:16:51   Page: 1

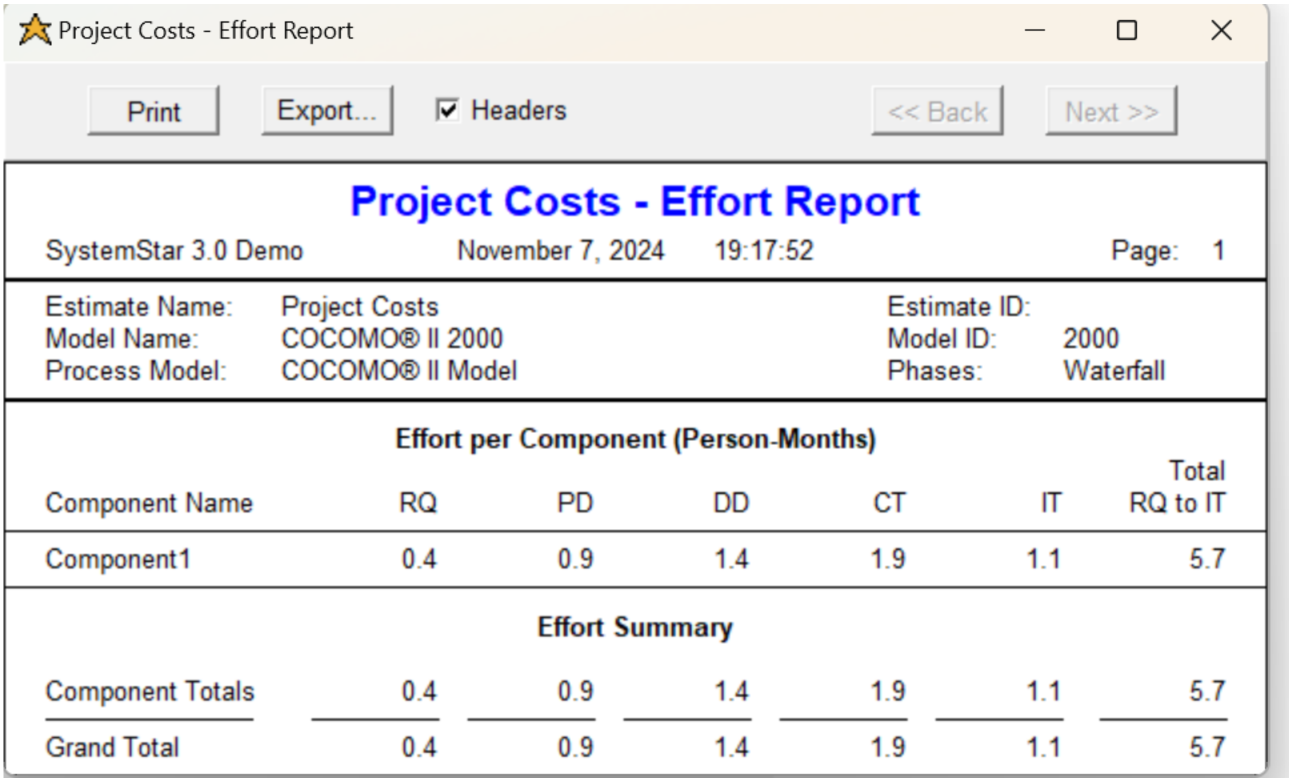
Estimate Name:	Project Costs	Estimate ID:	
Model Name:	COCOMO® II 2000	Model ID:	2000
Process Model:	COCOMO® II Model	Phases:	Waterfall

Cost per Component (K$)

Component Name	RQ	PD	DD	CT	IT	Total RQ to IT
Component1	9.0	5.5	21.3	38.6	16.5	90.9

Cost Summary

Component Totals	9.0	5.5	21.3	38.6	16.5	90.9
Grand Total	9.0	5.5	21.3	38.6	16.5	90.9



Project Costs - Effort Report

SystemStar 3.0 Demo   November 7, 2024   19:17:52   Page: 1

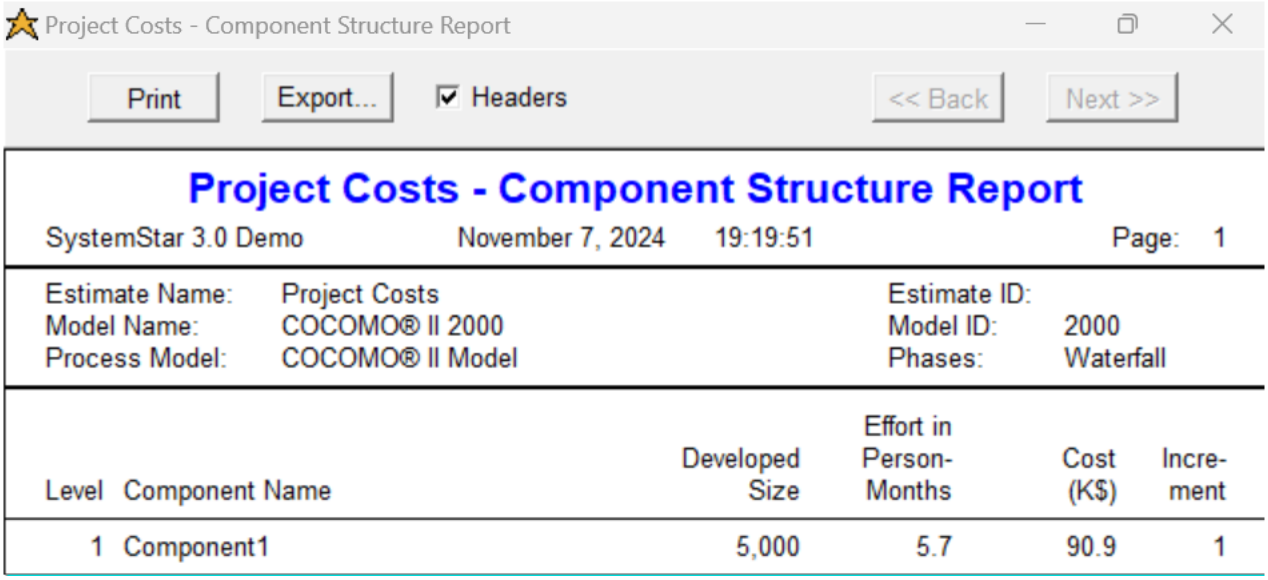
Estimate Name:	Project Costs	Estimate ID:	
Model Name:	COCOMO® II 2000	Model ID:	2000
Process Model:	COCOMO® II Model	Phases:	Waterfall

Effort per Component (Person-Months)

Component Name	RQ	PD	DD	CT	IT	Total RQ to IT
Component1	0.4	0.9	1.4	1.9	1.1	5.7

Effort Summary

Component Totals	0.4	0.9	1.4	1.9	1.1	5.7
Grand Total	0.4	0.9	1.4	1.9	1.1	5.7

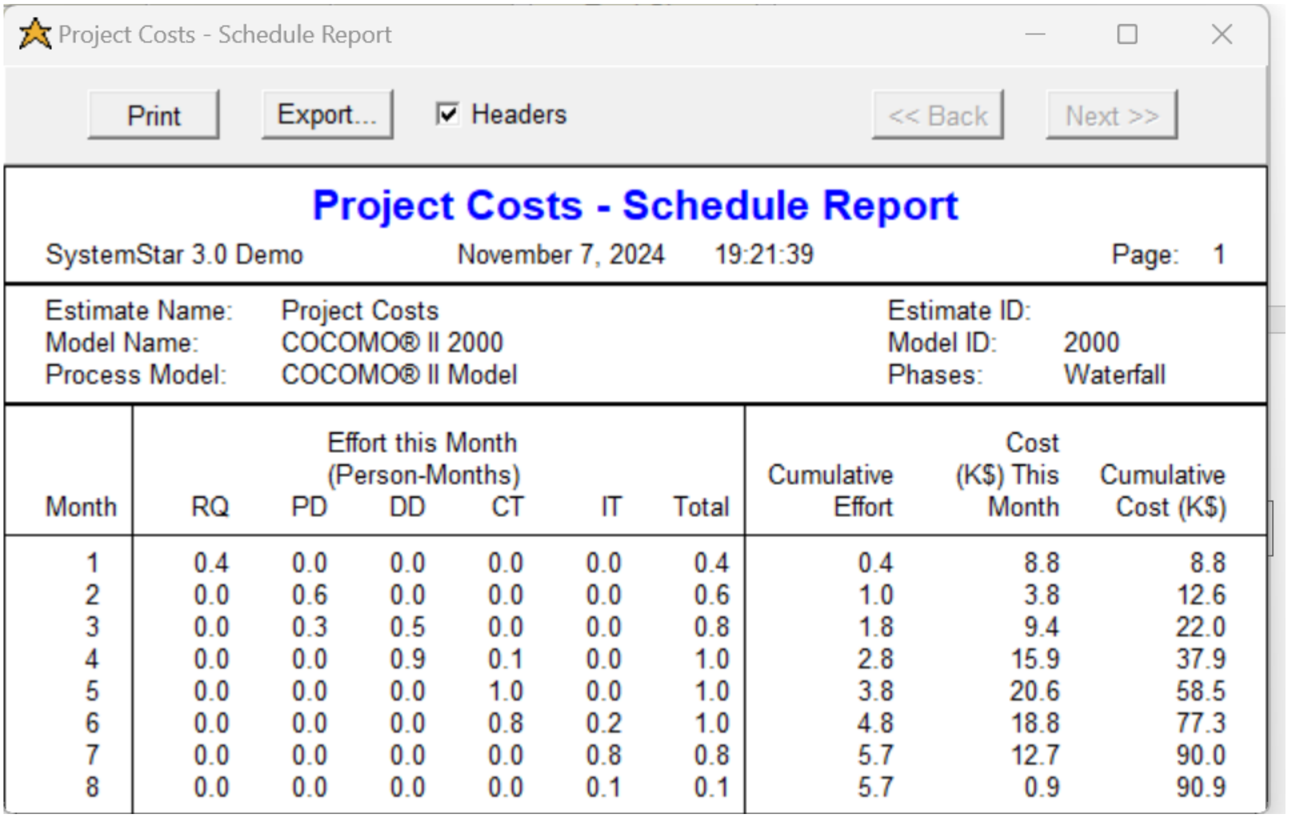


Project Costs - Component Structure Report

SystemStar 3.0 Demo November 7, 2024 19:19:51 Page: 1

Estimate Name:	Project Costs	Estimate ID:	
Model Name:	COCOMO® II 2000	Model ID:	2000
Process Model:	COCOMO® II Model	Phases:	Waterfall

Level	Component Name	Developed Size	Effort in Person-Months	Cost (K$)	Incre-ment
1	Component1	5,000	5.7	90.9	1



Project Costs - Schedule Report

SystemStar 3.0 Demo November 7, 2024 19:21:39 Page: 1

Estimate Name:	Project Costs	Estimate ID:	
Model Name:	COCOMO® II 2000	Model ID:	2000
Process Model:	COCOMO® II Model	Phases:	Waterfall

	Effort this Month (Person-Months)								
Month	RQ	PD	DD	CT	IT	Total	Cumulative Effort	Cost (K$) This Month	Cumulative Cost (K$)
1	0.4	0.0	0.0	0.0	0.0	0.4	0.4	8.8	8.8
2	0.0	0.6	0.0	0.0	0.0	0.6	1.0	3.8	12.6
3	0.0	0.3	0.5	0.0	0.0	0.8	1.8	9.4	22.0
4	0.0	0.0	0.9	0.1	0.0	1.0	2.8	15.9	37.9
5	0.0	0.0	0.0	1.0	0.0	1.0	3.8	20.6	58.5
6	0.0	0.0	0.0	0.8	0.2	1.0	4.8	18.8	77.3
7	0.0	0.0	0.0	0.0	0.8	0.8	5.7	12.7	90.0
8	0.0	0.0	0.0	0.0	0.1	0.1	5.7	0.9	90.9

References

1.Wikipedia

https://en.wikipedia.org/wiki/COCOMO

2.Geeksforgeeks

https://www.geeksforgeeks.org/software-engineering-cocomo-model/