Project Cost Estimator

A Decision Support System for Software Development

By Zayd N. Sukhun and Kevin C. Krefting

Purpose

The Project Cost Estimator (PCE) is a Model-Driven Decision Support System that is used to estimates software development costs based on the inputs required by the Constructive Cost Model (COCOMO). The COCOMO model was developed using data from a large set of projects that TRW, a California consulting firm, had gathered. Software cost estimation is a very important process as it predicts the amount of effort necessary to produce the final software product. PCE aids the user in cost estimation of a project by breaking down the measurable parts of the development process.

The main inputs for a project are placed into the following categories: function points, technical complexity, lines of code, effort mode, effort adjustment factor, and costs. With these inputs, the user may assign values that correspond to the work necessary for the proposed project. The values are placed into the equations provided by COCOMO model and the result is an estimate of the project cost. Using what-if analysis, the user can change the values of the drivers associated with the project in an effort to find a better cost for the client. This will be very beneficial to the customer, and should increase customer satisfaction while controlling costs and better defining the project.

Intended users

This spreadsheet-based DSS caters to multiple target users. Because the business of software development must be flexible, accurate, and timely, the list of people working directly with the client continues to grow. The target users for this DSS include:

- Project Managers
- Sales force
- Programmers who need to efficiently set the project plan

By carefully studying the system results and analyzing the cost drivers associated with the project, programmers and project managers can easily prioritize the project requirements and have more accurate scheduling to reflect the work needing to be done.

Although the specific user cannot be narrowed down into one job class, she must have all of the skills and knowledge necessary to estimate the amount of work it will take to complete a project. A thorough knowledge of programming and software development is a must. Knowledge of non-commented source lines of code (NCLOC) and commented lines of source code (CLOC) is also important for estimating the size of the project. This is because the LOC estimate is an important variable in the COCOMO model. A large mistake in this estimate will cause highly unreliable data to be returned by the system.

Discussion of needs, benefit, and limitations

This system needs valid inputs to make a proper cost estimate and what-if analysis. Because the inputs are so critical in this application, it is recommended that the user be highly competent in the business of software development. The greater the experience of the user with software development, the better the input estimate will be.

Accurate input estimates will result in a more accurate cost estimate. This cost estimate must be as close to the final cost or the result could be an angry customer.

The benefits of this DSS will come in the form of minimizing cost factors for the customer, presenting what-if analysis to the estimator, and performing feasibility studies. These benefits help a software development firm prepare and present a bid to a potential client. By minimizing costs, the development company will have a greater chance of acquiring the client by presenting the lowest price for a product. What-if analysis will help the user try out different scenarios and apply different variables to the potential project. This will allow for a better decision making when it comes time to deciding on a direction to take the project. This system also helps the user decide if the project is feasible by presenting cost and profit factors. These two factors result in a majority of the feasibility decisions that are made with software development.

The limitations of this system are similar to any decision support system. This DSS will not make a decision for the user. Its main purpose is to support the user in decision making. There can be many other factors that play a part of the decision process that this system does not cover. It is important for the user to use proper judgment when using this system. It is a tool that generates information for decision purposes, nothing more.

The models, inputs, and decision variables

The first step in the model-driven DSS is estimating the project size. When studying the two main size estimation tools (lines of code 'LOC' and Function Point Analysis), a problem is found; The COCOMO model strictly uses LOC while the FP analysis provided more capacity to estimate the project based on complexity, and technical user requirements. Simply using LOC means that the user has to roughly estimate the lines of code the program will require. FP analysis, however, is more thorough and model-driven as opposed to the educated guess that is LOC. Therefore, the DSS asks the user to enter an estimate for lines of code used per function point. The product of the function points and the user estimate will result in an LOC number that correctly reflects the user's specific requirements, and will provide the user with an LOC value to be used in the COCOMO equation.

The next task is to determine the mode that the project falls under. A simple project would fall under the Organic mode, while a Semi-detached mode reflects a more complex project with a tighter schedule, and finally an Embedded mode reflects a project that has a tight set of constraints and very rigid specifications. The type of mode the project falls under will be used to determine two factors (a and b) used in the COCOMO equation. Next, the user will estimate the rating for four types of cost drivers. Cost drivers are the assumptions made about the program that help determine the effort adjustment factor (EAF) used in the COCOMO equation. The next step is for the user to rate the cost drivers associated with the project. The four cost driver categories are *PRODUCT, COMPUTER, PERSONNEL,* and *PROJECT*. Product reflects the software's required reliability, database size and average complexity. Computer driver reflects the time, volatility, and other constraints. Personnel driver reflects the level of knowledge a programmer requires to work on the project, the programmer experience, and other labor oriented drivers. Finally, the Project driver estimates the cost of the extra tools required, the development schedule, and the use of modern programming practices.

After the COCOMO equation calculates the person-months required to complete the project, the user enters the estimated cost per person month and the system then calculates the estimated, pre-profit cost of the project. Finally, the user enters the markup percentage to give the final, profit-inclusive cost of the project.

User Inputs:

Function Points – user must input the count of function points and the complexity of those function points under each of the following categories.

- External Inputs
- External Outputs
- External Inquiries
- External Files
- Internal Files

Technical Complexity – user must input a value between 0 and 5 for each of the following items. 0 means the component has no influence on the system, and 5 means the component is essential to the system.

- Reliable back-up and recovery
- Data communications
- Distributed functions
- Performance
- Heavily used configuration
- Online data entry
- Operational ease
- Online update
- Complex interface
- Complex processing
- Reusability
- Installation ease
- Multiple sits
- Facilitate change

Lines of Code

- Lines of code per function point

Effort Mode – the user must select one of the following values to describe the project.

- Organic
- Semi-Detached
- Embedded

Effort Adjustment Factor – user must select very low, low, nominal, high, very high, or extra high for each of the following cost drivers.

- Product
  - Required software reliability

    - Database size
    - Product complexity
- Computer
    - Execution time constraint
    - Main storage constraint
    - Virtual machine volatility
    - Computer turnaround time
- Personnel
    - Analyst capability
    - Application experience
    - Programmer capability
    - Virtual machine experience
    - Language experience
- Project
    - Modern programming practices
    - Software tools
    - Development schedule

Costs

- Cost per Person Month
- Percent Profit

Calculated Values:

Function Points

- Unadjusted Function Point Count (UFC)
    - (colEInputs_S*celEInputs_S)+(colEInputs_A*celEInputs_A)+(colEInputs_C*celEInputs_C)+(colEO_S*celEO_S)+(colEO_A*celEO_A)+(colEO_C*celEO_C)+(colEInquiries_S*celEInquiries_S)+(colEInquiries_A*celEInquiries_A)+(colEInquiries_C*celEInquiries_C)+(colEF_S*celEF_S)+(colEF_A*celEF_A)+(colEF_C*celEF_C)+(colIF_S*celIF_S)+(colIF_A*celIF_A)+(colIF_C*celIF_C)

Technical Complexity

- Technical Complexity Factor (TFC)
    - 0.65+(0.01*SUM(R4:AE4))

Lines of Code

- Function Points
    - colUFC*colTCF
- Lines of Code
    - colFPs*colLOCFP

Effort Adjustment Factor

- Effort Adjustment Factor (EAF)
    - HLOOKUP(colRELY,tblEAF,3,FALSE)*HLOOKUP(colDATA,tblEAF,4,FALSE)*HLOOKUP(colCPLX,tblEAF,5,FALSE)*HLOOKUP(colTIME,tblEAF,7,FALSE)*HLOOKUP(colSTOR,tblEAF,8,FALSE)*HLOOKUP(colVIRT,tblEAF,9,FALSE)*HLOOKUP(

colTURN,tblEAF,10,FALSE)*HLOOKUP(colACAP,tblEAF,12,FALSE)*HLOOKUP(colAEXP,tblEAF,13,FALSE)*HLOOKUP(colPCAP,tblEAF,14,FALSE)*HLOOKUP(colVEXP,tblEAF,15,FALSE)*HLOOKUP(colLEXP,tblEAF,16,FALSE)*HLOOKUP(colMODP,tblEAF,18,FALSE)*HLOOKUP(colTOOL,tblEAF,19,FALSE)*HLOOKUP(colSCED,tblEAF,20,FALSE)

Final Price

- Effort in Person Months
  - colModeA*(colLOC/1000)^colModeB*colEAF
- Final Price
  - colEffortInPersonMonths*colCostPerPersonMonth*(1+colPercentProfit)

Column Names:

Column Title	Column Name
Name	colName
External Inputs (Simple)	colEInputs_S
External Inputs (Average)	colEInputs_A
External Inputs (Complex)	colEInputs_C
External Outputs (Simple)	colEO_S
External Outputs (Average)	colEO_A
External Outputs (Complex)	colEO_C
External Inquiries (Simple)	colEInquiries_S
External Inquiries (Average)	colEInquiries_A
External Inquiries (Complex)	colEInquiries_C
External Files (Simple)	colEF_S
External Files (Average)	colEF_A
External Files (Complex)	colEF_C
Internal Files (Simple)	colIF_S
Internal Files (Average)	colIF_A
Internal Files (Complex)	colIF_C
UFC	colUFC
F1	colF1
F2	colF2
F3	colF3
F4	colF4
F5	colF5
F6	colF6
F7	colF7
F8	colF8
F9	colF9
F10	colF10
F11	colF11
F12	colF12
F13	colF13
F14	colF14
TCF	colTCF
Function Points	colFPs

LOC per PF Estimate	colLOCFP
Lines of Code	colLOC
Mode	colMode
a	colModeA
b	colModeB
Product (RELY)	colRELY
Product (DATA)	colDATA
Product (CPLX)	colCPLX
Computer (TIME)	colTIME
Computer (STOR)	colSTOR
Computer (VIRT)	colVIRT
Computer (TURN)	colTURN
Personnel (ACAP)	colACAP
Personnel (AEXP)	colAEXP
Personnel (PCAP)	colPCAP
Personnel (VEXP)	colVEXP
Personnel (LEXP)	colLEXP
Project (MODP)	colMODP
Project (TOOL)	colTOOL
Project (SCED)	colSCED
EAF	colEAF
Effort in Person Months	colEffortInPersonMonths
Cost per Person Month	colCostPerPersonMonth
Percent Profit	colPercentProfit
Final Price	colFinalPrice

Data Tables:

Effort Mode Data Table

Mode	**a**	**b**
Organic	3.2	1.05
Semi-Detached	3	1.12
Embedded	2.8	1.2

Effort Adjustment Factor Data Table

Cost Drivers	**Description**	**Very Low**	**Low**	**Nominal**	**High**	**Very High**	**Extra High**
Product							
RELY	Required software reliability	0.75	0.88	1	1.15	1.4	1
DATA	Database size	1	0.94	1	1.08	1.16	1
CPLX	Product complexity	0.7	0.85	1	1.15	1.3	1.65
Computer							
TIME	Execution time constraint	1	1	1	1.11	1.3	1.66
STOR	Main storage constraint	1	1	1	1.06	1.21	1.56
VIRT	Virtual machine volatility	1	0.87	1	1.15	1.3	1
TURN	Computer turnaround time	1	0.87	1	1.07	1.15	1
Personnel							
ACAP	Analyst capability	1.46	1.19	1	0.86	0.71	1
AEXP	Applications experience	1.29	1.13	1	0.91	0.82	1

PCAP	Programmer capability	1.42	1.17	1	0.86	0.7	1
VEXP	Virtual machine experience	1.21	1.1	1	0.9	1	1
LEXP	Language experience	1.14	1.07	1	0.95	1	1
Project							
MODP	Modern programming practices	1.24	1.1	1	0.91	0.82	1
TOOL	Software Tools	1.24	1.1	1	0.91	0.83	1
SCED	Development Schedule	1.23	1.08	1	1.04	1.1	1

Function Point Weighting Factor Data Table

Item	**Simple**	**Average**	**Complex**
External Inputs	3.00	4.00	6.00
External Outputs	4.00	5.00	7.00
External Inquiries	3.00	4.00	6.00
External Files	7.00	10.00	15.00
Internal Files	5.00	7.00	10.00

User information / training issues

The Project Cost Estimator has been designed to be simple in navigation. It has been documented in a way that allows the user to view answers to the questions as they may arise. Each data entry page consists of a description that explains the use of the information for the page. Each label on the page has rollover text that gives a description of the item in question.

Because of the built-in documentation for the system, user training is not expected to be an issue. With this in mind, it is important to realize that the typical user for this system is expected to be knowledgeable in software development practices. Many of the terms used in the DSS may be misunderstood by a person without a development background. To reduce potential confusion and frustration, it is recommended that the official use of this software be restricted to I.T. professionals with basic software experience.

Design and development of the Model-Driven DSS

The design process for the Project Cost Estimator began with the idea of simplicity. It was developed for easy data entry and even easier changes to that data. In order to make the user interface simple to use, the navigation options are many. First of all there is a simple next button at the bottom of every page that aids the user in navigating through the system from start to finish. In the case that the user decides to make a change on a specific screen, she simply clicks on the specific tab and adjusts the inputs. What-if analysis capabilities are increased with the introduction of this simplifying feature.

The development of this spreadsheet-based DSS was based on the idea of separating the data and the user interface. The goal was to have all the data and computations hosted in Excel's spreadsheets and all user interface displays and inputs presented in forms. With the help of VBA, the connection between the spreadsheets and forms was completed. This allows for easy modification to the system to meet future needs.

The “database” table holds the entire user-entered data as well as the calculations necessary to perform system functions. This spreadsheet is color coded for simple comprehension. Each section is titled with a yellow background. All user entered values contain a title with a white background. Finally, each equation-based value is noted with a label that has a blue background. These simple notations will help future developers understand the different parts that make up the data side of the system.

All tables within the system have their own spreadsheet. This makes it easy to find the data that is being used for calculations or look-up functions required by the U.I. Combo box values are also provided on a separate spreadsheet. Each set of combo box data is given a title to simplify the assignment of the data to the combo box.

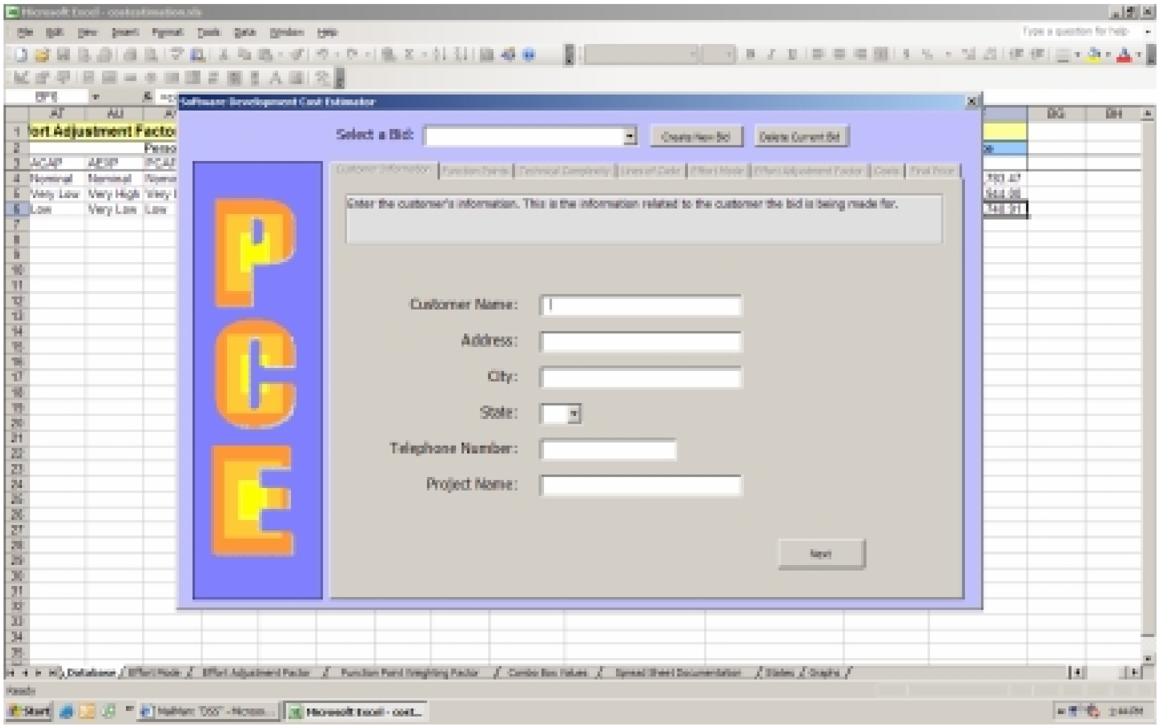
The structure that was used to design and develop this system helped keep the organization of the project at an optimum point. This made it easier for pieces of the system to be split up amongst developers and decrease the development time. This organization will also increase the speed of future changes to the system.

Maintaining the Decision Support System

This Decision Support System has been developed to be virtually maintenance free. Other than design changes or additions, the system’s user interface provides the means of adding and removing projects from the system. All other data values are fixed by the use of the COCOMO model. It was the goal of the development team that this system be maintenance free for the user, and this goal has been met.

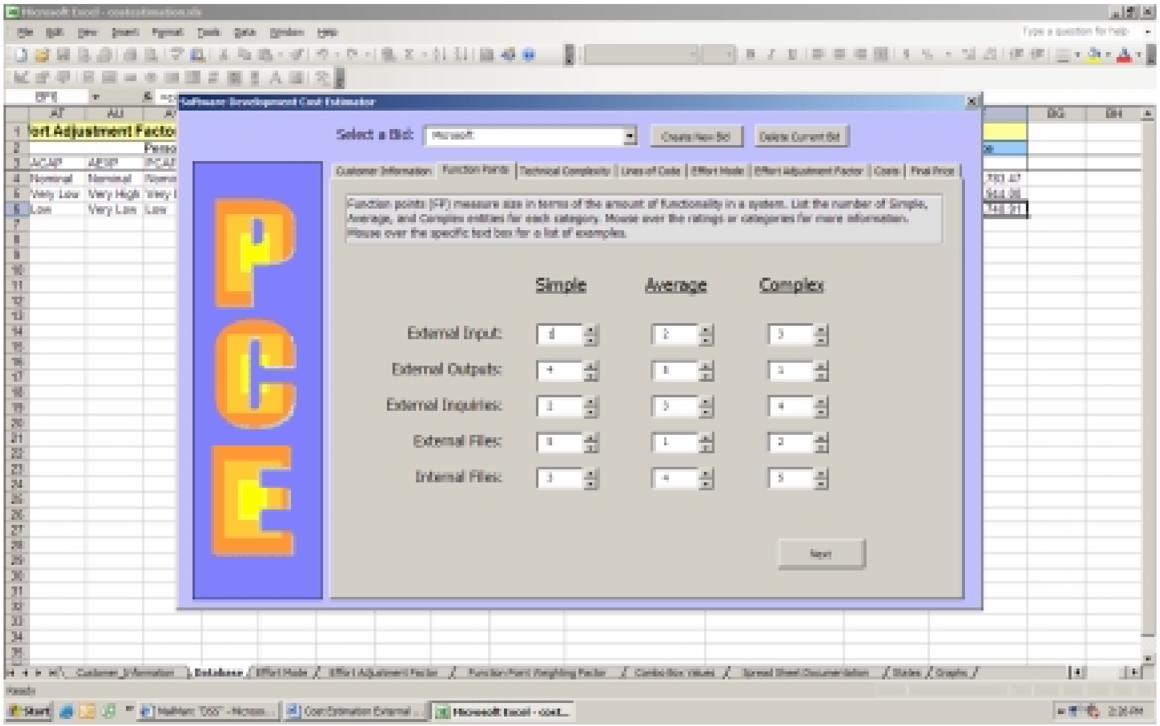
Screen Shots and Walk-Through

The following screen is used to input the customer’s contact data. It permits the user to print out a bid sheet that is personalized for the client.



Enter the customer's information. This is the information related to the customer the bid is being made for.

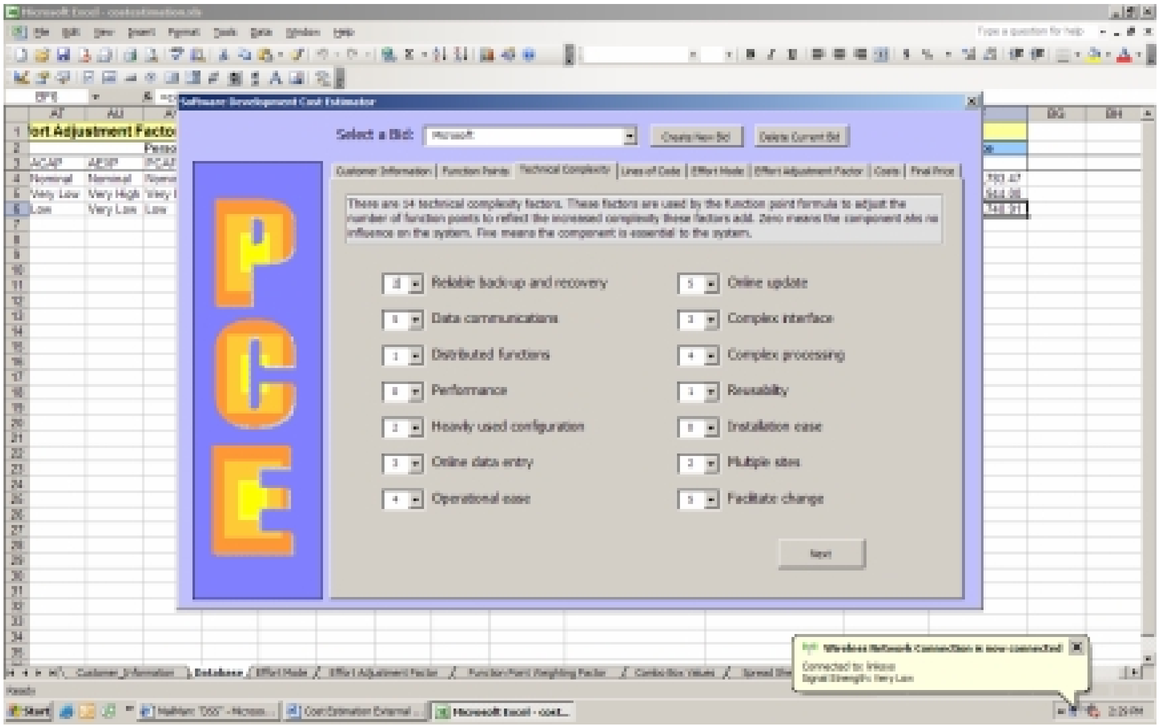
This screen allows the user to enter function points as needed to complete the COCOMO equation. This page, as well as all others, can be changed at any time by using the arrow buttons to the right of every text box. This will help during what-if analysis.



Function points (FP) measure size in terms of the amount of functionality in a system. List the number of Simple, Average, and Complex entities for each category. Mouse over the ratings or categories for more information. Mouse over the specific text box for a list of examples.

	Simple	Average	Complex
External Input:	1	2	3
External Outputs:	4	5	1
External Inquiries:	2	3	4
External Files:	5	1	2
Internal Files:	3	4	5

The Technical Complexity screen uses drop down boxes to choose a value between 0 and 5. This value represents the importance of each factor.

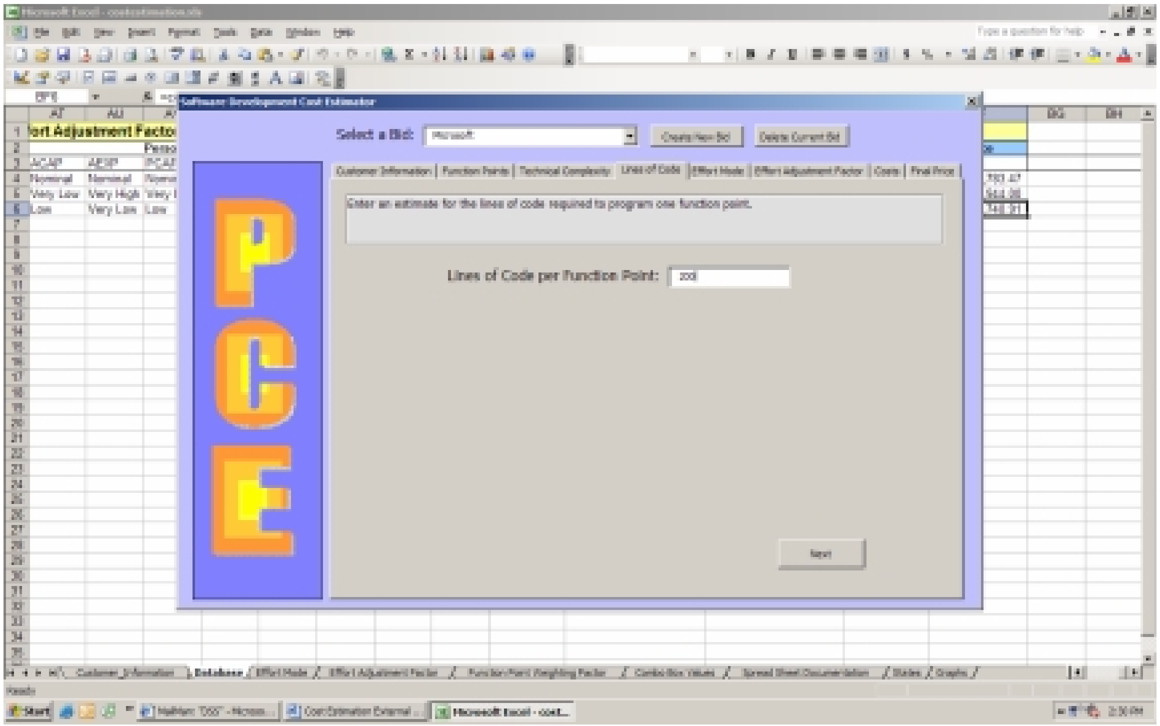


Select a Bid: Microsoft

There are 14 technical complexity factors. These factors are used by the function point formula to adjust the number of function points to reflect the increased complexity these factors add. Zero means the component has no influence on the system. Five means the component is essential to the system.

3 Reliable back-up and recovery
0 Data communications
1 Distributed functions
0 Performance
2 Heavily used configuration
3 Online data entry
4 Operational ease
5 Online update
3 Complex interface
4 Complex processing
1 Reusability
0 Installation ease
2 Multiple sites
3 Facilitate change

The Lines of Code screen allows the user to input the estimated value for lines of code per function point. This is a very important variable used in the COCOMO equation.



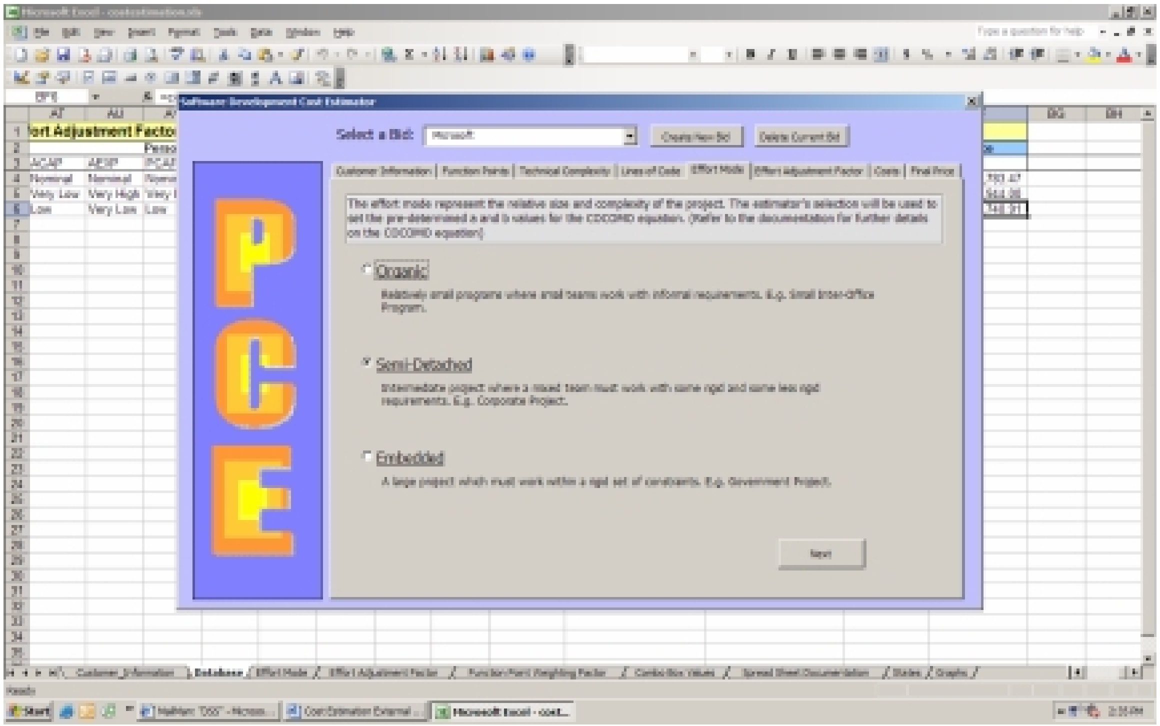
Software Development Cost Estimator

Select a Bid: Microsoft

Enter an estimate for the lines of code required to program one function point.

Lines of Code per Function Point: 200

The Effort Mode screen allows the user to choose between Organic, Semi-Detached, and Embedded project complexities. This allows the model to estimate the amount of effort necessary to complete the task.



The effort mode represent the relative size and complexity of the project. The estimator's selection will be used to set the pre-determined a and b values for the COCOMO equation. (Refer to the documentation for further details on the COCOMO equation)

○ Organic

Relatively small programs where small teams work with informal requirements. E.g. Small Inter-Office Program.

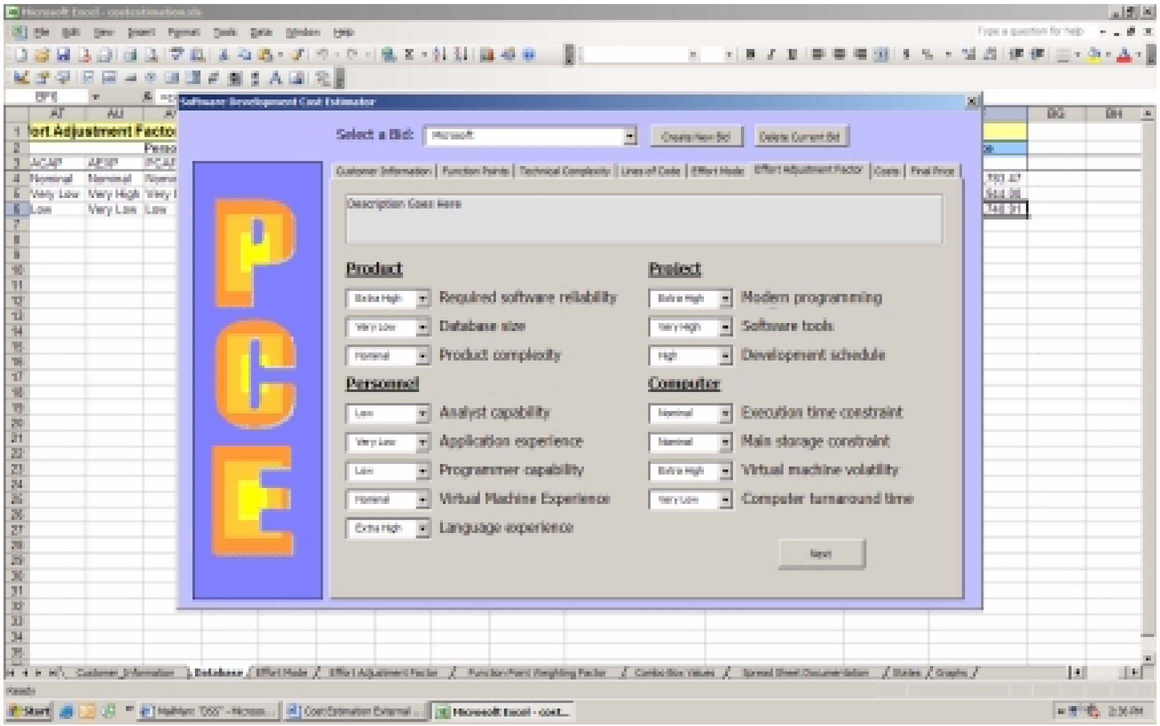
● Semi-Detached

Intermediate project where a mixed team must work with some rigid and some less rigid requirements. E.g. Corporate Project.

○ Embedded

A large project which must work within a rigid set of constraints. E.g. Government Project.

The Effort Adjustment Factor page plays the role of collecting data that displays the importance of different areas of development. Each one of the fifteen pieces help the COCOMO model compute the final effort adjustment factor. The inputs are controlled by drop-down menus that contain the values very low, low, nominal, high, very high, and extra high.



Select a Bid: Microsoft

Description Goes Here

Product

- Extra High — Required software reliability
- Very Low — Database size
- Nominal — Product complexity

Personnel

- Low — Analyst capability
- Very Low — Application experience
- Low — Programmer capability
- Nominal — Virtual Machine Experience
- Extra High — Language experience

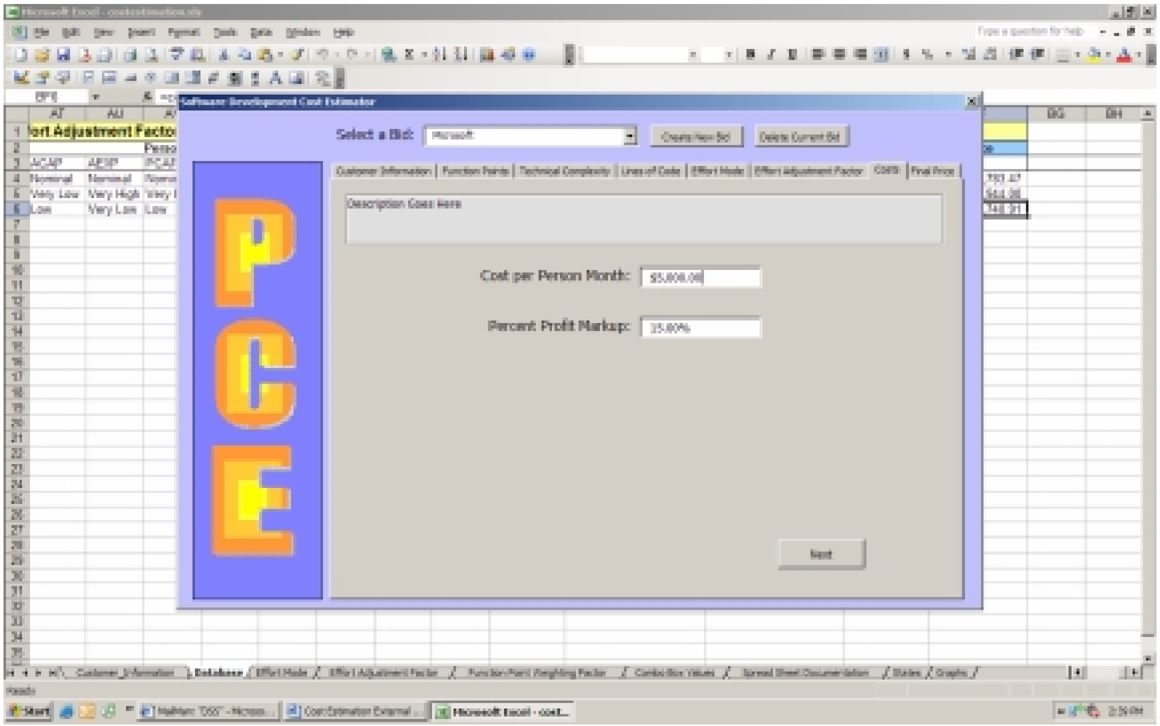
Project

- Extra High — Modern programming
- Very High — Software tools
- High — Development schedule

Computer

- Nominal — Execution time constraint
- Nominal — Main storage constraint
- Extra High — Virtual machine volatility
- Very Low — Computer turnaround time

The Costs screen allows the user to enter the cost per person month estimate and the percent profit margin. Each variable has a direct impact on the final price that is shown in the next screen.



Software Development Cost Estimator

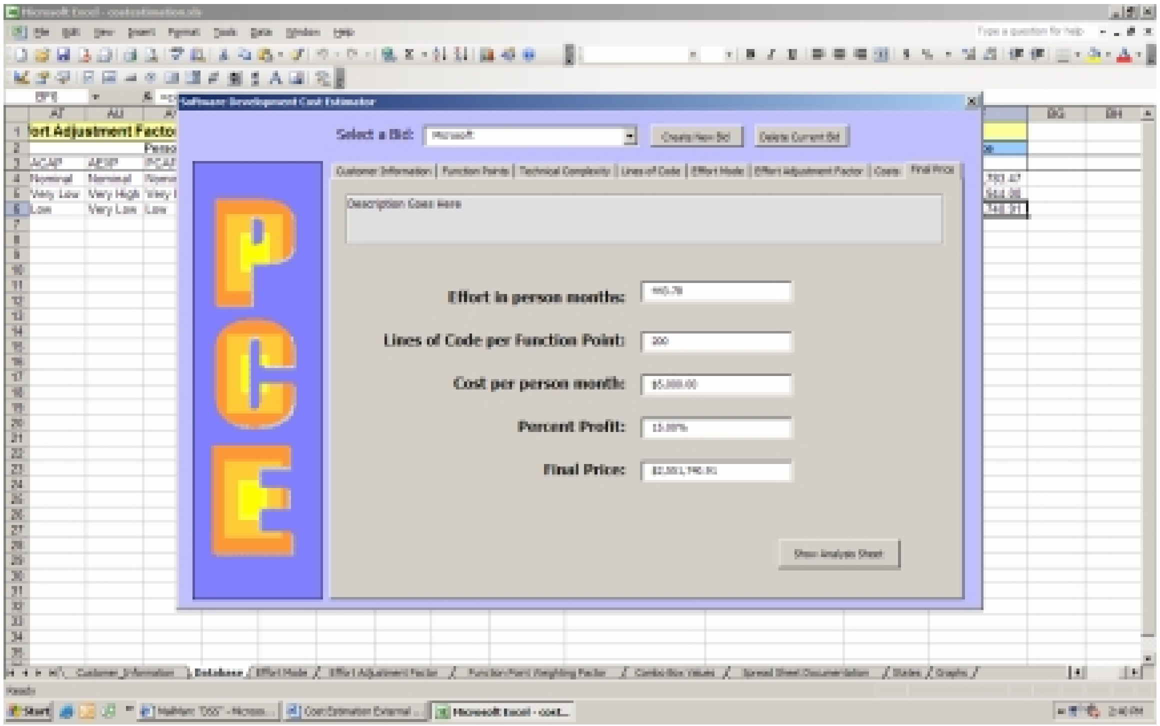
Select a Bid: Microsoft

Description Goes Here

Cost per Person Month: $5,000.00

Percent Profit Markup: 15.00%

The final price screen is used for viewing values that are computed by the system. It also provides buttons that let the user view the analysis sheet for decision support as well as the bid sheet to send to the customer.



Select a Bid: Microsoft

Description Goes Here

Effort in person months: 443.78

Lines of Code per Function Point: 200

Cost per person month: $5,000.00

Percent Profit: 15.00%

Final Price: $2,551,790.91