

# USC COCOMOII

## Reference Manual

University of Southern California

This manual is compatible with USC-COCOMOII.1999 version 0.

### **Copyright Notice**

This document is copyrighted, and all rights are reserved by University of Southern California. Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and full citation on the first page. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, or to redistribute to lists requires prior specific permission and/or fee.

Copyright © 1995 - 1999 USC

All rights reserved.

### **Warranty**

This manual is provided “as is” without warranty of any kind, either express or implied, including, but not limited to the implied warranties of merchantability and fitness for a particular purpose. Moreover, USC reserves the right to revise this manual and to make changes periodically without obligation to notify any person or organization of such revision or changes.

### **Trademark Acknowledgment**

USC has made every effort to supply trademark information about company names, products, and services mentioned in this document. Trademarks indicated below were derived from various sources.

UNIX is a registered trademark of AT&T Bell Laboratories Motif is a trademark of Open Software Foundation, Inc. Sun Microsystem and Sun Workstation are registered trademarks, and OpenWindows, Sun-3, Sun-4, and SPARCstation are trademarks, of Sun Microsystems, Inc.

X Window System is a trademark of the Massachusetts Institute of Technology.

MS Windows95 is a trademark of Microsoft Corporation

### **Acknowledgments**

COCOMO81,Version 1.0:

Principal Investigator - Dr. Ellis Horowitz  
Student Designers, Testers and Programmers - Alfredo Arcilla, Joyce Balimbin, Gina Gaborno, Larry Klein, Robert Kosai, Deseree Moon, Jason Pan, Thomas Quayle, Isaiah Simmons, Scott Zechiel

COCOMO 81, Version 1.1:

Principal Investigator - Dr. Ellis Horowitz  
Student Designers, Testers and Programmers - Ing-Jer Huang

COCOMO 81, Version 10.0:

Principal Investigator - Dr. Ellis Horowitz  
Student Designers, Testers and Programmers - M. Susan Lane, Ping Luo, Lorna Zorman

COCOMO 2.0, Version 2.0:

Principal Investigator - Dr. Ellis Horowitz  
Student Designers, Testers and Programmers - Wiryadi Adidharma, Sen-Ho Chang, Shu-fen Cheng, Yu-Chuan Lin, Steve K. Luk, Shawne Robinson, Tuan Ton

COCOMO 2.0 Version 2.0.5:

Principal Investigator - Dr. Ellis Horowitz  
Student Designers, Testers and Programmers - Thomas Majchrowski, Suppachai Sangtongkhamsuk, Lloyd Manglapus

COCOMO II.1997.2

Principal Investigator - Dr. Ellis Horowitz  
Student Designers, Testers and Programmers: Jongmoon Baik, Jungwon Park, James Chou, Dana Flora-Adams, Sang Hyun, and Eunsook Bang

COCOMO II.1998.0

Principal Investigator - Dr. Ellis Horowitz  
Student Designers, Testers and Programmers: Jongmoon Baik

COCOMO II.1999.0

Principal Investigator - Dr. Ellis Horowitz  
Student Designers, Testers and Programmers: Jongmoon Baik, Cyrus Fakharzadeh, Marwan Abi-Antoun

**Note** - This manual is sufficient for the X Windows and MS Windows 95/98/NT versions of USC-COCOMOII.1999 programs. Though there are some user interface differences between these systems, the core component of all versions is identical.

Some of the material used in this manual has been taken from Software Engineering Economics, by Barry Boehm, Prentice-Hall, with permission of the author.

Chapter 1: Introduction .....	3
1.1 What is COCOMO?.....	3
1.1.1 Effort Estimation Equation .....	4
1.1.2 Schedule Estimation Equation .....	5
1.1.3 Scale Factors .....	5
1.1.4 Sizing Methods .....	6
1.1.5 FP: Counting with Unadjusted Function Points .....	9
1.1.6 AAF: Adaptation Adjustment Factors .....	10
1.1.7 Effort Multipliers .....	10
1.2 Navigating COCOMO .....	12
1.3 Begin Using COCOMO .....	21
1.4 Running COCOMO: Windows/NT and UNIX .....	22
Chapter 2: File Menu .....	23
2.1 New .....	24
2.2 Load Project .....	25
2.3 Save Project .....	26
2.4 Save As Project .....	27
2.5 Load Model .....	28
2.6 Save Model .....	30
2.7 Save As Model .....	30
2.8 Make Report .....	31
2.9 Export .....	32
2.10 Save Screens .....	37
2.11 Print Screen .....	37
2.11 Print Preview .....	38
2.12 Print Setup .....	39
2.13 Exit .....	40
Chapter 3: Edit Menu .....	41
3.1 Add Module .....	41
3.2 Clear All Module .....	41
3.3 Snapshot .....	42
3.4 Undo .....	43
3.5 Cut .....	43
3.6 Copy .....	44
3.7 Paste .....	44
Chapter 4: Parameters Menu .....	45
4.1 Post Architecture Model .....	45
4.1.1 Product .....	45
4.1.2 Platform .....	46
4.1.3 Personnel .....	47
4.1.4 Project .....	47
4.1.5 User Defined EAF .....	48
4.2 Early Design Model .....	49
4.3 Scale Factors .....	49
4.4 Equation .....	50
4.5 Reset .....	51
Chapter 5: Calibrate Menu .....	52
5.1 File Load .....	52
5.2 File Save .....	53
5.3 File Save As .....	54
5.4 Project .....	54
5.5 Compute .....	56
Chapter 6: Phase Distribution .....	57
6.1 Project Phase Distribution .....	58
6.1.1 Overall Project Phase Distribution .....	58
6.1.2 Plans and Requirements Project Phase Distribution .....	59
6.1.3 Programming Project Phase .....	60

6.1.4 Product Design Phase.....	61
6.1.5 Integration and Test Project Phase .....	61
6.2 Module Phase Distribution.....	62
6.2.1 Overall Module Phase Distribution.....	64
6.2.2 Plans and Requirements Module Phase Distribution .....	64
6.2.3 Programming Module Phase Distribution.....	65
6.2.4 Product Design Phase Distribution .....	66
6.2.5 Integration and Test Module Phase Distribution.....	67
Chapter 7: Maintenance .....	69
7.1 Project Maintenance.....	70
7.2 Module Maintenance.....	75
References.....	81
Appendix A: Accelerator Keys.....	82

# Chapter 1: Introduction

## 1.1 What is COCOMO?

COCOMO (CONstructive COSt MOdel) refers to a parametric model used to estimate effort and schedule for software development projects. The COCOMO software system is a screen-oriented, interactive program that assists in budgetary planning and schedule estimation of a software development project. Through the flexibility of COCOMO, a software project manager (or team leader) can develop a model (or multiple models) of projects in order to identify potential problems in resources, personnel, budgets, and schedules both before and while the potential software package is being developed.

The COCOMO software package is based upon the software cost and schedule estimation model: CONstructive COSt MOdel version II (COCOMOII). This is the newly revised version of the original CONstructive COSt MOdel (COCOMO) first published by Dr. Barry Boehm in his book Software Engineering Economics, Prentice-Hall (1981), and Ada COCOMO (1989) predecessors. The current model is described in [Boehm et al. 1995]

The primary objectives of the COCOMOII.1999 effort are:

- To develop a software cost and schedule estimation model tuned to the life cycle practices of the 1990's and 2000's.

- To develop software cost database and tool support capabilities for continuous model improvement.

- To provide a quantitative analytic framework, and set of tools and techniques for evaluating the effects of software technology improvements on software life cycle costs and schedules.

The full COCOMOII model includes three stages. Stage 1 supports estimation of prototyping or applications composition efforts. Stage 2 supports estimation in the Early Design stage of a project, when less is known about the project's cost drivers. Stage 3 supports estimation in the Post-Architecture stage of a project.

This version of USC COCOMOII implements stage 3 formulas to estimate the effort, schedule, and cost required to develop a software product. It also provides the breakdown of effort and schedule into software life-cycle phases and activities from the original COCOMO manual. These are still reasonably valid for waterfall model software projects, but need to be interpreted for non-waterfall projects.

### 1.1.1 Effort Estimation Equation

Estimate effort with:

(EQ 1-1)

$$PM = \prod_{i=1}^{17} (EM_i) \cdot A \cdot \left[ \left( 1 + \frac{BRAK}{100} \right) \cdot Size \right]^{0.91 + 0.01 \sum_{j=1}^5 SF_j} + \left( \frac{ASLOC \cdot \left( \frac{AT}{100} \right)}{ATPROD} \right)$$

where

$$Size = KNSLOC + \left[ KASLOC \cdot \left( \frac{100 - AT}{100} \right) \cdot \frac{(AA + SU + 0.4 \cdot DM + 0.3 \cdot CM + 0.3 \cdot IM)}{100} \right]$$

$$B = 0.91 + 0.01 \sum_{j=1}^5 SF$$

Symbol	Description
A	Constant, currently calibrated as 2.45
AA	Assessment and assimilation
ADAPT	Percentage of components adapted (represents the effort required in understanding software)
AT	Percentage of components that are automatically translated
ATPROD	Automatic translation productivity
BRAK	Breakage: Percentage of code thrown away due to requirements volatility
CM	Percentage of code modified
DM	Percentage of design modified
EM	Effort Multipliers: RELY, DATA, CPLX, RUSE, DOCU, TIME, STOR, PVOL, ACAP, PCAP, PCON, AEXP, PEXP, LTEX, TOOL, SITE
IM	Percentage of integration and test modified
KASLOC	Size of the adapted component expressed in thousands of adapted source lines of code
KNSLOC	Size of component expressed in thousands of new source lines of code
PM	Person Months of estimated effort
SF	Scale Factors: PREC, FLEX, RESL, TEAM, PMAT
SIZE	In thousands of lines of code, of new or adapted source lines of code
SU	Software understanding (zero if DM = 0 and CM = 0)

### 1.1.2 Schedule Estimation Equation

Determine time to develop (TDEV) with an estimated effort, PM, that excludes the effect of the SCED effort multiplier:

(EQ 1-2)

$$TDEV = \left[ 3.67 \times (PM)^{(0.28 + 0.2 \times (B - 1.01))} \right] \cdot \frac{SCED\%}{100}$$

where

$$B = 0.91 + 0.01 \sum_{j=1}^5 SF_j$$

Symbol	Description
PM	Person Months of estimated effort from Early Design or Post-Architecture models (excluding the effect of the SCED effort multiplier).
SF	Scale Factors: PREC, FLEX, RESL, TEAM, PMAT
TDEV	Time to develop
SCED	Schedule
SCED%	The compression / expansion percentage corresponding to the SCED effort multiplier

### 1.1.3 Scale Factors

Table 1.1 provides the rating levels for the COCOMOII scale drivers. The selection of scale drivers is based on the rationale that they are a significant source of exponential variation on a project's effort or productivity variation. Each scale driver has a range of rating levels, from Very Low to Extra High. Each rating level has a weight, W, and the specific value of the weight is called a scale factor. A project's scale factors,  $W_i$ , are summed across all of the factors, and used to determine a scale exponent, B, via the following formula:

$$B = 0.91 + 0.01 \times \sum_{j=1}^5 SF_j$$

(EQ 1-3)

For example, if scale factors with an Extra High rating are each assigned a weight of (0), then a 100 KSLOC project with Extra High ratings for all factors will have  $SF_j = 0$ ,  $B = .91$ , and a relative effort  $E = 100^{0.91} = 66 PM$ . If scale factors with Very Low rating are each assigned a weight of (5), then a project with Very Low (5) ratings for all factors will have  $SF_j = 5$ ,  $B = 1.16$ , and a relative effort  $E = 209 PM$ . This represents a large variation, but the increase involved in a one level step change in one of the factors is only about 4.7%.

Scale Factors (SF <sub>j</sub> )	Very Low	Low	Nominal	High	Very High	Extra High
PREC	thoroughly unprecedented	largely unprecedented	Somewhat unprecedented	generally familiar	largely familiar	thoroughly familiar
FLEX	rigorous	occasional relaxation	some relaxation	general conformity	some conformity	general goals
RESL <sup>1</sup>	little (20%)	some (40%)	often (60%)	Generally (75%)	mostly (90%)	full (100%)
TEAM	very difficult interactions	some difficult interactions	Basically cooperative interactions	largely cooperative	highly cooperative	seamless interactions
PMAT	Weighted average of "Yes" answers to CMM Maturity Questionnaire					

**Table 1.1: Scale Factors for COCOMO.II Early Design and Post-Architecture Models**

### 1.1.4 Sizing Methods

#### **SLOC: Lines of Code Counting Rules**

In COCOMOII, the logical source statement has been chosen as the standard line of code. Defining a line of code is difficult due to conceptual differences involved in accounting for executable statements and data declarations in different languages. The goal is to measure the amount of intellectual work put into program development, but difficulties arise when trying to define consistent measures across different languages. Breakage due to change of requirements also complicates sizing. To minimize these problems, the Software Engineering Institute (SEI) definition checklist for a logical source statement is used in defining the line of code measure. The Software Engineering Institute (SEI) has developed this checklist as part of a system of definition checklists, report forms and supplemental forms to support measurement

---

<sup>1</sup>. % significant module interfaces specified,% significant risks eliminated.

definitions [Park 1992] [Goethert et al. 1992].

Figure 1-1 shows a portion of the definition checklist as it is being applied to support the development of the COCOMOII model. Each checkmark in the “Includes” column identifies a particular statement type or attribute included in the definition, and vice-versa for the excludes. Other sections in the definition clarify statement attributes for usage, delivery, functionality, replications and development status. There are also clarifications for language specific statements for ADA, C, C++, CMS-2, COBOL, FORTRAN, JOVIAL and Pascal.

Some changes were made to the line-of-code definition that departs from the default definition provided in [Park 1992]. These changes eliminate categories of software, which are generally small sources of project effort. Not included in the definition are commercial-off-the-shelf software (COTS), government-furnished software (GFS), other products, language support libraries and operating systems, or other commercial libraries. Code generated with source code generators is not included though measurements will be taken with and without generated code to support analysis.

The “COCOMOII line-of-code definition” can be calculated in several ways. One way is to use the suite of software programs known as Code Count, which are available from the Center for Software Engineering website under category Tools.

**Figure 1-1 Definition Checklist for Source Statements Counts**

**Definition Checklist for Source Statements Counts**

Definition name: Logical Source Statements Date: \_\_\_\_\_  
 \_\_\_\_\_ (basic definition) Originator: COCOMO II

Measurement unit	Physical source lines				
	Logical source statements	<input checked="" type="checkbox"/>			
Statement type	Definition <input checked="" type="checkbox"/>	Data Array			Includes Excludes
<i>When a line or statement contains more than one type, classify it as the type with the highest precedence.</i>					
1 Executable	Order of precedence →		1	<input checked="" type="checkbox"/>	
2 Nonexecutable					
3 Declarations			2	<input checked="" type="checkbox"/>	
4 Compiler directives			3	<input checked="" type="checkbox"/>	
5 Comments					
6 On their own lines			4		<input checked="" type="checkbox"/>
7 On lines with source code			5		<input checked="" type="checkbox"/>
8 Banners and non-blank spacers			6		<input checked="" type="checkbox"/>
9 Blank (empty) comments			7		<input checked="" type="checkbox"/>
10 Blank lines			8		<input checked="" type="checkbox"/>
11					
12					
How produced	Definition <input checked="" type="checkbox"/>	Data array			Includes Excludes
1 Programmed				<input checked="" type="checkbox"/>	
2 Generated with source code generators					<input checked="" type="checkbox"/>
3 Converted with automated translators				<input checked="" type="checkbox"/>	
4 Copied or reused without change				<input checked="" type="checkbox"/>	
5 Modified				<input checked="" type="checkbox"/>	
6 Removed					<input checked="" type="checkbox"/>
7					
8					
Origin	Definition <input checked="" type="checkbox"/>	Data array			Includes Excludes
1 New work: no prior existence				<input checked="" type="checkbox"/>	
2 Prior work: taken or adapted from					
3 A previous version, build, or release				<input checked="" type="checkbox"/>	
4 Commercial, off-the-shelf software (COTS), other than libraries					<input checked="" type="checkbox"/>
5 Government furnished software (GFS), other than reuse libraries					<input checked="" type="checkbox"/>
6 Another product					<input checked="" type="checkbox"/>
7 A vendor-supplied language support library (unmodified)					<input checked="" type="checkbox"/>
8 A vendor-supplied operating system or utility (unmodified)					<input checked="" type="checkbox"/>
9 A local or modified language support library or operating system					<input checked="" type="checkbox"/>
10 Other commercial library					<input checked="" type="checkbox"/>
11 A reuse library (software designed for reuse)				<input checked="" type="checkbox"/>	
12 Other software component or library				<input checked="" type="checkbox"/>	
13					
14					

### 1.1.5 FP: Counting with Unadjusted Function Points

The function point cost estimation approach is based on the amount of functionality in a software project and a set of individual project factors [Behrens 1983][Kunkler 1985][IFPUG 1994]. Function points are useful estimators since they are based on information that is available early in the project life cycle. A brief summary of function points and their calculation in COCOMOII is as follows.

Function points measure a software project by quantifying the information processing functionality associated with major external data input, output, or file types. Five user function types should be identified as defined in the Table 1-2.

External Input (Inputs)	Count each unique user data or user control input type that (i) enters the external boundary of the software system being measured and (ii) adds or changes data in a logical internal file.
External Output (Outputs)	Count each unique user data or control output type that leaves the external boundary of the software system being measured.
Internal Logical File (Files)	Count each major logical group of user data or control information in the software system as a logical internal file type. Include each logical file (e.g., each logical group of data) that is generated, used, or maintained by the software system.
External Interface Files (Interfaces)	Files passed or shared between software systems should be counted as external interface file types within each system.
External Inquiry (Queries)	Count each unique input-output combination, where an input causes and generates an immediate output, as an external inquiry type.

**Table 1-2: User Function Types**

Each instance of these function types is then classified by complexity level. The complexity levels determine a set of weights, which are applied to their corresponding function counts to determine the Unadjusted Function Points quantity. This is the Function Point sizing metric used by COCOMOII. The usual Function Point procedure involves assessing the degree of influence (DI) of fourteen application characteristics on the software project determined according to a rating scale of 0.0 to 0.05 for each characteristic. The 14 ratings are added together, and added to a base level of 0.65 to produce a general characteristics adjustment factor that ranges from 0.65 to 1.35.

Each of these fourteen characteristics, such as distributed functions, performance, and reusability, thus have a maximum of 5% contribution to estimated effort. This is inconsistent with COCOMO experience; thus COCOMO.II uses Unadjusted Function Points for sizing, and applies its reuse factors, cost driver effort multipliers, and exponent scale factors to this sizing quantity.

## 1.1.6 AAF: Adaptation Adjustment Factors

### Adaptation of Existing Code

COCOMO is not only capable of estimating the cost and schedule for a development started from "scratch", but it is also able to estimate the cost and schedule for products that are built upon already existing code, i.e. reused code. Adaptation considerations have also been incorporated into COCOMO, where an estimate for KSLOC will be calculated. This value will be substituted in place of the SLOC found in the equations already discussed. This adaptation of code utilizes an additional set of equations that are used to calculate the final count on source instructions and related cost and schedule. These equations use the following values as components:

Adapted Source Lines of Code (ASLOC). The number of source lines of code adapted from existing software used in developing the new product.

Percent of Design Modification (DM). The percentage of the adapted software's design that received modification to fulfill the objectives and environment of the new product.

Percent of Code Modification (CM). The percentage of the adapted software's code that receives modification to fulfill the objectives and environment of the new product.

Percent of Integration Required for Modified Software (IM). The percentage of effort needed for integrating and testing of the adapted software in order to combine it into the new product.

Percentage of reuse effort due to Software Understanding (SU).

Percentage of reuse effort due to Assessment and Assimilation (AA).

Programmer Unfamiliarity with Software (UNFM)

These components are brought together in Figure 1-6. The AAF is the adaptation adjustment factor. The AAF is the calculated degree to which the adapted software will affect overall development.

## 1.1.7 Effort Multipliers

There are a number of contributing factors to a project's delivery time and effort. Development productivity was found to be affected by additional factors that were found to fall under the headings: product attributes, platform attributes, personnel attributes, and project attributes.

*Product attributes* refer to the constraints and requirements placed upon the project to be developed. These included

Required software reliability (RELY)

Database size (DATA)

Documentation match to life-cycle needs (DOCU)

Product complexity (CPLX)

Required Reusability (RUSE)

*Platform attributes* refer to the limitations placed upon development effort by the hardware and operating system being used to run the project. These limitations are

listed below.

- Execution time constraint (TIME)
- Main storage constraint (STOR)
- Platform volatility (PVOL)

*Personnel attributes* refer to the level of skills that are possessed by the personnel. The skills in question are general professional ability, programming ability, experience with the development environment and familiarity with the project's domain. These skills are characterized below.

- Analyst capabilities (ACAP)
- Applications experience (AEXP)
- Programmer capabilities (PCAP)
- Platform experience (PEXP)
- Programming language experience (LEXP)
- Personnel Continuity (PCON)

*Project attributes* refer to the constraints and conditions under which project development takes place. The issues that affect development are:

- Use of software tools (TOOL)
- Multisite Development (SITE)

These 16 factors are incorporated into calculating an estimated effort and schedule. Each of the factors has associated with it up to six ratings. These ratings are *very low*, *low*, *nominal*, *high*, *very high*, and *extra high*. Each rating has a corresponding real number based upon the factor and the degree to which the factor can influence productivity. A rating less than 1 denotes a factor that can decrease the schedule and effort. A rating greater than 1 denotes a factor that extends the schedule or effort. Lastly, a rating equal to 1 does not extend nor decrease the schedule and effort (this rating is called *nominal*).

These 16 factors (or effort multipliers) are incorporated into the schedule and effort estimation formulas by multiplying them together (see Figure 1-8 for the COCOMO dialog box). The numerical value of the  $i^{\text{th}}$  adjustment factor (there are 16 of them) is called  $EM_i$  and their product is called the adjustment factor or EAF. The actual effort,  $PM_{\text{total}}$  is the product of the nominal effort times the EAF.

In addition to the 16 EAF factors there are two user defined factors named USR1 and USR2. Their initial values are all set to 1. They may be redefined by using the Parameters-User Defined EAF menu item.

A final effort multiplier, Required Development Schedule (SCED) is treated separately as it operates at the overall project level rather than potentially varying from module to module.

$$PM_{total} = (SCED) \times PM_{nominal} \times \prod_{i=1}^{18} E_{Mi}$$

## 1.2 Navigating COCOMO

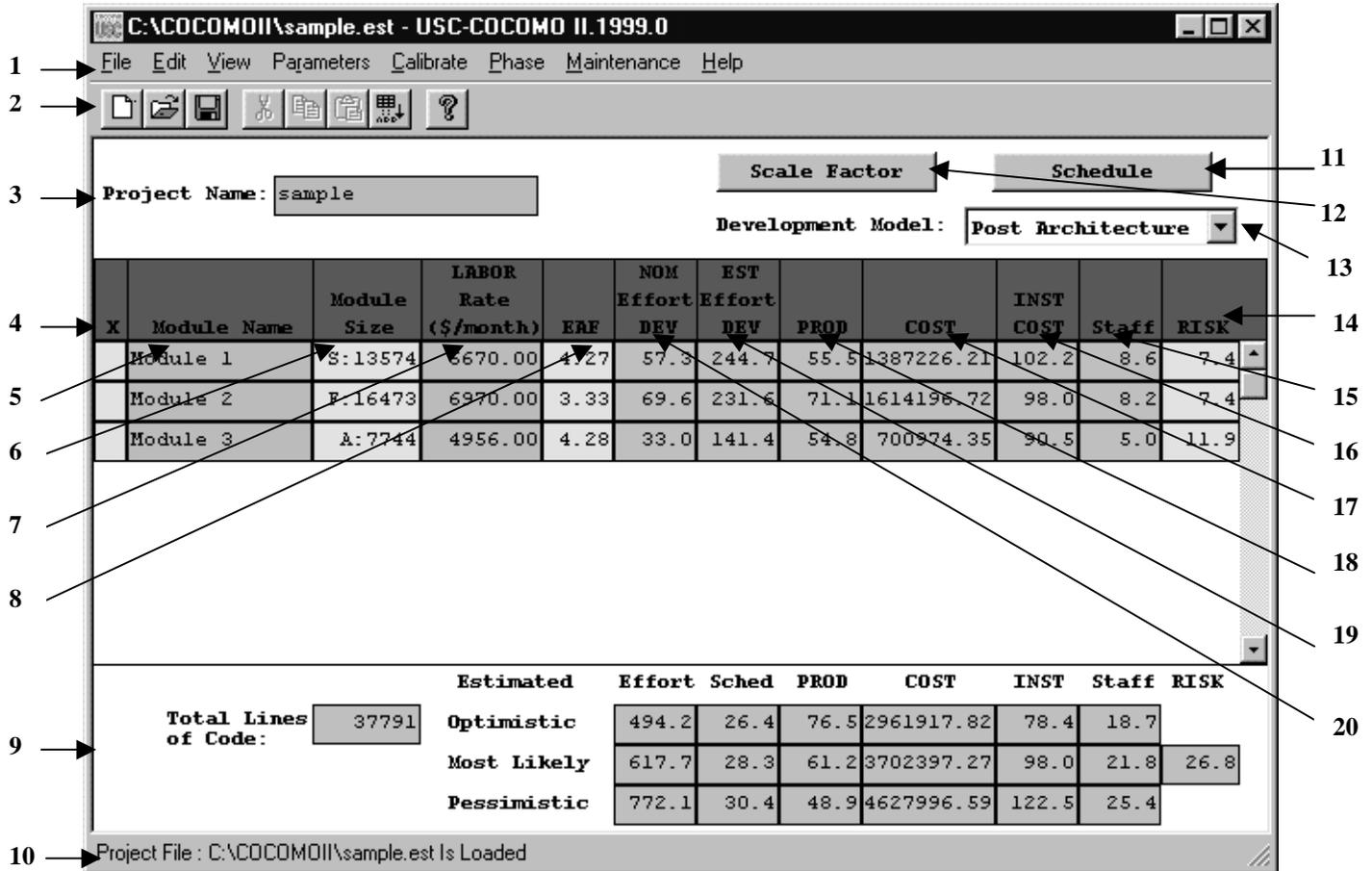
This software is a stand-alone software system intended for a single user. The software is user interactive in that it attempts to interface well with a user's needs, using extensive mouse interaction wherever possible.

On the screen in Figure 1-3 is the CLEF (Component Level Estimation Form). This is where all of the entered information will be displayed. The top of the screen shows all of the subfunctions which the user may call. The choices appear in pop down menus according to the major headings of Project, Model, and Phase.

In order to efficiently use COCOMO, you must become familiar with the Component Level Estimating Form (CLEF). The different areas that are to be discussed have been given a corresponding number. These sections are given a descriptive label as a point of reference as well as a summary of their contents and functions

The sections found in Figure 1-3 and their descriptions are as follows:

1. **Main Menu bar** - This area contains the menu selection of the main functions of COCOMO. These selections are *File*, *Edit*, *View*, *Parameters*, *Calibrate*, *Phase [Distribution]*, *Maintenance*, and *Help*. *File*, *Edit*, *View*, *Parameters*, *Calibrate*, *Maintenance*, and *Phase Distribution* are discussed in chapters 2, 3, 4, 5, 6 and 7, respectively. *Help* is the selection used to receive on-line assistance with the available functions.
2. **Tool bar** - This area contains image buttons like other windows applications for *New Project*, *Open Project*, *Save Project*, *Delete Module*, *Copy & Paste*, *Insert clipboard content*, *Insert a module*, and *About* functions.
3. **Project Name** - This editable field displays the name of the currently displayed project. To edit the name click twice upon this field and proceed to edit name. Upon completion of editing press the "Return" key. The Default name of a new project is "<sample>".



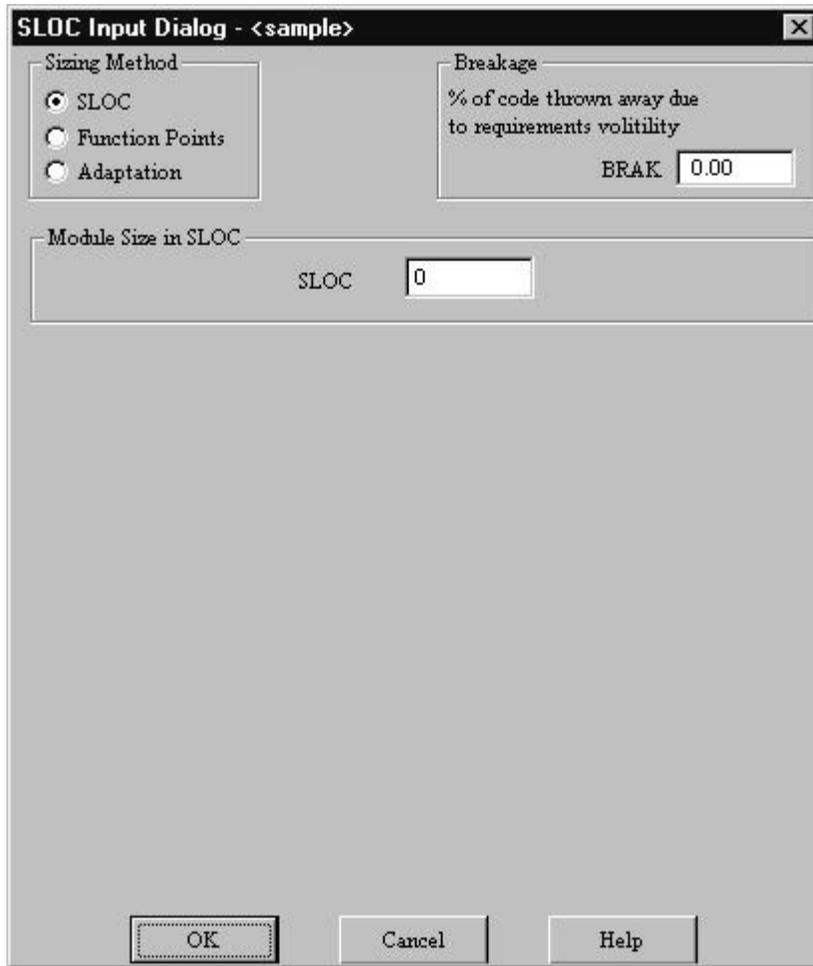
4. **X** - This column is reserved for identifying a module. Pressing upon this field for a given module will mark the desired module. Marking is denoted by an x that appears in this column. Only one module can be marked at a time. Modules are marked in order to perform module deletion, cutting, copying or pasting.

5. **Module Name Column** - This column is used to house the name of each module located in the Module Area. The module name can be changed by clicking twice on the desired module name box and entering the changes into the module name field. Upon completion of editing press "Return".

6. **Module Size (SLOC) Column** - This column shows the SLOC of each module located in the Module Area. The value for SLOC can be computed in one of three ways. One, the value can be entered directly in the SLOC field as shown in Figure 1-4. Two, by using the function point model as shown in Figure 1-5. Three, by using Adaptation Adjustment Factor as shown in Figure 1-6. Each of the three dialog boxes allows the setting of BRAK. Upon completion click on OK in the associated dialog box. There is a limit to the range of input. The input or computed value for SLOC must be within the range 0 - 9,999,999.

Note - COCOMO is not calibrated for Total SLOC < 2000 and a warning is issued..

**FIGURE 1-4** SLOC Dialog Box - Source Lines of Code (SLOC)



**SLOC Input Dialog - <sample>** [X]

**Sizing Method**

SLOC  
 **Function Points**  
 Adaptation

**Breakage**  
 % of code thrown away due to requirements volatility  
 BRAK:

**Module Size in Function Points**

Language:  [v]

Function Type	# of Function Points			SubTotal
	Low	Average	High	
Inputs	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	0
Outputs	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	0
Files	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	0
Interfaces	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	0
Queries	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	0
Total Unadjusted Function				0
Equivalent Total in SLOC				0

OK    Cancel    Help

**SLOC Input Dialog - <sample>** [X]

**Sizing Method**

SLOC  
 Function Points  
 Adaptation

**Breakage**  
 % of code thrown away due to requirements volatility  
 BRAK: 0.00

**Adaptation**

Initial SLOC	0	
% Design Modified	100	%
% Code Modified	100	%
% Integration Modified	100	%
Software Understanding (SU)	30	SU
Assesment & Assimilation (AA)	4	AA
Unfamiliarity with Software	0.4	UNFM
% Components Automatically Translated	0	%
Automatic Translation Productivity	2400	
Computed Adaptation Adjustment Factor	100	
Computed ASLOC	0	

OK    Cancel    Help

7. **Labor Rate Column** - This column contains the amount of money at which a developer working on a particular module would be paid per month. The labor rate can be edited by clicking on the corresponding Labor Rate box and entering the new value via the edit area. The range on labor rate is between \$0 and \$99,999.

8. **Effort Adjustment Factor (EAF) Column** - This column displays the product of the effort multiplier cost drivers for each specific module. By clicking on this field a dialog box appears (see Figure 1-7). This box displays all of the cost drivers, inter cost drivers and their current ratings. The cost drivers are divided into the groupings: *Product*, *Platform*, *Personnel* and *Project*. The increments on the cost drivers are rated as 0%, 25%, 50%, and 75%, i.e the amount of change to the next rating level. The ratings for each multiplier can be changed by cycling through the available ratings until the desired rating is displayed. As the cost driver ratings are changed the total product of the cost drivers is displayed in the upper right hand corner of the dialog box along with the module name. The final rating of a cost driver is calculated using this formula for the interpolation.

Final rating = (Next cost driver rating - Current cost driver rating) \* Current inter cost driver / 100

COCOMO supports two different models, called Early Design and Post Architecture. The Early Design model is supposed to be used at the earliest phase of a software project. The Post Architecture Model applies once a software architecture has been formulated. In terms of the COCOMO program, the Early Design Model differs from the Post Architecture Model in its use of Effort Adjustment Factors. The Early Design Model considers only seven pre-defined effort adjustment factors (including schedule SCHED). Only six are shown in Figure 1-7. The Post Architecture Model makes use of seventeen pre-defined effort adjustment factors and sixteen of these are shown in Figure 1-8. Each of the models can be specified by selecting it using the button in the upper right portion of the COCOMO main screen.

9. **Totals Area** - This area houses the calculated results of all of the modules combined. Within this area is the total SLOC count for the project, the total estimated: effort (EST PM), schedule (Sched), productivity (PROD), cost (COST), cost per instruction (INST), full time software programmers (FSWP), and risk. In addition to a most likely estimate there is also an optimistic estimate (no less than this, 90% of the time) and a pessimistic estimate (no greater than this, 90% of the time).

10. **Status bar** - This window displays a short definition of the column headings clicked upon and also displays a short description of the result of the last function initiated by the user.

11. **Schedule Button** - This button displays the Schedule Dialog Box as shown in Figure 1-9. Schedule applies to the entire project.

12. **Scale Factor Button** - This button displays the Scale Factor Dialog Box as shown in Figure 1-10. Scale factors apply to the project size as a whole.

13. **Model Selection button** - This button displays COCOMOII Post Architecture and Early Design Model. One of the Models can be selected and applied to the project.

14. **Risk Column** - This column contains the Total risk level for each specific module. By clicking on this field a dialog box appears (see Figure 1-11). This box displays all of the COCOMOII-related risk levels for the chosen module. The total risk of a module is computed as:

total\_risk=schedule\_risk+product\_risk+personnel\_risk+process\_risk+platform\_risk+reuse\_risk;

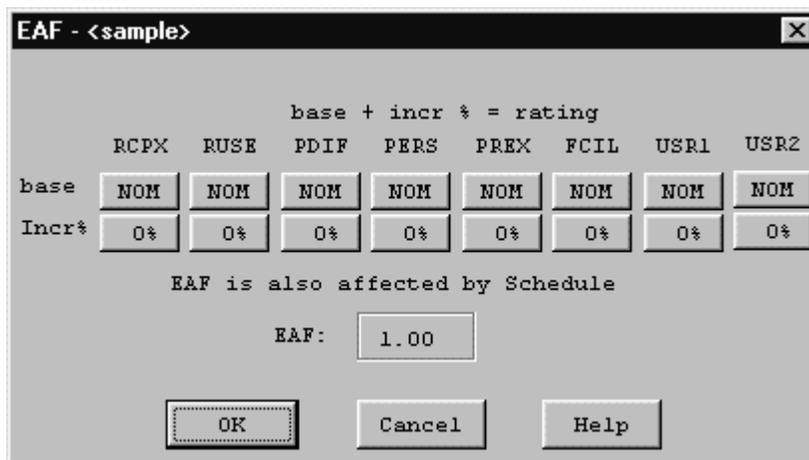
total risk of a module=total\_risk/373.\*100.;

For the definitions of schedule risk, product risk, platform risk, personnel risk, process risk, and reuse risk, see [Madachy 1997].

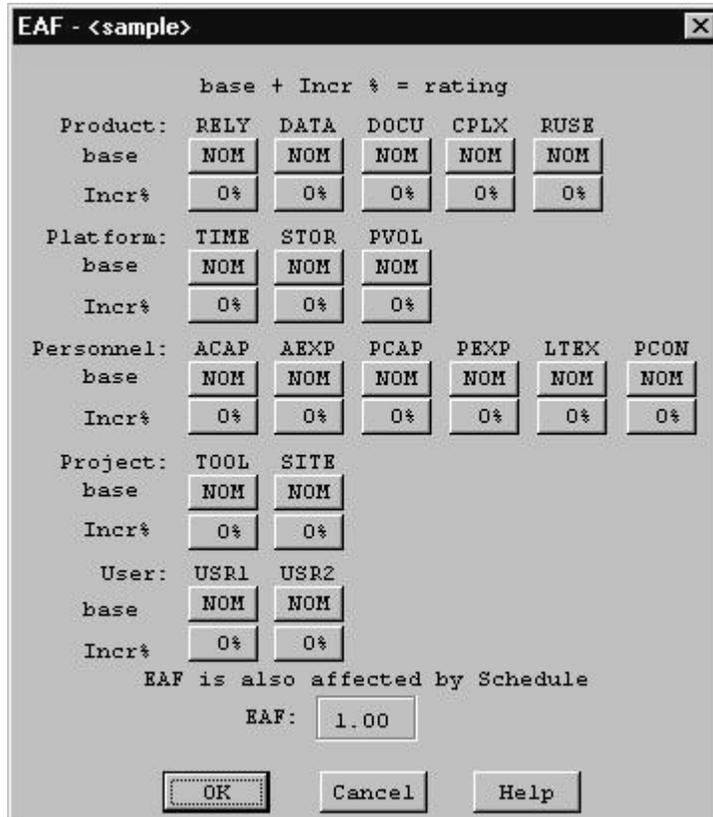
15. **Full-time SoftWare Personnel (FSWP) Column** - This column houses the calculated most likely estimate for the number of full-time developers that would be needed to complete a module in the estimated development time.

16. **SLOC Cost Column** - This column contains the calculated most likely cost per SLOC. This number is calculated from Cost/SLOC in each module.

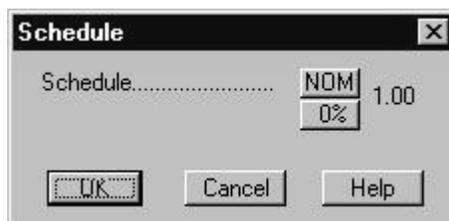
**FIGURE 1-7 EAF Dialog Box – Early Design**



**FIGURE 1-8 EAF Dialog Box – Post Architecture**



**FIGURE 1-9 Schedule Dialog Box**



17. **Cost Column** - This column contains the calculated most likely estimate of the development cost for a particular module.

18. **Productivity (PROD) Column** - This column contains the calculated result of the module's individual SLOC divided by the module's most likely effort estimate.

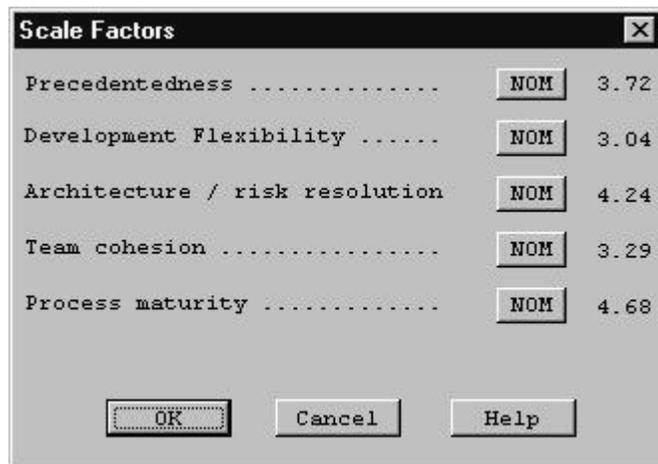
19. **Estimated Person-Month (EST PM) Column** - This column holds the module's most likely effort estimate obtained from multiplying Effort Adjustment Factor (EAF) by Nominal Person Month (NOM PM).

20. **Nominal Person-Month (NOM PM) Column** - This column holds the module's most likely effort estimate without incorporating the Effort Adjustment Factors (EAF), but including the proportional effect of the scale factors.

---

**FIGURE 1-10**                      **Scale Factor Dialog Box**

---



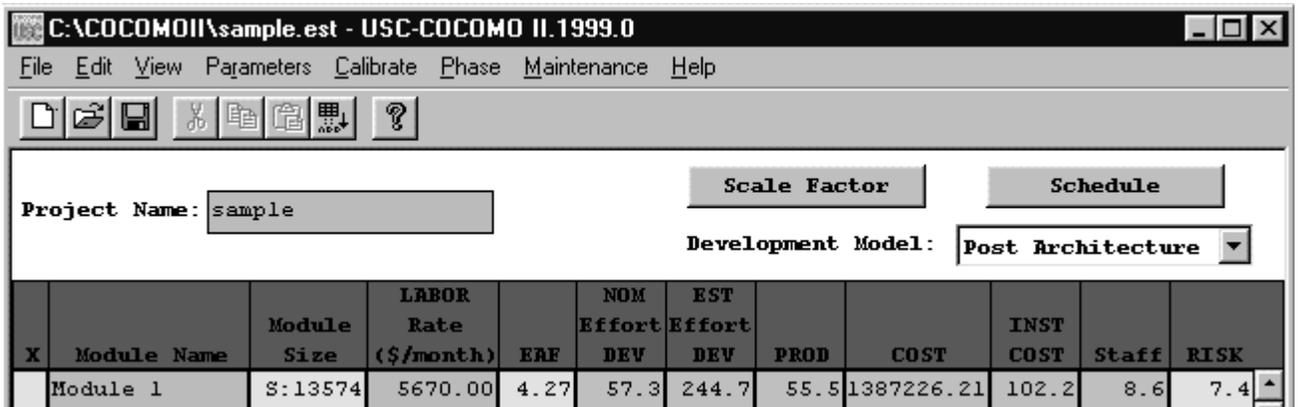
**FIGURE 1-11 Risk Level Dialog Box**



### 1.3 Begin Using COCOMO

To begin entering a new module, either click on the "Add Module" button on the Tool bar or on the pull-down menu item(Edit|Add Module). At this point, a new module will appear in the CLEF with all values set to their respective defaults. Double click upon the module name field in order to give the new module a name. Upon typing the module name press "Return.". A value for SLOC and Labor rate may also be given by clicking on the respective field and editing appropriately (see Figure 1-12).

**FIGURE 1-12 Create Sample Module and give values to SLOC and Labor Rate**



*NOTE* - In order to change any of the editable fields, just click on the desired field twice and begin editing the field. Upon completing editing, either hit the "Return" key, or click on OK. All of the final results can be found at the bottom of the CLEF in the Totals area (see Figure 1-13).

**FIGURE 1-13 Totals area after calculations have been completed**

		Estimated	Effort	Sched	PROD	COST	INST	Staff	RISK
Total Lines of Code:	37791	Optimistic	494.2	26.4	76.5	2961917.82	78.4	18.7	
		Most Likely	617.7	28.3	61.2	3702397.27	98.0	21.8	26.8
		Pessimistic	772.1	30.4	48.9	4627996.59	122.5	25.4	

Project Is Saved To File : C:\COCOMOII\sample.est

## 1.4 Running COCOMO: Windows/NT and UNIX

Currently, there are two implementations of COCOMOII: a Windows 95/98/NT version and a Sun Microsystems Sparcstation under Solaris version.

To download any these versions, you should enter this in a web browser:

<http://sunset.usc.edu/COCOMOII/cocomo.html>

and scroll down to the section labeled **COCOMOII Downloads(Software and Documentation)**

or

[ftp://ftp.usc.edu/pub/soft\\_engineering/COCOMOII/](ftp://ftp.usc.edu/pub/soft_engineering/COCOMOII/) where you will see 6 files:

c99sunos.tar.gz, c99windows.zip, usersman.ps, usersman.pdf, modelman.ps, modelman.pdf

## Chapter 2: File Menu

The COCOMO file types include: project file, model file, report file, calibration file, and csv file. The first three are discussed here. The others are discussed in later chapters.

The project file in COCOMO stores a project's data, which include project name, project scale factors, project schedule constraint, module name, SLOC, labor rate, effort adjustment factors (EAF), and COCOMO related calculation results. The COCOMO system gives all project files an ".est" extension.

Regarding the model file, as we mentioned in chapter one, COCOMO incorporates 17 predictor factors, or cost driver attributes, which are grouped into four categories: software product attributes, platform attributes, personnel attributes, and project attributes. Each of these cost drivers attributes determines a multiplicative factor, which estimates the effect of the attribute in software development effort. There are also two user defined EAF factors plus the project-level required development schedule EAF factor. Besides these cost drivers, COCOMO also has scale factors. These multiplying factors and effort estimating equations constitute the model of a project. COCOMO has assigned default values and equations for the annually calibrated default model. Each time a COCOMO project is created, its effort estimate is based on the default model. COCOMO provides flexibility in changing the values of multiplicative factors, effort estimating equation, or schedule estimating equation. An adjusted model can no longer be considered default values, and therefore it will be lost if COCOMO is exited without saving it in a model file. Upon saving this model file, these altered values can be applied to another project by loading the saved model file. The COCOMO system gives all model files a ".mod" extension.

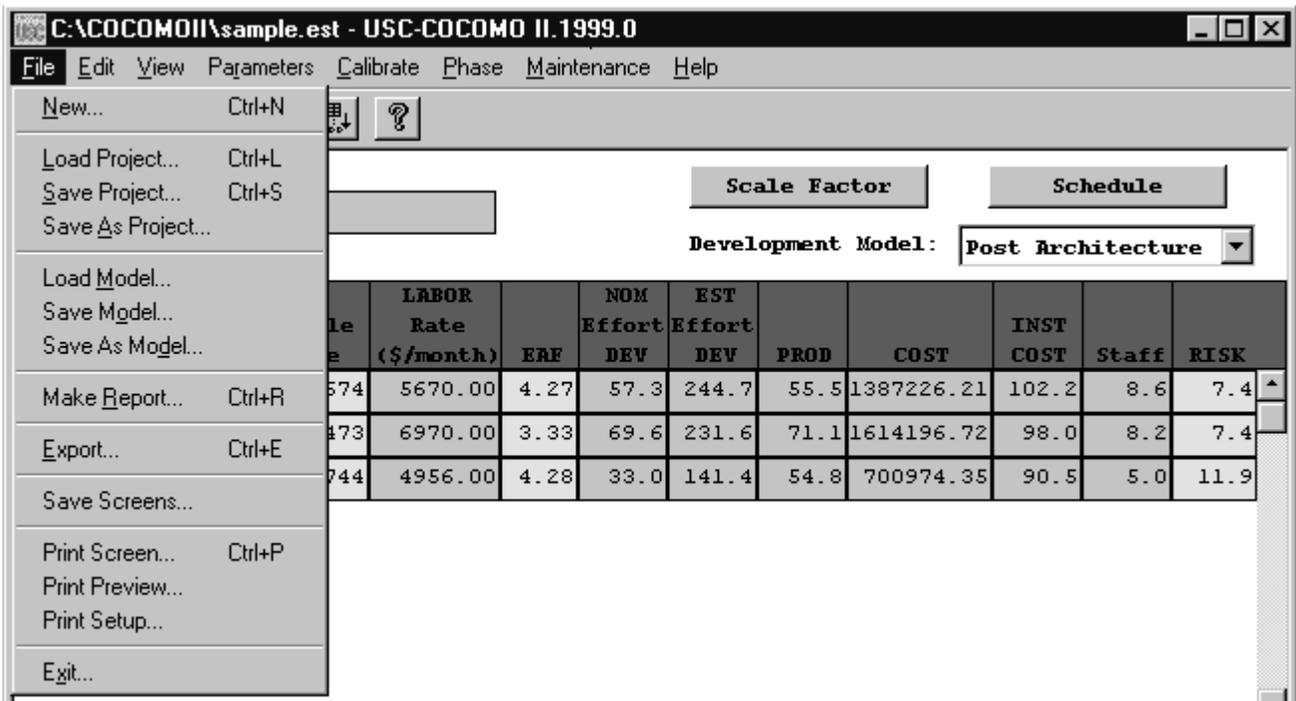
The report file is a summary report of the COCOMO project. This report contains all entered and calculated values of a project in a formatted text file. These files are given a ".rpt" extension.

The File menu option will enable you to create, retrieve, save, or print COCOMO files.

To select the File menu and its options, click on File with the mouse. The File menu will appear as Figure 2-1.

FIGURE 2-1

File Menu



## 2.1 New

The New option creates a new project file in the COCOMO working window, replacing any previous project file in the working window.

### To Create a New Working File

1. Choose New from the File menu with mouse.

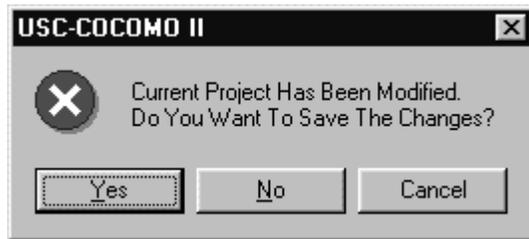
The working window will now be clear; the previous project file in the working window has been removed.

**Note:** New can be selected anytime; however, if the previous project file or model file has been modified, a warning dialog box will appear and requests confirmation. (as seen in Figure 2-2)

---

**FIGURE 2-2****Warning Dialog Box**

---



2. If the modifications on the previous file are not to be saved, choose Yes, otherwise choose No. If the No is selected, a Save File dialog will appear. (See Save Project and Save Model respectively)

## **2.2 Load Project**

The Load Project option is used to retrieve a project file as well as loading it on the working window.

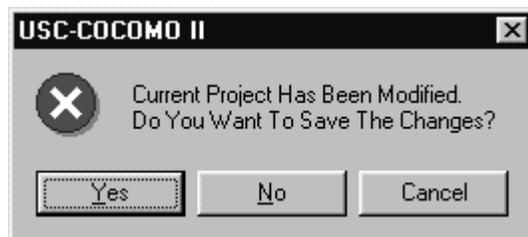
### **To Retrieve or Load a Project File**

1. Choose Load Project from the File menu with the mouse.
2. If a previous project file has been modified in the working window, the dialog box as in Figure 2-3 will appear.

---

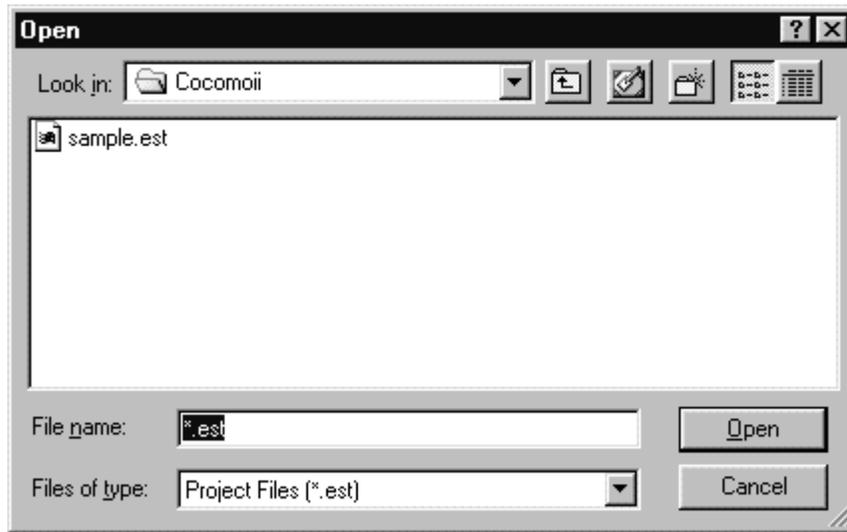
**FIGURE 2-3****Warning Dialog Box**

---



3. If the previous project file is to be saved, choose Yes, then a Save File dialog box will appear. (See Save Project). If the modified file is not to be saved, choose No.

4. The Load Project dialog box will appear as seen in Figure 2-4.



The file name of a COCOMO project has a default format with ".est" as an extension. With this window, the desired project file can be selected from the Files scroll list for loading. If the desired project file does not exist in the scroll list, it is necessary to navigate to an appropriate directory.

5. Choose desired directory for file loading

6. When the desired file is shown on the Files list, click it, and click the "OK" button to initiate project loading.

7. After a project file is loaded, its file name will be displayed on the PROJECT FILE field at upper left corner on the working window, and all modules and related items will be displayed in the CLEF area. If the number of modules is beyond the window scope, the scroll bar can be used to look at all items.

### 2.3 **Save Project**

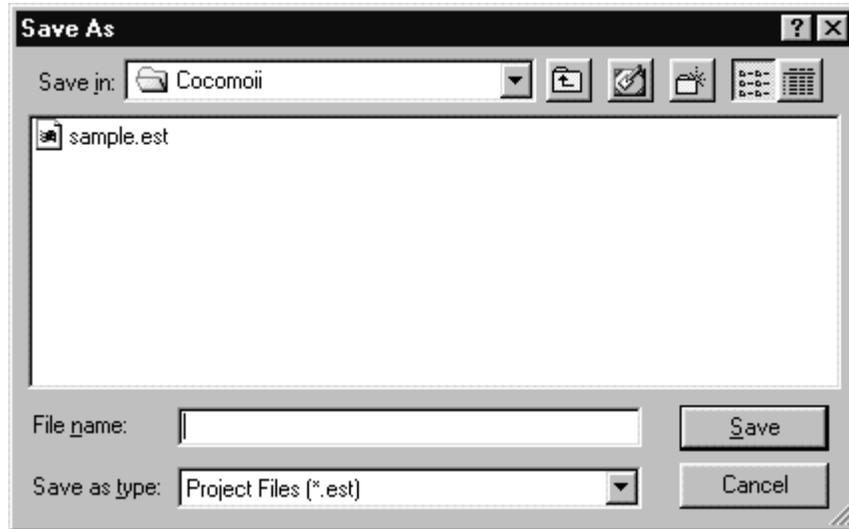
The Save Project option is used to store the results of the current COCOMO project as a file with ".est" extension.

#### **To Store the Results of Current Project**

1. Choose Save Project from the File menu with the mouse. If the current project is loaded from a previously stored project file, the Save Project will overwrite the same project file with the current project.

2. If the current project is a new one, i.e., being created by the New command, the Project Save dialog box will appear, as seen in Figure 2-5.

**FIGURE 2-5** Save Project Dialog Box



3. Look at the Files scroll window. If the file saving is to update (overwrite) a existing project file, the desired filename should be found in the Files scroll list. If the filename can not be found from current list, change the directory from the Directories scroll list until the desired filename is being shown. When the desired filename is on the list, click it.

4. If the file saving is to store a new project file, choose the desired directory, then type in a new filename.

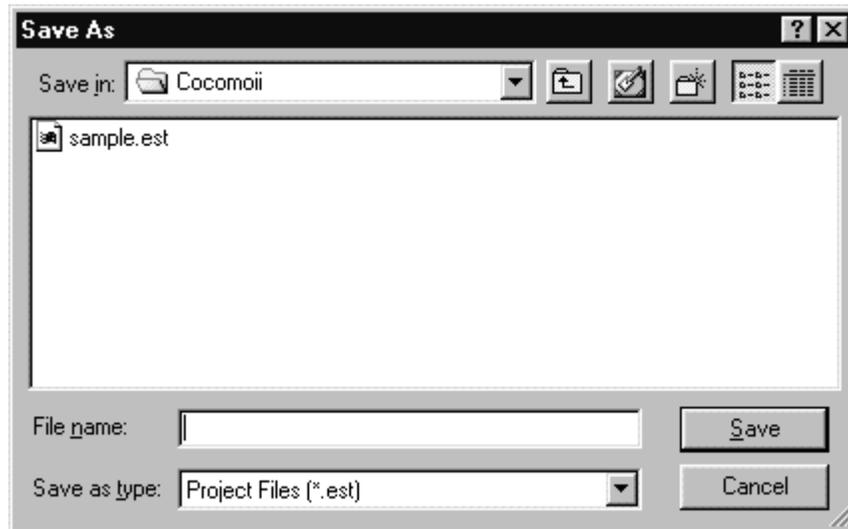
5. After the desired filename is selected or input, click the OK button to initiate project saving.

## **2.4 Save As Project**

The Save As Project option is to store the current project as a COCOMO project file, which has a file name different from current file.

### **To Store Current Project With different File Name**

1. Choose Save As Project from the File menu with the mouse.
2. The Save Project dialog box will appear, as seen in Figure 2-6.



3. Look at the Files scroll window. If the file saving is to update (overwrite) a existing project file, the desired filename should be found in the Files scroll list. If the filename can not be found from current list, change the directory from the Directories scroll list until the desired filename is being shown. When the desired filename is on the list, click it.

4. If the file saving is to store a new project file, choose the desired directory, then type in a new filename in the SELECTION box.

5. After the desired filename is selected or input, click the OK button to initiate project saving. After a project file is saved, the project file name will be displayed on the PROJECT FILE field at the upper left corner of the working window.

## 2.5 Load Model

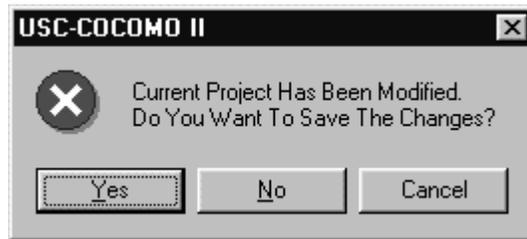
The Load Model command is used when a specific model, in which the values of multiplying factors and scale factors are different from the COCOMO default model, is to be applied to the current project.

The Load Model option is used to retrieve a model file as well as loading it for the current project.

### To Retrieve or Load a Model File

1. Choose Load Model from the File menu.
2. If a previous model has been modified in the current project, the following dialog box will appear.

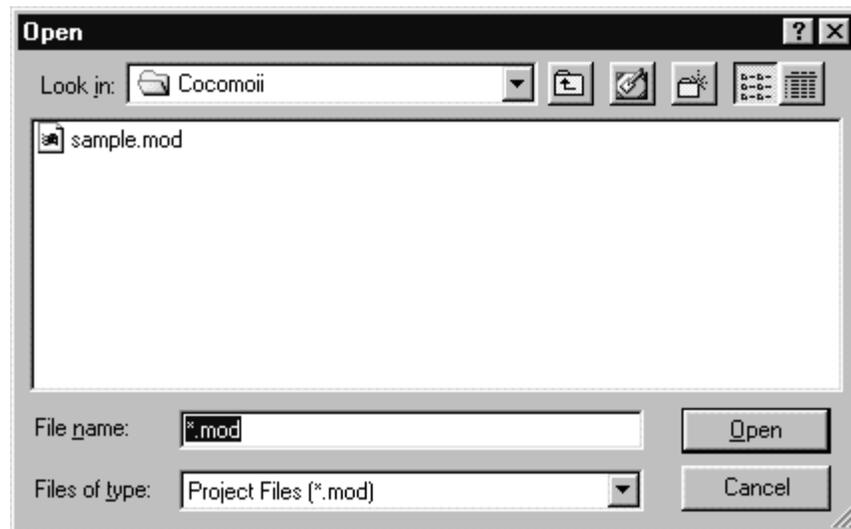
**FIGURE 2-7** Warning Dialog Box



3. If the previous model file is to be saved, choose Yes, then a Save Model dialog box will appear. (See Save Model). If the modified model is not to be saved, choose No.

4. The Load Model dialog box will appear as seen in Figure 2-8.

**FIGURE 2-8** Load Model Dialog Box



The file name of a COCOMO model has a default format with ".mod" as an extension. With this window, the desired model file can be selected from the Files scroll list for loading. If the desired model file does not exist in the scroll list, look for it in the other directories.

5. Choose desired directory for file loading

6. When the desired file is shown on the Files list, click it, and click the "OK" button to initiate model loading.

7. After a model file is loaded, its file name will be displayed on the MODEL FILE field at upper left corner on the working window, and the related costs of current project will be recalculated and shown on the working window.

## 2.6 Save Model

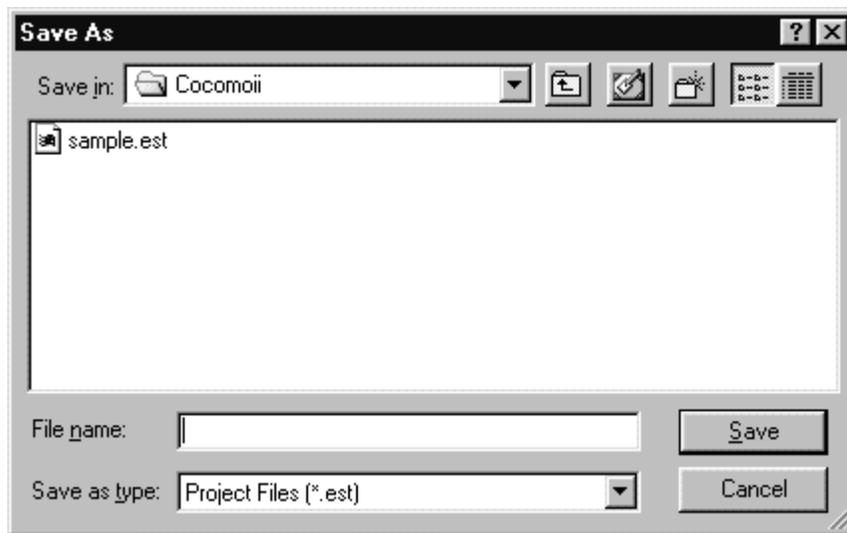
The Save Model option is used to store the results of the current COCOMO model as a file with ".mod" extension.

### To Store the Results of Current Model

1. Choose Save Model from the File menu. If the current model is loaded from a previously stored model file, the Save Model will overwrite the same model file with the current model.

2. If the current model is a new one, the Save Model dialog box will appear, as seen in Figure 2-9.

**FIGURE 2-9** Save Model Dialog Box



3. Look at the Files scroll window. If the file saving is to update (overwrite) an existing model file, the desired filename should be found in the Files scroll list. If the filename can not be found on the list, change the directory from the Directories scroll list until the desired filename is being shown. When the desired filename is on the list, click it.

4. If the file saving is to store a new model file, choose the desired directory, then type in the filename.

5. After the desired filename is selected or input, click the OK button to initiate model saving.

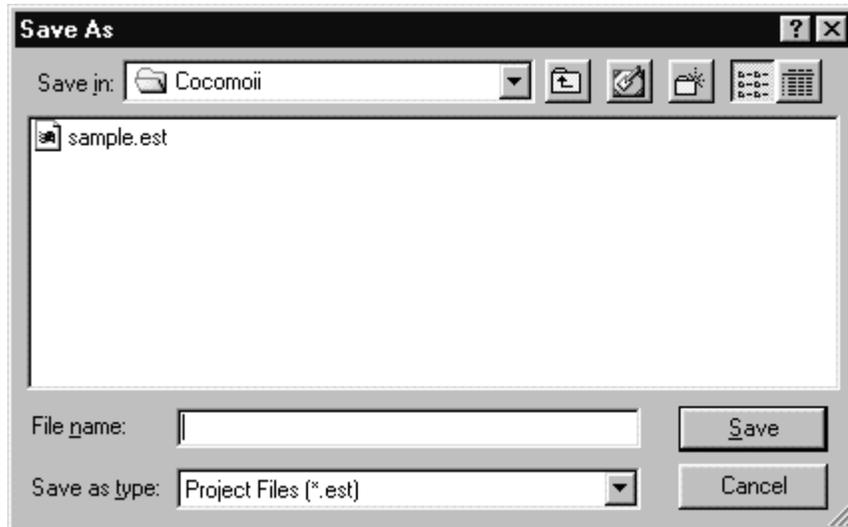
## 2.7 Save As Model

The Save As Model option is to store the current model as a COCOMO model file, which has a file name different from current model.

### To Store Current Model With different File Name

1. Choose Save As Model from the File menu.
2. The Save Model dialog box will appear, as seen in Figure 2-10.

**FIGURE 2-10** Save As Model Dialog Box



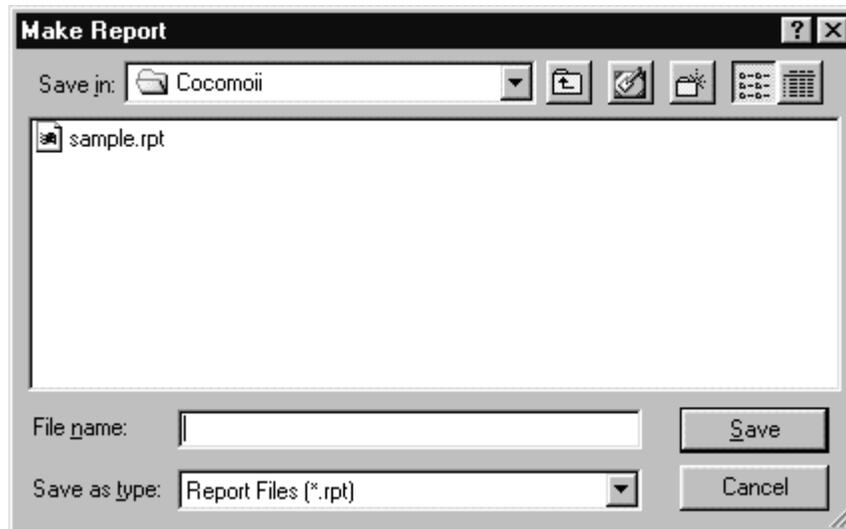
3. Look at the Files scroll window. If the file saving is to update (overwrite) an existing model file, the desired filename should be found in the Files scroll list. If the filename can not be found on the current list, change the directory from the Directories scroll list until the desired filename is being shown. When the desired filename is on the list, click it.
4. If the file saving is to store a new model file, choose the desired directory, then type in the filename in the SELECTION box.
5. After the desired filename is selected or input, click the OK button to initiate model saving. After a model file is saved, the project file name will be displayed on the MODEL FILE field at the upper left corner of the working window.

## **2.8 Make Report**

The Make Report option creates a COCOMO report in the form of a text file for printing.

### **To Create Project Report**

1. Choose Make Report from the File menu.
2. The Make Report dialog box will appear, as seen in Figure 2-11.



3. Look at the Files scroll window. If the file saving is to update (overwrite) an existing report file, the desired filename should be found in the Files scroll list. If the filename can not be found on the current list, change the directory from the Directories scroll list until the desired filename is shown. When the desired filename is on the list, click it.

4. If the file saving is to store a new report file, choose the desired directory, then type in the filename.

5. Choose desired directory for file saving: Look at the filter input box. The path found in this box represents the directory where the report file is going to be saved. This path will be changed after each directory change. To change the directory, click the appropriate directory choice from the Directories scroll list, then click the "Filter" button.

6. After the desired filename is selected or inputted, click the OK button to initiate report file saving.

7. To print a COCOMO project report, execute the local commands for your system in order to send the file for printing.

## 2.9 *Export*

The Export option lets you select a directory to write files that can be imported into Excel.

COCOMO Import & Analyze Tool is an Excel template which automates the process of importing the Comma Separated Values (CSV) files, produced by the COCOMO File | Export command. This command produces two sheets: Main, which contains Project and Module information, and Phases, which contains the phase distribution information. DO NOT rename those worksheets.

The template automates the process of generating charts which are useful for software project management. Though the template produces many useful charts, it is still possible to perform your own analysis of the COCOMO data. However to do that you must have a detailed knowledge of Excel and become familiar with the format of the csv files produced by COCOMO. Note that Excel does not permit you to open two workbooks with the same name, even if the workbooks are in different folders. To open the second workbook, you can either close the current workbook, or rename it. However it is recommended that you do not rename cocomo.xls

#### Step 1: Start Excel and import the files generated by COCOMO

Click on Import COCOMO Files. The Browse For Folder dialog box will appear. Select the folder containing the Comma-Separated Values Files (CSV) generated by USC-COCOMO. If COCOMO Import&Analyze can find the files it is looking for (Main.csv and Phases.csv) it will open each file, read its contents and place it in the appropriate worksheet.

#### Step 2: Generate Charts

Once the sheets 'Main' and 'Phases' have been correctly initialized, you can generate meaningful analyses of the data, with the click of a button. Specific help on each chart is given below.

If the sheet corresponding to the chart you are trying to create has been already created, then the existing sheet will be activated, and get the input focus. If you want to overwrite the existing sheet, you will have to delete it by selecting the Delete Sheet command from the Edit menu.

#### Performing your own analyses

The charts that the COCOMO Import and Analyze tool can generate for you are just a subset of what you can do with the data imported into the 'Main' and 'Phases' sheets. COCOMO Import and Analyze Tool provides you with maximum flexibility by:

Copying the relevant parts of 'Main' and 'Phases' sheets into the sheet for the purpose of the generation of graphs; Using references to the values entered in 'Main' and 'Phases' sheets, so that if you update the values in those sheets, the changes will be reflected in all the other 'calculated' sheets, as well as the charts. Note however, that if you introduce changes in any calculated sheet, the changes won't be carried over to Main or Phases; Allowing you to change some values. In particular, the sheets 'Project Cumulative Cost' and 'Project Cost per Phase' allow you to enter the Cost per Person-Month per Phase. The provided value is just an average computed from the COCOMO values. However you can use the value which reflects more accurately your organizational process.

If you click on the button marked Delete All Generated Sheets, and you select OK, all the 'calculated' sheets will be deleted. If you introduce changes to one of the 'calculated' sheets, and to prevent the sheet from being deleted, it is recommended that you rename the sheet.

The 'Menu' sheet is protected to avoid accidental displacement of the various buttons. It also disables resizing rows and columns, as well as selecting a range of cells. If you need to make changes to the 'Menu' sheet, toggle the 'Protect Sheet/Unprotect Sheet' from the Protection command under the Tools menu.

#### Chart Explanations

##### *Project Schedule per Phase*

This chart is useful for determining the total calendar time required for the completion of the project.

COCOMO computes the schedule as the calendar time in months from the determination of a product's requirements baseline to the completion of an acceptance activity certifying that the product satisfies its requirements. Time spent in requirements is computed as an additional 20%.

### *Cost per Phase*

This chart is useful for determining the cumulative cost as the project moves from one phase to another. It also displays the milestones between the phases (using the assumption of the Waterfall Model).

To obtain the Cost per Phase, given that the COCOMO computes only the Effort per Phase, it is necessary to compute the average Cost per Person-Month as  $\text{Total Cost for entire project (excluding requirements) / Total Effort in Person-Months}$ . Effort for requirements is assumed at an additional 7% of the Total Effort, and with the same Cost per Person-Month as the overall project. If your organization uses other Costs per Person-Month, you can use those values instead.

### *Cumulative Cost*

This chart is useful for determining the cumulative cost of a project over time. It also displays the milestones between the phases (using the assumption of the Waterfall Model).

To obtain the Cost per Phase, given that the COCOMO computes only the Effort per Phase, it is necessary to compute the average Cost per Person-Month as  $\text{Total Cost for entire project (excluding requirements) / Total Effort in Person-Months}$ . Effort for requirements is assumed at an additional 7% of the Total Effort, and with the same Cost per Person-Month as the overall project. If your organization uses other Costs per Person-Month, you can use those values instead.

### *Cost per Module*

This chart is useful for determining the cost of every module, and comparing which modules are accounting for the highest part of the cost. The cost of the entire project is the sum of the costs of the individual modules. Since these are only estimates, error bars are used to display the range within which the actual cost will fall, with a very high probability. COCOMO generates only the Optimistic (as  $0.8 \times \text{Most Likely}$ ) and the Pessimistic bound (as  $1.25 \times \text{Most Likely}$ ) for the cost of the entire project. The Optimistic/Pessimistic bounds on the cost estimates of the individual modules are similarly computed. This assumes that the error on the cost estimate for the entire project is the sum of the errors on the cost of the component modules. You can modify the Optimistic/Pessimistic bounds on the individual modules to match your organizational experience. The error bars will be adjusted accordingly.

### *Effort*

#### *Project Effort per Phase*

This chart displays how the total effort for the overall project is distributed across phases.

The Effort percentages computed by COCOMO, that are displayed under 'Percentage' exclude the effort spent on requirements. However, the percentages displayed on the chart include the effort spent on Requirements, hence the slight difference.

### *Effort per Phase and Module*

This chart displays the effort spent on every module, during the phases of the project. It can be used to quickly identify the modules which are the most effort-intensive. Effort for requirements is assumed at an additional 7% of the Total Effort.

### *Staffing*

*Project Personnel per Schedule* This chart displays the staffing levels of the project over its duration. The cutoff points between the various phases are also clearly displayed. Staffing levels are assumed to remain constant during a given phase, and to experience a sharp increase/decrease at the beginning/end of a phase.

#### *Project Personnel per Phase*

This chart displays the staffing levels of the project during the various phases. Staffing levels are assumed to remain constant during a given phase, and to experience a sharp increase/decrease at the beginning/end of a phase.

#### *Personnel per Module*

This chart displays the staffing requirements for every module. It is useful for identifying the most labor intensive modules, and to appropriately plan for the staffing of teams which are typically responsible for modules within the project.

#### *Activity*

**Project Activity per Phase** This chart displays the effort spent by activity during the project lifetime. The activities correspond to the ones associated with a waterfall model.

The effort spent per activity is computed as a percentage from the total effort spent during a phase. Refer to the EFFORT (%) section of the 'Phases' sheet to view the percentages used. The same percentages are used for the overall project, and for the individual modules. If you think from your organizational experience that some modules may require different activity distribution than others you can recompute the effort spent by activity for those modules.

#### *Project Personnel per Activity and Phase*

This chart displays the personnel required for every activity during each phase of the project. The activities correspond to the ones associated with a waterfall model.

The personnel required for every activity is computed from the effort spent per activity. The effort spent per activity is computed as a percentage from the total effort spent during a phase. Refer to the EFFORT (%) section of the 'Phases' sheet to view the percentages used. The same percentages are used for the overall project, and for the individual modules. If you think from your organizational experience that some modules may require different activity distribution than others, you can recompute the effort spent by activity for those modules.

#### *General Assumptions*

The general assumptions behind COCOMO are briefly listed below.

For all the analyses, it is assumed that the waterfall model is used; in particular, a phase does not start until the previous one has been completed.

COCOMO treats the effort and the schedule spent on plans and requirements as separate from the effort and the schedule spent on the other phases (Product Design, Programming, Integration and Test).

The Effort percentages computed by COCOMO, exclude the effort spent on requirements

COCOMO computes the schedule as the calendar time in months from the determination of a product's requirements baseline to the completion of an acceptance activity certifying that the product satisfies its requirements. Time spent in requirements is computed as an additional 20%.

COCOMO uses a single labor rate for a given module within the project, and therefore, does not use Cost per Person-Month for a phase. To compute the cumulative cost per phase, or over time, the Average Cost per Person-Month is computed as  $\text{Total Cost for entire project (excluding requirements)} / \text{Total Effort in Person-Months (excluding requirements)}$

$\text{Total Effort} = \text{Effort (Product Design)} + \text{Effort (Programming)} + \text{Effort (Integration and Test)}$

Effort for requirements is assumed at an additional 7% of the Total Effort, and with the same Cost per Person-Month as the overall project

### *Known Problems*

If the project does not have any cost information (I.e., cost-related data has not been entered and defaults to zero), some charts may be incorrectly displayed, and some cells may have a 'Division by Zero' error.

For unusually large COCOMO data sets, e.g. 50 modules, the predefined graphs may not display correctly. However, by disabling the labeling of the data points, resizing the graphs, you may be able to achieve an acceptable display.

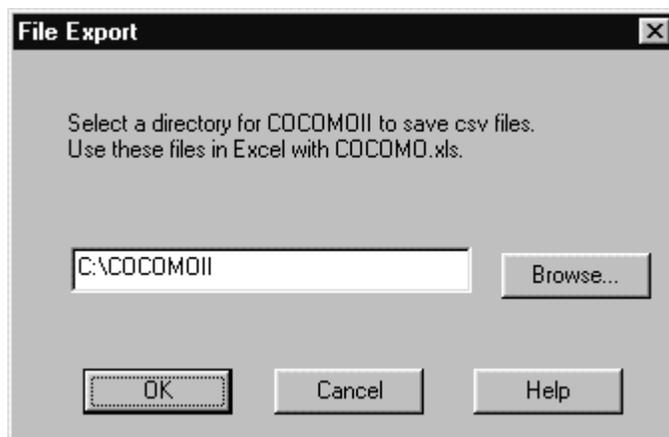
### **To Export**

1. Choose Export from the File menu.
2. The Export dialog box will appear, as seen in Figure 2-12.

---

**Figure 2-12** File Export Dialog Box

---

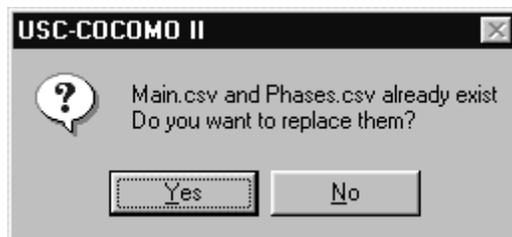


3. When you click on OK, a dialog box appears if Main.csv and Phases.csv already exist, as shown in Figure 2-13.

---

**Figure 2-13** Subsequent File Export Dialog Box

---



4. If Yes is selected, COCOMO saves two files(main.csv and phase.csv) in the chosen directory.  
If No is selected, these files will not be replaced.

## 2.10 **Save Screens**

The Save Screens option allows the user to save the image of any Cocomo window.

### To Save Screens

1. Choose Save Screens from the File menu.
2. The Save Screens dialog box will appear, as seen in Figure 2-14.
3. Follow the directions on the dialog box.

---

**Figure 2-14** Save Screens Dialog Box

---

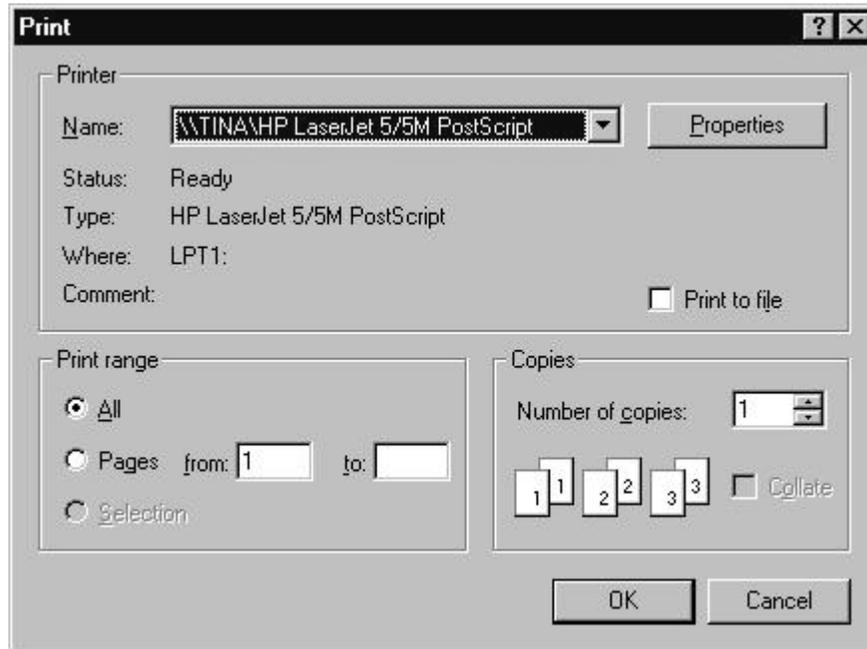


## 2.11 **Print Screen**

The Print Screen option prints the screen of the main Cocomo window.

### To Print Screen

1. Choose Print Screen from the File menu.
2. The Print Screen dialog box will appear, as seen in Figure 2-15.



3. The Name of the printer can be selected from the dropdown list. Alternatively, you can print to a file by clicking on the Print to file checkbox.
4. Properties of the printer can be set by clicking on the Properties button.
5. The Print range can be All or Pages (e.g. from 1 to 3).
6. The Number of copies can be selected by clicking the up and down arrows, or by typing a number directly.
7. Select OK when finished to print or select Cancel to not print.

## 2.11 Print Preview

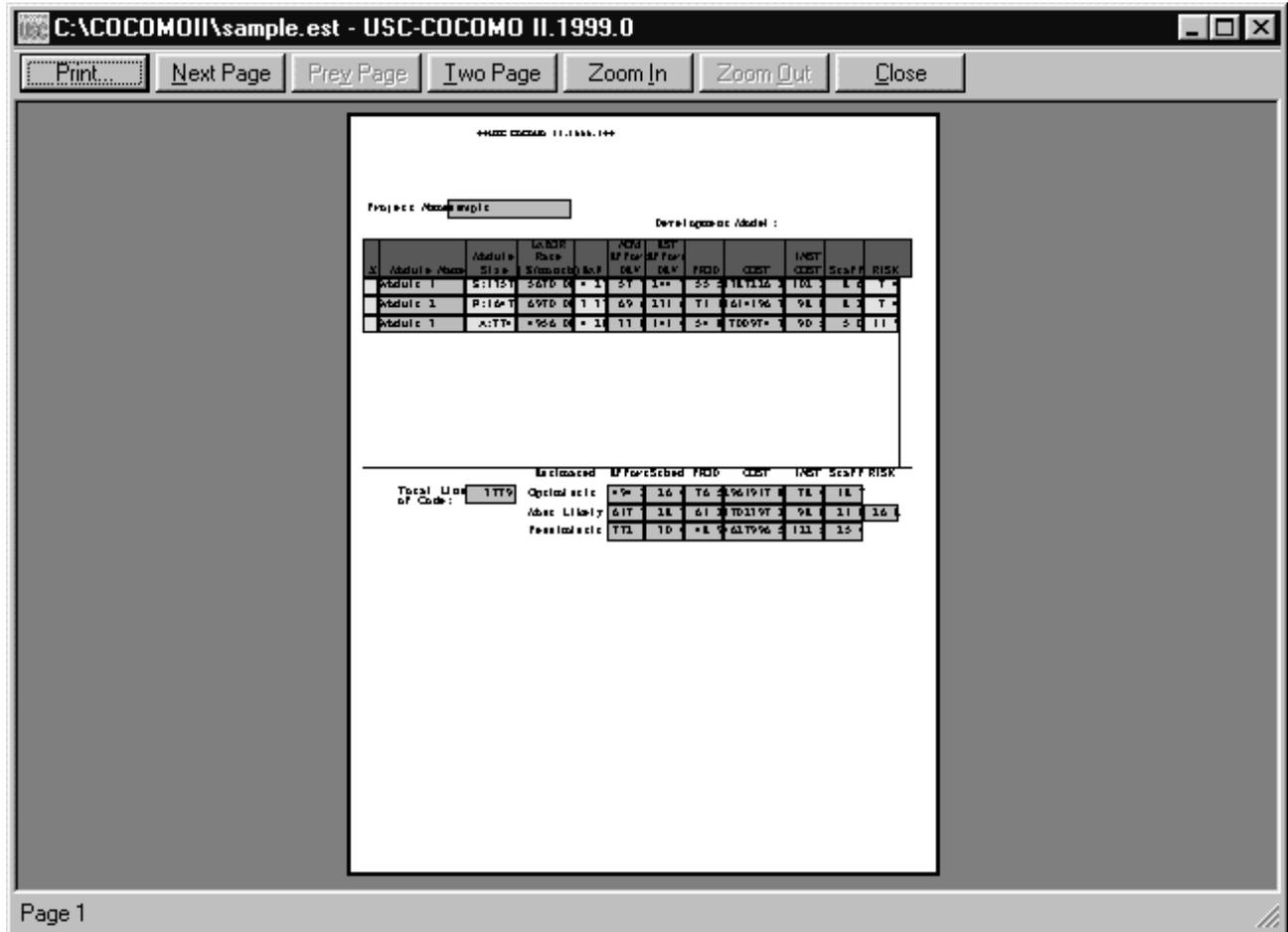
The Print Preview option displays what will be printed, if Print Screen is selected from the File menu.

### To Preview what is to be printed

1. Choose Print Preview from the File menu.
2. The Print Preview dialog box will appear, as seen in Figure 2-16.

Figure 2-16

Print Preview Dialog Box



3. Select the Print button to print.
4. The Next Page button to advance to the next Page.
5. The Zoom In button zooms in, and the Zoom Out button becomes enabled so that the user can zoom out.
6. The Close button closes the Print Preview dialog.

## 2.12 Print Setup

The Print Setup option allows the user to set up printing.

### To set up Printing

1. Choose Print Setup from the File menu.
2. The Print Setup dialog box will appear, as seen in Figure 2-17.



3. The Name of the printer can be selected from the dropdown list.
4. Properties of the printer can be set by clicking on the Properties button.
5. The Size and Source of the paper can be selected from the dropdown lists.
6. The Network button can be selected to connect to a printer on a network.
7. Select OK when finished to print or select Cancel to not print.

### 2.13 Exit

The Exit option leaves the COCOMO system.

#### To Exit COCOMO

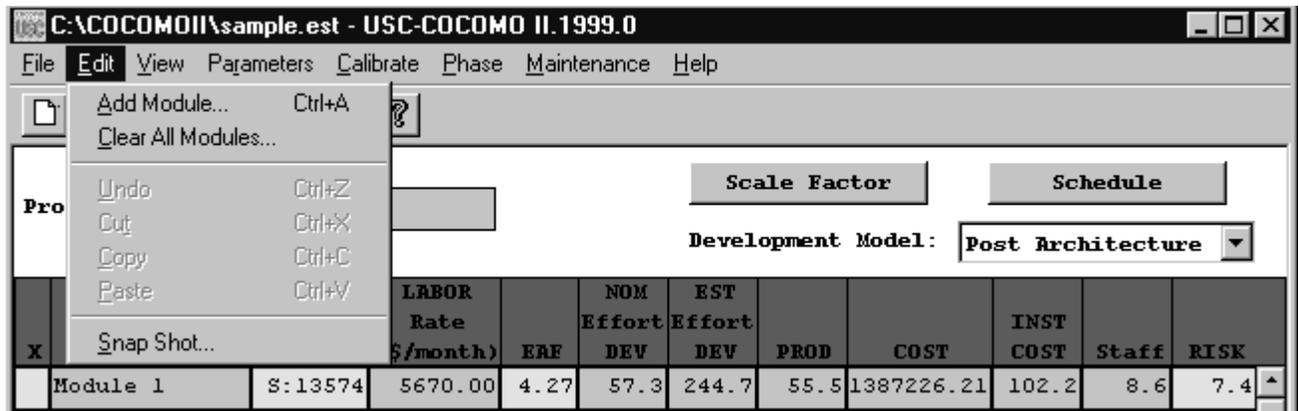
1. Choose Exit from the File menu with the mouse.
2. This causes your system to terminate the cocomo program.

## Chapter 3: Edit Menu

The Edit Menu option supplies several useful commands, which will enable you to establish a project more conveniently.

To select the Edit menu and its options, click on Edit with the mouse, then the Edit menu will appear as Figure 3-1.

**FIGURE 3-1** Edit Menu



### 3.1 Add Module

The Add Module option adds a new module to the project that is currently being worked upon by the user. This Add Module function can be done by pressing the Add Module button in the Tool bar area.

### 3.2 Clear All Module

The Clear All option erases all modules of the current project on the working window.

#### To Erase All Modules of Current Project

1. Choose Clear from the Edit menu.

During the execution of the Clear command, if some changes have occurred on the currently viewed project and have not been saved, the warning dialog box will appear as Figure 3.2.

2. If you really want to clear, click Yes. If not, click No.

3. After Clear, all modules of current project will disappear.

FIGURE 3-2

Warning Dialog Box



### 3.3 Snapshot

The Snapshot option enables users to compare the effort estimation change for a project so that he/she can decide to apply the change or not. This function makes COCOMO more convenient and powerful for software project decision analyses.

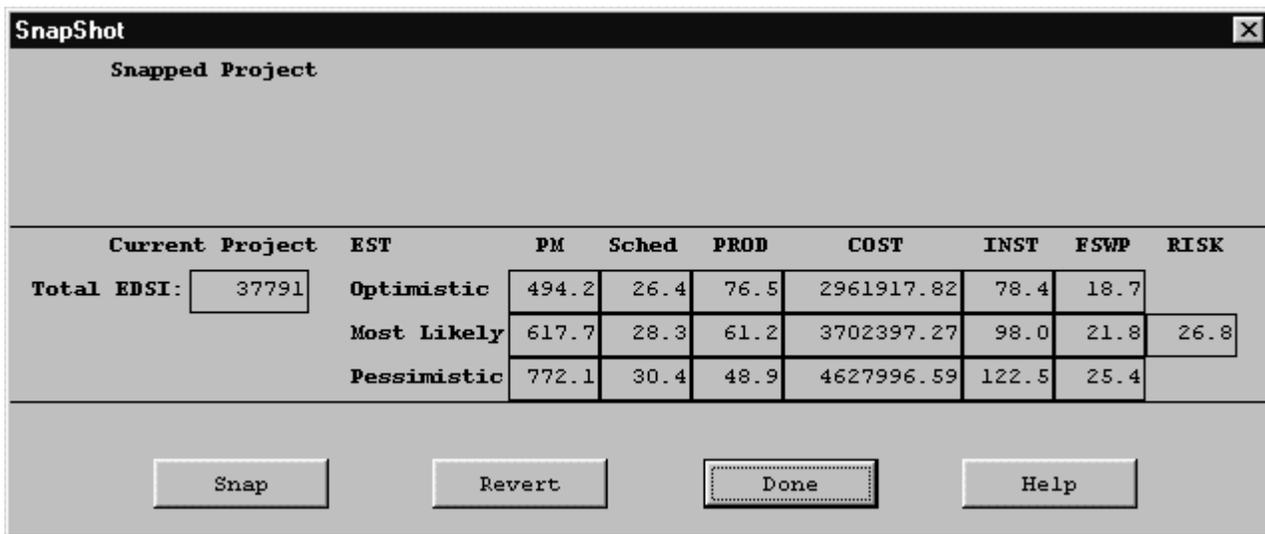
The Snapshot command stores the current set of modules, effort adjustment factors and all other data associated with a project. At a later time this data can be restored.

#### To Compare the Overall Change of a Project

1. Choose Snapshot from the Edit menu. The Snapshot dialog box will initially appear as Figure 3-3.

FIGURE 3-3

Snapshot Dialog Box-1



In the dialog box, the lower section represents the current results for the project. The upper section is previously snapped results. The current project can be snapped by clicking upon the Snap button. After completing this action the upper and lower section of the Snapshot window will contain identical information. At this point changes can be made to the current project values after clicking upon the Done button.

2. Upon completing the modification of the project values, a comparison can be made between the previously snapped project and the modified project by clicking again upon the Snapshot option in the Edit menu.
2. Now the values in the upper part of the window will likely be different from the current values, in the lower part. To restore the upper values, click on Revert. The two sets of values are interchanged.

**FIGURE 3-4 Snapshot Dialog Box-2**

SnapShot									
Snapped Project		EST	PM	Sched	PROD	COST	INST	FSWP	RISK
Total EDSI:	37791	Optimistic	494.2	26.4	76.5	2961917.82	78.4	18.7	
		Most Likely	617.7	28.3	61.2	3702397.27	98.0	21.8	26.8
		Pessimistic	772.1	30.4	48.9	4627996.59	122.5	25.4	
Current Project		EST	PM	Sched	PROD	COST	INST	FSWP	RISK
Total EDSI:	37791	Optimistic	560.0	27.4	67.5	3312882.63	87.7	20.4	
		Most Likely	700.0	29.5	54.0	4141103.29	109.6	23.8	29.3
		Pessimistic	875.0	31.6	43.2	5176379.12	137.0	27.7	

4. When finished, click the Done button.

### 3.4 Undo

The Undo option retracts the previous cut or paste done on a module.

#### To Retract Previous Cut/Paste for a Module

1. Choose Undo from the Edit menu with the mouse.
2. The changed module will go back to its previous status.

### 3.5 Cut

The Cut option copies a module into the cut buffer and removes it from the current project. The cut module can be used for Paste.

### **To Cut a Module and Remove It From the CLEF**

1. Check the module which is to be cut. The Check boxes for modules are located in the leftmost column of the CLEF area. Place the mouse in the box just to the left of the module name, and click.
2. Choose Cut from the Edit menu with the mouse.
3. The cut module disappears.

## **3.6 Copy**

The Copy option copies a module. The copied module can be used for Paste.

### **To Copy a Module**

1. Check the module which is to be copied. The Check boxes for modules are located in the leftmost column of the CLEF area.
2. Choose Copy from the Edit menu with the mouse.
3. The cross sign in the check box disappears.

## **3.7 Paste**

The Paste option pastes a previously copied or cut module in the CLEF.

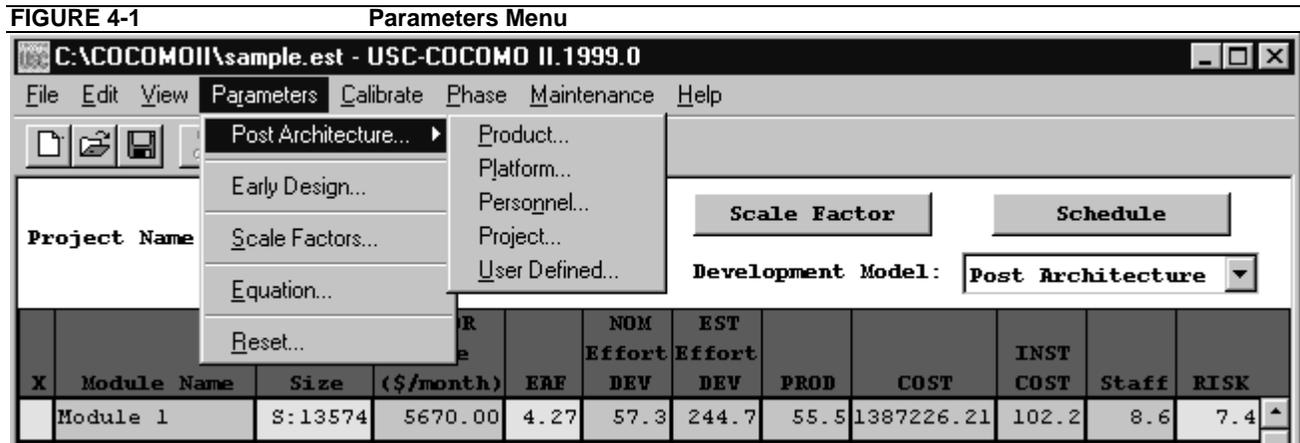
### **To Paste a Previously Copied or Cut Module**

1. Check the module above which the previously copied or cut module is to be pasted. The Check boxes for modules are located in the leftmost column of CLEF area.
2. Choose Paste from the Edit menu with the mouse.
3. The pasted module appears at the checked position, and the modules lower than it were pushed one row down.
4. If there is no module checked, the Paste will attach the previously copied or cut module at the end.

## Chapter 4: Parameters Menu

The Parameters menu option will enable you to look at, or change the values of effort adjustment factors, scale factors and effort/schedule estimating equations factors for the current project.

To choose the Parameters menu and its options, click on Parameters with the mouse. The Parameters menu will appear as Figure 4-1.



### 4.1 Post Architecture Model

#### 4.1.1 Product

The Product option displays five cost drivers: RELY, DATA, DOCU, CPLX, and RUSE and their corresponding ratings and multiplier values. Select Product from the Parameters menu with the mouse. The Product Dialog Box will appear as Figure 4-2.

**FIGURE 4-2** Product Dialog Box

	VLO	LO	NOM	HI	VHI	XHI
RELY	0.82	0.92	1.00	1.10	1.26	XXXX
DATA	XXXX	0.90	1.00	1.14	1.28	XXXX
DOCU	0.81	0.91	1.00	1.11	1.23	XXXX
CPLX	0.73	0.87	1.00	1.17	1.34	1.74
RUSE	XXXX	0.95	1.00	1.07	1.15	1.24

OK Cancel Help

To modify these values, go straight to those edit boxes and type new values. When finished with the modification, click the OK button.

#### 4.1.2 Platform

The Platform option displays three cost drivers: TIME, STOR and PVOL, and their corresponding ratings and multiplier values.

Select Platform from the Parameters menu with the mouse. The Platform Dialog Box will appear as Figure 4-3.

**FIGURE 4-3** Platform Dialog Box

	VLO	LO	NOM	HI	VHI	XHI
TIME	XXXX	XXXX	1.00	1.11	1.29	1.63
STOR	XXXX	XXXX	1.00	1.05	1.17	1.46
PVOL	XXXX	0.87	1.00	1.15	1.30	XXXX

OK Cancel Help

To modify these values, go straight to those edit boxes and type new values. When finished with the modification, click the OK button.

### 4.1.3 Personnel

The Personnel option displays six cost drivers: ACAP, AEXP, PCAP, PEXP, LEXP, and PCON and their corresponding ratings and multiplier values.

Select Personnel from the Parameters menu with the mouse. The Personnel Dialog Box will appear as Figure 4-4.

**FIGURE 4-4 Personnel Dialog Box**

	VLO	LO	NOM	HI	VHI	XHI
ACAP	1.42	1.19	1.00	0.85	0.71	XXXX
AEXP	1.22	1.10	1.00	0.88	0.81	XXXX
PCAP	1.34	1.15	1.00	0.88	0.76	XXXX
PEXP	1.19	1.09	1.00	0.91	0.85	XXXX
LTEX	1.20	1.09	1.00	0.91	0.84	XXXX
PCON	1.29	1.12	1.00	0.90	0.81	XXXX

Buttons: OK, Cancel, Help

To modify these values, go straight to those edit boxes and type new values. When finished with the modification, click the OK button.

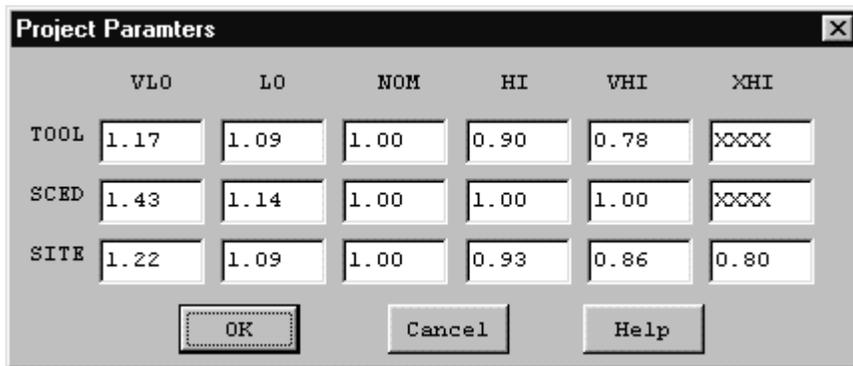
### 4.1.4 Project

The Project option displays three cost drivers: TOOL, SCED, and SITE and their corresponding ratings and multiplier values.

Select Project from the Parameters menu with the mouse. The Project Dialog Box will appear as Figure 4-5.

**FIGURE 4-5 Project Dialog Box**

To modify these values, go straight to those edit boxes and type new values. When finished with the



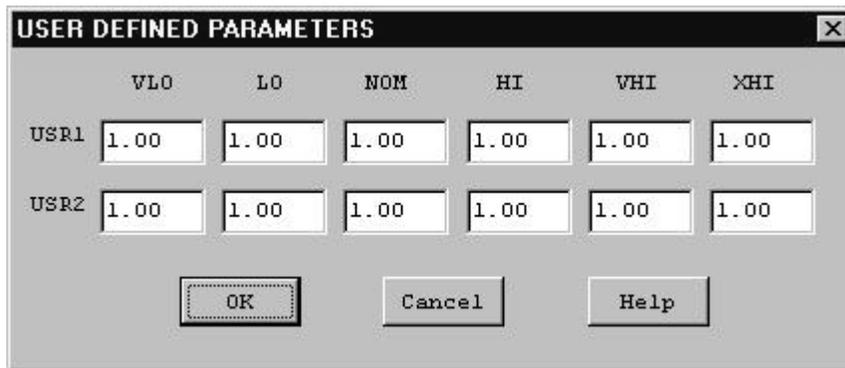
modification, click the OK button.

#### 4.1.5 User Defined EAF

The User Defined EAF option displays two cost driver: USR1 and USR2, and their corresponding ratings and multipliers.

Select User EAF from the Parameters menu with the mouse. The User EAF Dialog Box will appear as Figure 4-6.

**FIGURE 4-6 User Defined EAF Dialog Box**



To modify these values, go straight to those edit boxes and type new values. When finished with the modification, click the OK button.

## 4.2 Early Design Model

FIGURE 4-7 Early Design Model - EAF Dialog Box

	XLO	VLO	LO	NOM	HI	VHI	XHI
RCPX	0.73	0.81	0.98	1.00	1.30	1.74	2.38
RUSE	XXXX	XXXX	0.95	1.00	1.07	1.15	1.24
PDIF	XXXX	XXXX	0.87	1.00	1.29	1.81	2.61
PERS	2.12	1.62	1.26	1.00	0.83	0.63	0.50
PREX	1.59	1.33	1.12	1.00	0.87	0.71	0.62
FCIL	1.43	1.30	1.10	1.00	0.87	0.73	0.62
SCED	XXXX	1.43	1.14	1.00	1.00	1.00	XXXX
USR1	XXXX	1.00	1.00	1.00	1.00	1.00	XXXX
USR2	XXXX	1.00	1.00	1.00	1.00	1.00	XXXX

## 4.3 Scale Factors

The Scale Factors option displays five development attributes: PREC, FLEX, RESL, TEAM and PMAT, and their corresponding ratings and values.

Select Scale Factors from the Parameters menu with the mouse. The Scale Factor Dialog Box will appear as Figure 4-8.

FIGURE 4-8

Scale Factors Dialog Box

	VLO	LO	NOM	HI	VHI	XHI
PREC	6.20	4.96	3.72	2.48	1.24	0.00
FLEX	5.07	4.05	3.04	2.03	1.01	0.00
RESL	7.07	5.65	4.24	2.83	1.41	0.00
TEAM	5.48	4.38	3.29	2.19	1.10	0.00
PMAT	7.80	6.24	4.68	3.12	1.56	0.00

OK Cancel Help

To modify these values, go straight to those edit boxes and type new values. When finished with the modification, click OK button.

#### 4.4 Equation

The Equation option displays effort and schedule equations.

Select Equation from the Parameters menu with the mouse. The Equation Dialog Box will appear as Figure 4-9.

The dialog box, titled "Equation Parameters", contains three sections for defining equations:

- Exponent Equation:**  $B = 0.91 + 0.01 (SF1 + \dots + SF5)$
- Effort Equation:**  $PM = EM1 * \dots * EM17 * 2.94 * (Size)^B + (ASLOC * (AT/100) / ATPROD)$
- Schedule Equation:**  $TDEV = [ 3.67 * PM^{( 0.28 + 0.2 * (B - 0.91))}] * (SCED\$/100)$

At the bottom of the dialog are three buttons: "OK", "Cancel", and "Help".

To modify these values, go straight to those edit boxes and type new values. When finished with the modification, click the OK button.

#### 4.5 *Reset*

The Reset option resets the values of multiplying factors and effort/schedule estimating equations of the current project back to the COCOMO default values.

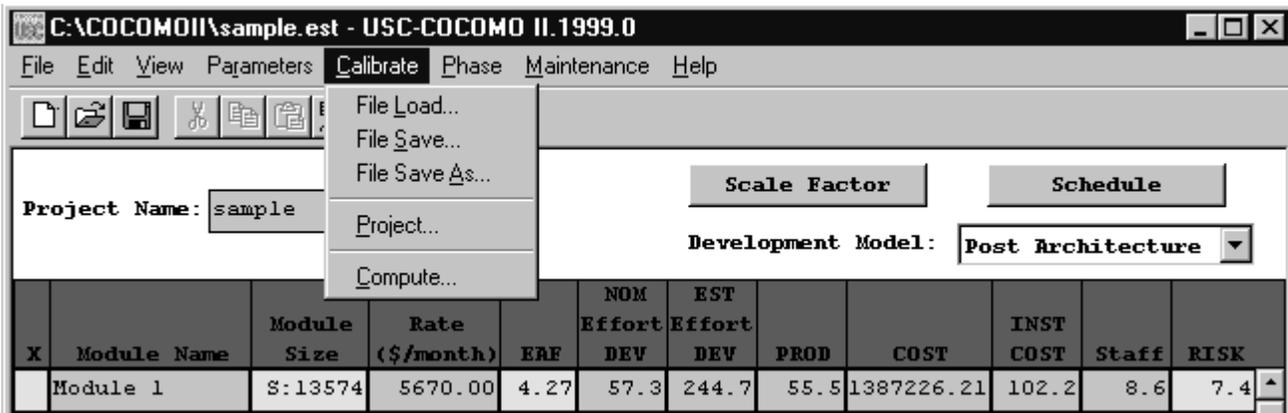
Select Reset from the Parameters menu with mouse. The command will be executed directly, and there is no warning message for users. After the RESET, the values of all multiplying factors and effort estimating equations of current project will be changed to the COCOMO default values.

## Chapter 5: Calibrate Menu

COCOMOII now has the ability to archive your own software project data. Using this data, COCOMOII will compute various coefficients and exponents involved in the effort and schedule equations. This will make your COCOMOII estimates even more reliable.

Each software project to be archived is described as a complete COCOMOII project. It may include multiple modules, each with their own SLOC estimate and EAF factors. In addition, a software project consists of a name, date/time, actual effort and actual schedule. The actual effort and actual schedule must be supplied by the COCOMOII user. Entering revised values for effort and schedule are always possible. Effort is given in units of person/months. Schedule is given in units of months. Both assume the COCOMO hours per month value of 152.

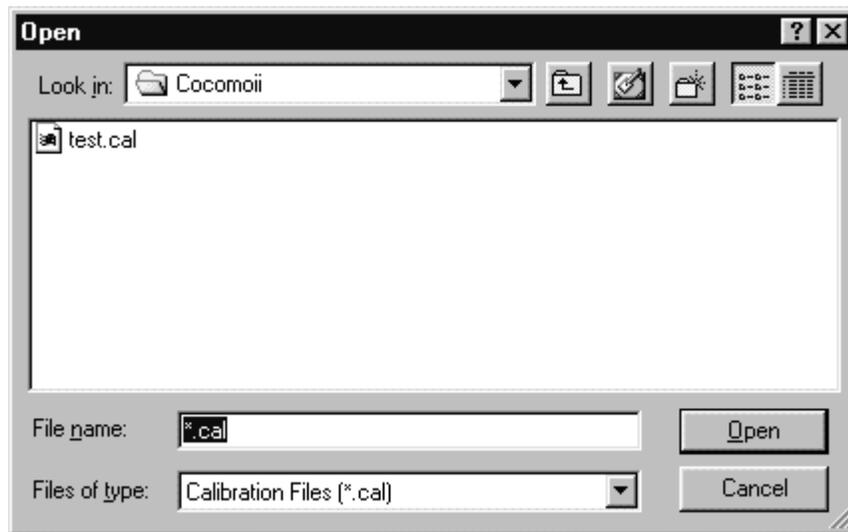
**FIGURE 5-1** Calibrate Menu



### 5.1 File Load

The Calibrate File Load option is used to retrieve a calibration project file as well as loading all project data on the working project window(Figure5-2).

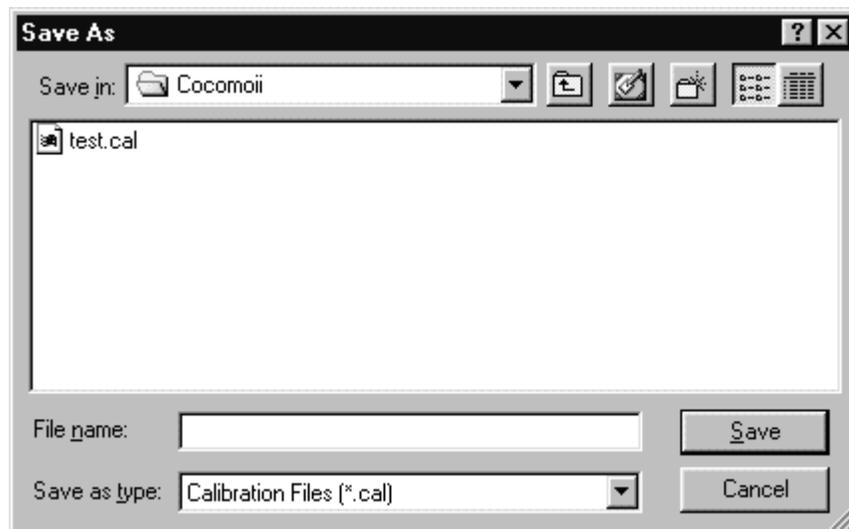
**FIGURE 5-2** Load Calibration Dialog Box



## 5.2 File Save

The Calibrate File Save command saves the current calibration data in the file whose name was previously identified using File Save As.

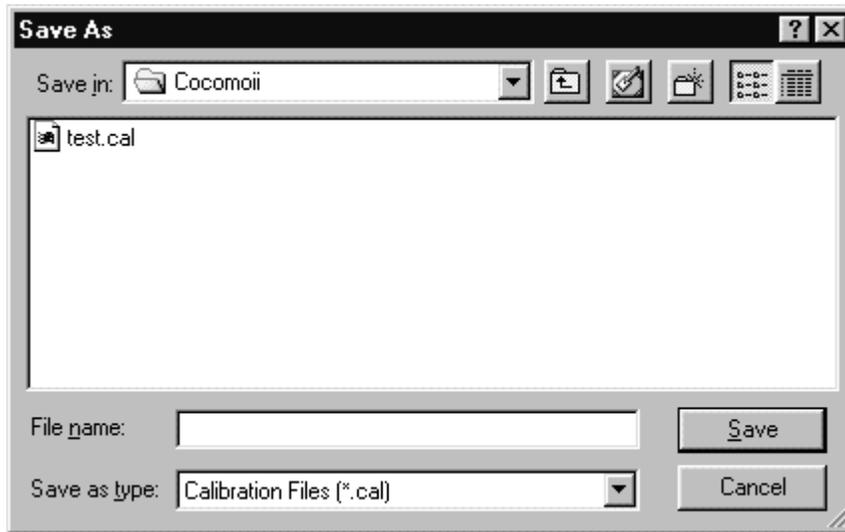
**FIGURE 5-3** Save Calibration Dialog Box



### 5.3 File Save As

The Calibrate File Save As command stores the current calibration data as a \*.cal file, which has a different file name from the current file. This command works precisely the same as the File Save As for \*.est and \*.mod files (see Figure 5-4).

FIGURE 5-4 Save As Calibration Dialog Box

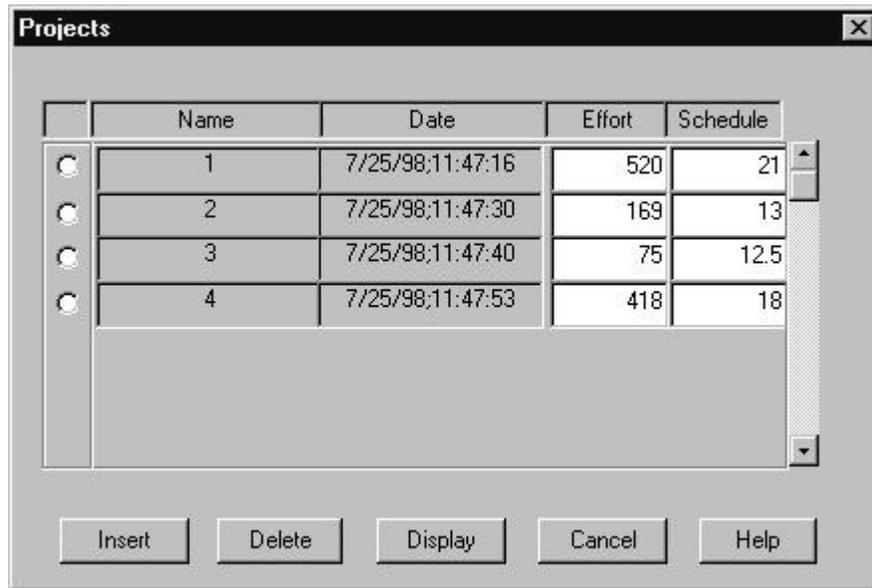


### 5.4 Project

A window appears (shown in Figure 5-5) which displays the archived project data.

- To remove the window, click on Cancel.
- To delete an existing entry, first place an x at the leftmost end of the row and click on Delete. A warning box appears as shown in Figure 5-6.
- To display the entire set of values for an archived project, click on Display. Since the display of an archived project eliminates the display of any existing CLEF data, a warning message appears as shown in Figure 5-7.
- To insert a new archived project from the CLEF, click on Insert.

**FIGURE 5-5** Projects Dialog Box



**FIGURE 5-6** Delete Warning Dialog

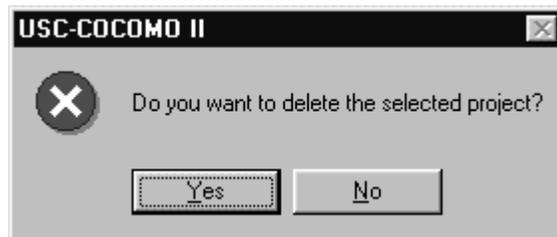
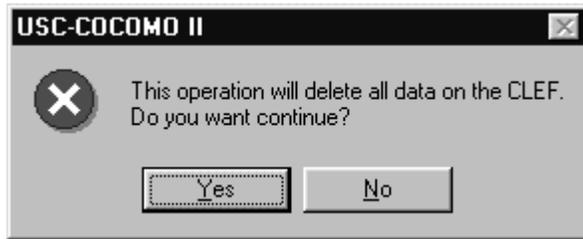


FIGURE 5-7

Display Warning Dialog

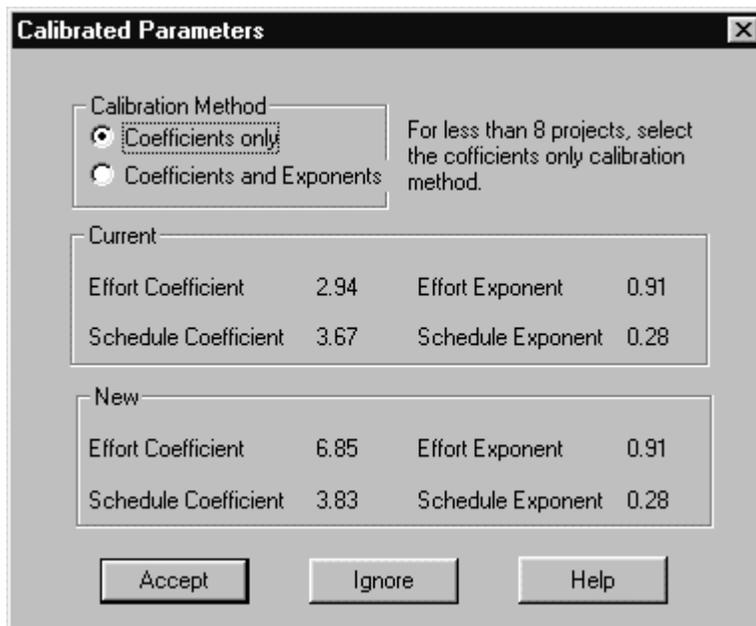


## 5.5 Compute

This command takes all of the data that has been archived and uses it to compute new constant and/or exponent values for the effort equation and for the schedule equation. There are two options to calibrate equation parameters. One is the Constant Term and the other one is Development Mode. Those two options are explained below in detail. They are displayed in this window and compared to the values currently used by COCOMOII. To get COCOMOII to use these values, click on Accept, as shown Figure 5-8.

FIGURE 5-8

Compute Dialog Box



## Chapter 6: Phase Distribution

The Phase Distribution is one of the menu selections in the menu bar that can be accessed by clicking upon Phase Distribution in the main menu. Its function is to display a breakdown of the software effort and schedule into the phases of the waterfall development cycle. These phases are plans & requirements, design, programming and integration & test. These phases are described as follows:

**Plans & Requirements** - In this phase, a statement for the required functions, interfaces and performance is created. These expectations are used to define the capabilities of the software product as expressed by representatives of all interested parties.

**Product Design** - In this phase, a hardware/software architecture, control structure, and data structure for the product are defined. A draft of the user's manual and test plans are also created during this phase.

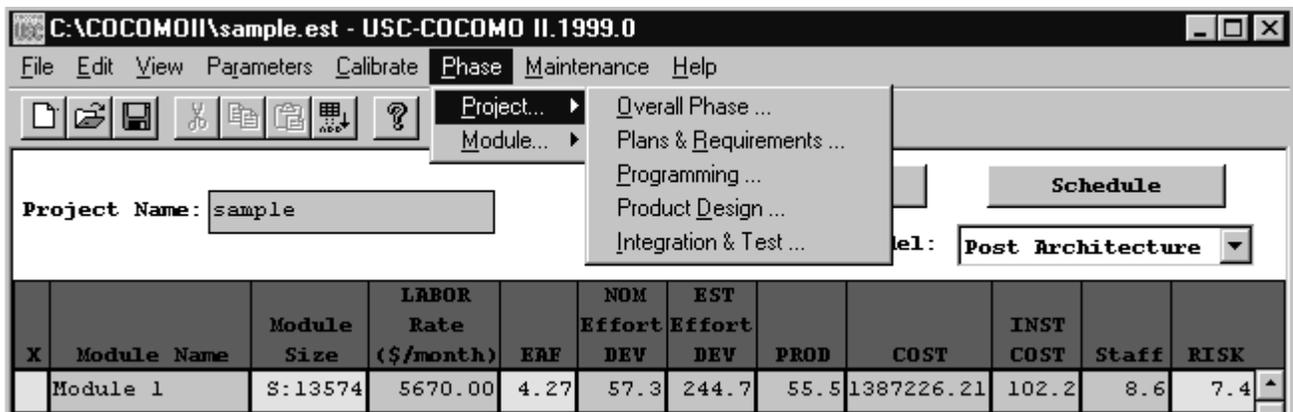
**Programming** - In this phase, the design of the previous phase is implemented in the creation of complete sets of software components.

**Integration & Test** - In this phase, the various software components are brought together in order to achieve a properly functioning software product composed of loosely coupled modules. The requirements as defined in the first phase are used to determine the fitness of the delivered product.

The phase distribution menu has two selections: project phase distribution and module phase distribution. The project phase distribution allows the user to view the development phases for the entire project all together or individually. The module phase distribution allows the user to view the development phases for a particular module either all together or individually. These two variations of phase distribution are discussed further in this chapter under sections 6.1 and 6.2 in this chapter.

**NOTE:** These phase distribution estimates are retained from the COCOMO81 model, which assumed a waterfall (sequentially phased) process model. If your project's process model is not a waterfall, these phase distributions are either inapplicable or need to be reinterpreted.

**FIGURE 6-1** Phase Distribution Sub-menu



## 6.1 *Project Phase Distribution*

In order to view the phase distribution of an entire project, the user can click on the Project Phase Distribution button under the Phase Distribution menu (see FIGURE 6-1). Four formats for viewing will appear in another menu: overall phase, plan & requirements, programming, and integration & test. Each of these menu selections will be discussed in sections 6.1.1 - 6.1.4, respectively. The phase distribution of plan & requirements, programming and integration & test are broken down into sub-phases. These phases include: requirements analysis, product design, programming, test planning, verification & validation, project office, CM/QA, and manuals. For each of these sub-phases the percentage of the phase, the estimated effort, the estimated schedule, and the estimated FSWP is displayed. A description of each of these sub-phases follows:

Requirements analysis: Determination, specification review and update of software functional, performance, interface, and verification requirements.

Product Design: Determination, specification, review and update of hardware-software architecture, program design, and database design.

Programming: Detailed design, code, unit test, and integration of individual computer program components. Includes programming personnel planning, tool acquisitions, database development, component level documentation, and intermediate level programming management.

Test Planning: Specification, review, and update of product test and acceptance test plans. Acquisition of associated test drivers, test tools, and test data.

Verification & Validation(V&V): Performance of independent requirements validation, design V&V, product test, and acceptance test. Acquisition of requirements and design V&V tools. "Are we building the product right?" and "are we building the right product?"

Project Office Functions: Project level management functions. Includes project level planning and control, contract and subcontract management, and customer interface.

Configuration Management and Quality Assurance (CM/QA): Configuration management includes product identification, change control, status accounting, operation of program support library, development and monitoring of end item acceptance plan. Quality assurance includes development and monitoring of project standards, and technical audits of software products and processes.

Manuals: Development and update of users' manuals, operators' manuals and maintenance manuals.

### 6.1.1 Overall Project Phase Distribution

The overall phase distribution allows the user to view an entire project's estimated effort, schedule and number of personnel needed for phase completion. Upon clicking on "Overall Phase," a window will be displayed showing the phase breakdown of the current project in COCOMO (see FIGURE 6-2). This window displays the project name, project SLOC, and the total estimated effort for the project. Looking at FIGURE 6-1, this information can be seen in the upper left corner of the window.

**FIGURE 6-2 Phase Distribution window displaying a sample project's overall phase distribution**

The screenshot shows a window titled "Phase Distribution - Project Overall" with a close button in the top right corner. The window displays the following text:

```

Overall Phase Distribution
=====
PROJECT                sample
SLOC                   37791
TOTAL EFFORT           700.036 Person Months
=====

```

	PCNT	EFFORT (PM)	PCNT	SCHEDULE	FSWP
Plans And Requirements	7.000	49.002	20.121	5.928	8.267
Product Design	17.000	119.006	26.060	7.678	15.501
Programming	57.819	404.754	47.759	14.070	28.767
- Detailed Design	24.940	174.587	----	----	----
- Code and Unit Test	32.879	230.167	----	----	----
Integration and Test	25.181	176.276	26.181	7.713	22.854

At the bottom of the window, there are two buttons: "OK" and "Help".

In addition, each phase of the project's development cycle is represented by the estimated effort, the estimated schedule and the estimated number of personnel needed for phase completion. Again looking at FIGURE 6-2, the information has been separated into columns. The first column displays the phase name. The second column displays the percentage that the corresponding phase takes in the estimated effort. The third column displays the estimated effort for each phase. The fourth column displays the percentage of the estimated schedule that is dedicated to the corresponding phase's completion. The fifth column displays the estimated schedule for phase completion. And the last column displays the estimated number of personnel needed for phase completion (FSWP).

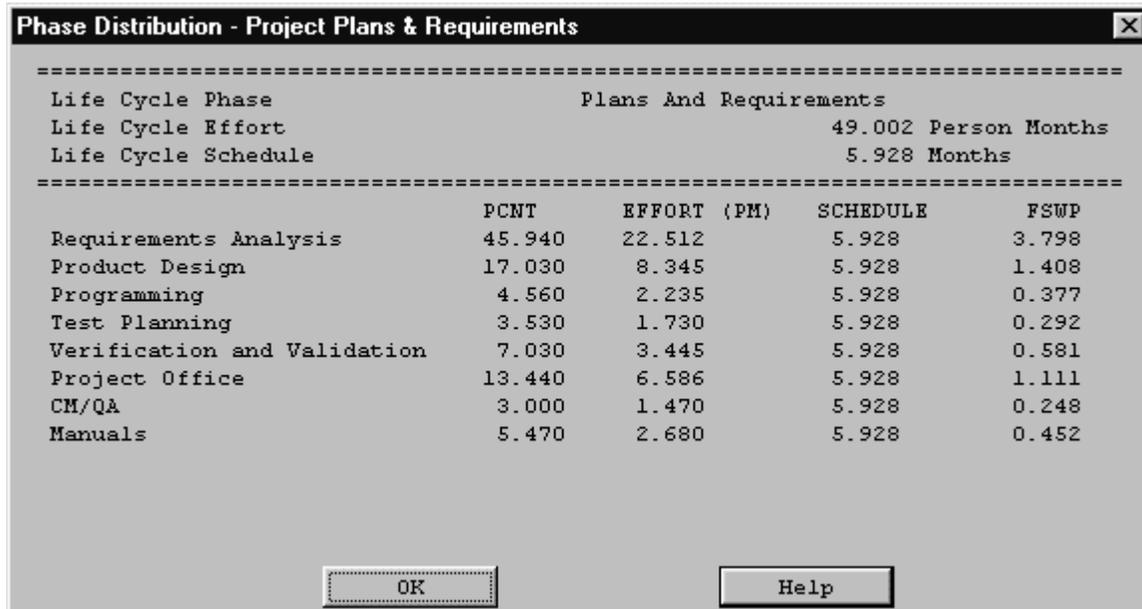
Note: The programming phase has been broken down into two additional phases: "Detailed Design" and "Code and Unit Test." The detailed design is a follow-up to the product design phase. In this sub phase, those points developed in the product design are elaborated to a point necessary to breakdown agreed functions into units necessary for coding. The code and unit test sub-phases house the actual coding effort of the individual units of code. The testing of these units (upon completion) is also encompassed within this sub phase.

### 6.1.2 Plans and Requirements Project Phase Distribution

The plans and requirements phase distribution allows the user to view the components of this particular phase. When the Plans and Requirements distribution is chosen from the Project Phase distribution menu, the window shown in FIGURE 6-3 is displayed. This window displays the following information: project name, the total project SLOC, the total estimated project effort, and the total estimated project schedule. In addition the window displays the estimated effort for the activities of requirements analysis, product design, programming, test planning, verification & validation, project office, CM/QA, and manuals. These activity estimates are accompanied with a percentage of the phase effort that they encompass, the estimated effort, schedule and FSWP for the activity's completion as shown in FIGURE 6-3. To exit from this window click the OK button.

FIGURE 6-3

Plans and Requirements Phase window for the overall project

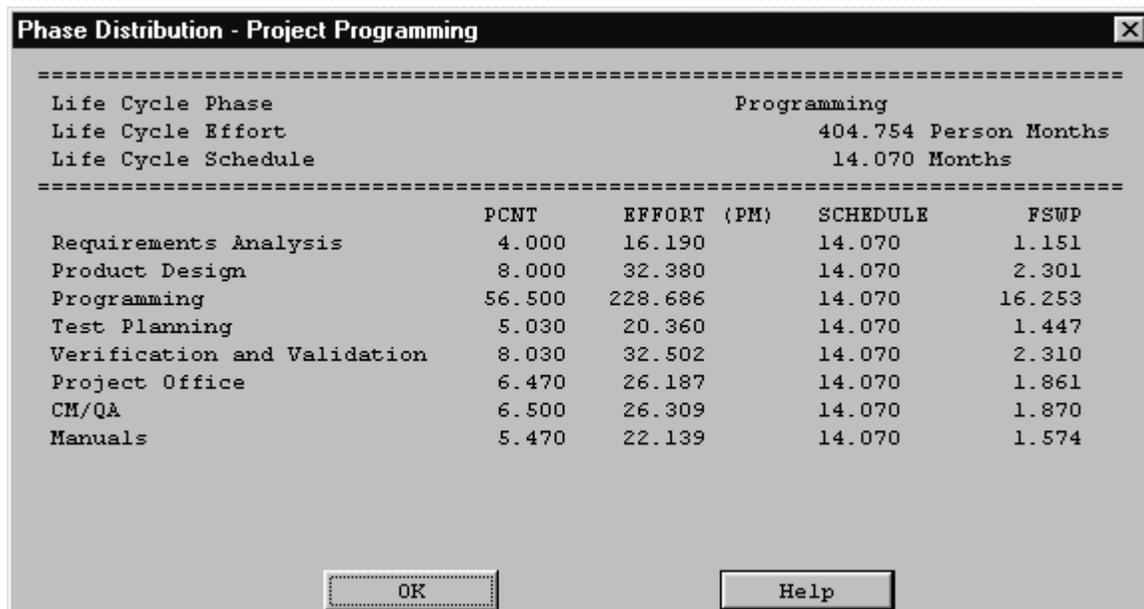


### 6.1.3 Programming Project Phase

The programming phase distribution allows the user to view the components of this particular phase. When the Programming distribution is chosen from the Project Phase distribution menu, the window shown in FIGURE 6-4 is displayed. This window displays the following information: project name, the total project SLOC, the total estimated project effort, and the total estimated project schedule. In addition the window displays the estimated effort for the activities of requirements analysis, product design, programming, test planning, verification & validation, project office, CM/QA, and manuals. These activities are accompanied with a percentage of the phase effort that they encompass, the estimated effort, schedule and FSWP for the activity's completion as shown in FIGURE 6-4. To exit from this window click the OK button.

FIGURE 6-4

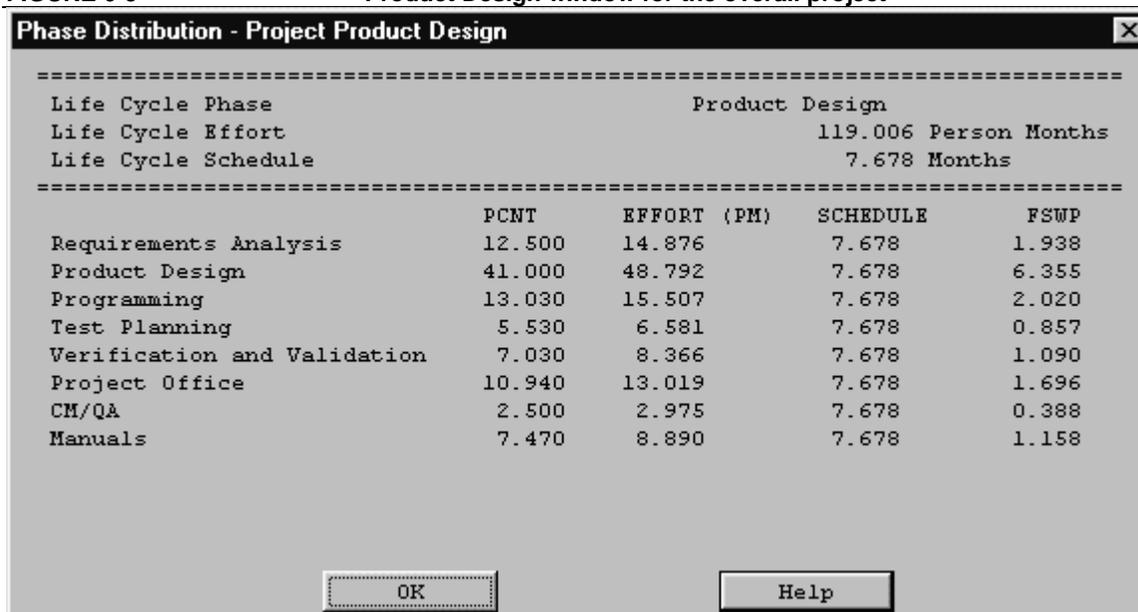
Programming Phase window for the overall project



### 6.1.4 Product Design Phase

The product design phase distribution allows the user to view the components of this particular phase. When the Product Design distribution is chosen from the Project Phase distribution menu, the window shown in FIGURE 6-5 is displayed. This window displays the following information: project name, the total project SLOC, the total estimated project effort, and the total estimated project schedule. In addition the window displays the estimated effort for the activities of requirements analysis, product design, programming, test planning, verification & validation, project office, CM/QA, and manuals. These activity estimates are accompanied with a percentage of the phase effort that they encompass, the estimated effort, schedule and FSWP for the activity's completion as shown in FIGURE 6-5. To exit from this window click the OK button.

FIGURE 6-5 Product Design window for the overall project



The screenshot shows a window titled "Phase Distribution - Project Product Design" with a close button (X) in the top right corner. The window content is as follows:

```
=====
Life Cycle Phase                Product Design
Life Cycle Effort                119.006 Person Months
Life Cycle Schedule              7.678 Months
=====
```

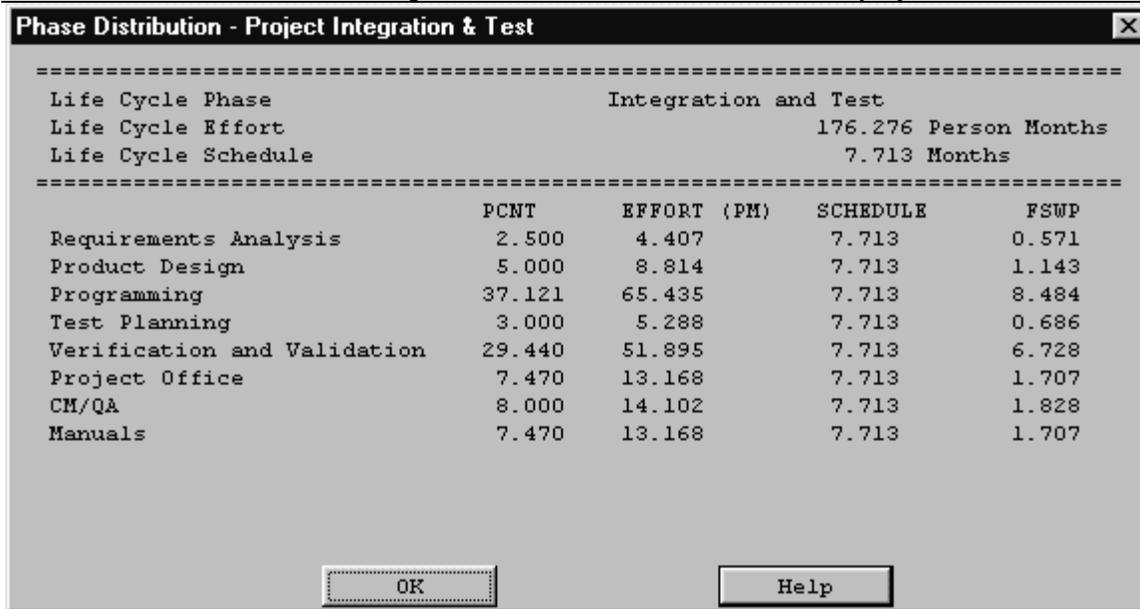
	PCNT	EFFORT (PM)	SCHEDULE	FSWP
Requirements Analysis	12.500	14.876	7.678	1.938
Product Design	41.000	48.792	7.678	6.355
Programming	13.030	15.507	7.678	2.020
Test Planning	5.530	6.581	7.678	0.857
Verification and Validation	7.030	8.366	7.678	1.090
Project Office	10.940	13.019	7.678	1.696
CM/QA	2.500	2.975	7.678	0.388
Manuals	7.470	8.890	7.678	1.158

At the bottom of the window, there are two buttons: "OK" and "Help".

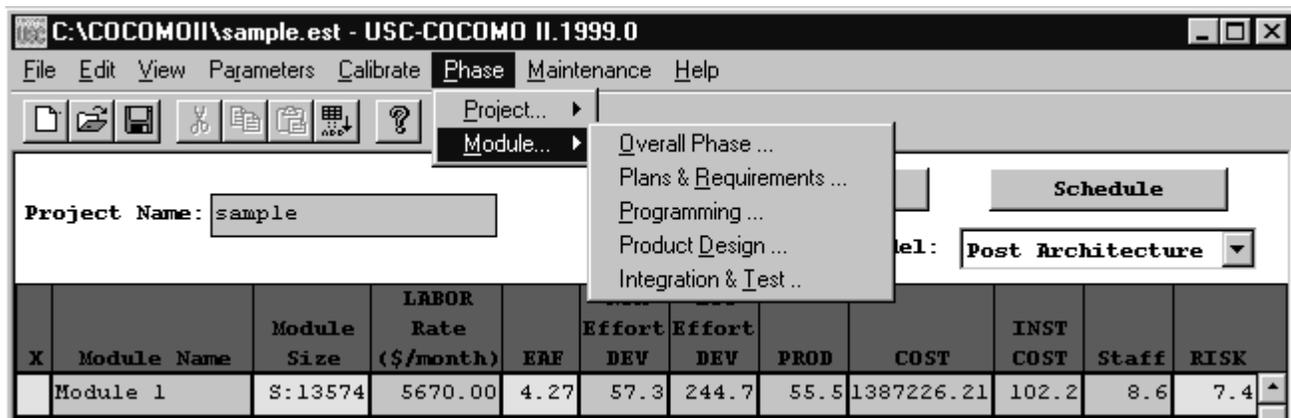
### 6.1.5 Integration and Test Project Phase

The integration & test phase distribution allows the user to view the components of this particular phase. When the Integration and Test distribution is chosen from the Project Phase distribution menu, the window shown in FIGURE 6-6 is displayed. This window displays the following information: project name, the total project SLOC, the total estimated project effort, and the total estimated project schedule. In addition the window displays the estimated effort for the activities of requirements analysis, product design, programming, test planning, verification & validation, project office, CM/QA, and manuals. These activity estimates are accompanied with a percentage of the phase effort that they encompass the estimated effort, schedule and FSWP for the activity's completion as shown in FIGURE 6-6. To exit from this window click the OK button.

**FIGURE 6-6** Integration and Test window for the overall project



**FIGURE 6-7** Phase Distribution Module Sub-menu



## 6.2 Module Phase Distribution

Four formats for viewing will appear in another menu: overall phase, plan & requirements, programming, and integration & test (see FIGURE 6-7). Each of these menu selections will be discussed in sections 6.2.1 - 6.2.4, respectively. The phase distribution of plan & requirements, programming and integration & test are broken down into activities. These activities include: requirements analysis, product design, programming, test planning, verification & validation, Project office, CM/QA, and manuals. For each of these activities, the percentage of the phase, the estimated effort, the estimated schedule, and the estimated FSWP is displayed. A description of each of these activities follows:

Requirements analysis: Determination, specification review and update of software functional, performance, interface, and verification requirements.

Product Design: Determination, specification, review and update of hardware-software architecture, program design, and database design.

Programming: Detailed design, code, unit test, and integration of individual computer program components. Includes programming personnel planning, tool acquisitions, database development, component level documentation, and intermediate level programming management.

Test Planning: Specification, review, and update of product test and acceptance test plans. Acquisition of associated test drivers, test tools, and test data.

Verification & Validation(V&V): Performance of independent requirements validation, design V&V, product test, and acceptance test. Acquisition of requirements and design V&V tools. "Are we building the product right?" and "are we building the right product?"

Project Office Functions: Project level management functions. Includes Project level planning and control, contract and subcontract management, and customer interface.

Configuration Management and Quality Assurance (CM/QA): Configuration management includes product identification, change control, status accounting, operation of program support library, development and monitoring of end item acceptance plan. Quality assurance includes development and monitoring of Module standards, and technical audits of software products and processes.

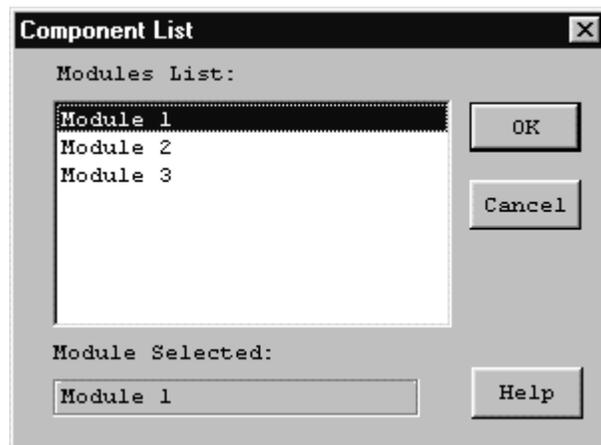
Manuals: Development and update of users' manuals, operators' manuals and maintenance manuals.

In order to view the phase distribution of an entire Module, the user can click on the Module Phase Distribution button under the Phase Distribution menu. When choosing any of the views of phase distribution, you will be confronted with a module selection window (see FIGURE 6-8). At this point, you may choose which module is to be viewed by clicking on the desired module name, which will be highlighted after the click. Click the OK button in order to initiate phase distribution of the chosen module.

---

**FIGURE 6-8** **Module selection window**

---



## 6.2.1 Overall Module Phase Distribution

The overall phase distribution allows the user to view an entire Module's estimated effort, schedule and number of personnel needed for phase completion. Upon clicking on "Overall Phase," a window will be displayed showing the phase breakdown four formats for viewing will appear in another menu: overall phase, plan & requirements, programming, and integration & test (see FIGURE 6-9). To exit from this window click the OK button.

**FIGURE 6-9** Phase Distribution window displaying a sample Module's overall phase distribution

Overall Phase Distribution					
=====					
MODULE	Module 1				
SLOC	13574				
TOTAL EFFORT	348.682 Person Months				
=====					
	PCNT	EFFORT (PM)	PCNT	SCHEDULE	FSWP
Plans And Requirements	7.000	24.408	20.121	5.928	4.118
Product Design	17.000	59.276	26.060	7.678	7.721
Programming	57.819	201.604	47.759	14.070	14.329
- Detailed Design	24.940	86.960	----	----	----
- Code and Unit Test	32.879	114.644	----	----	----
Integration and Test	25.181	87.801	26.181	7.713	11.383

In addition, each phase of the Module's development cycle is represented by the estimated effort, the estimated schedule and the estimated number of personnel needed for phase completion. Again looking at FIGURE 6-9, the information has been separated into columns. The first column displays the phase name. The second column displays the percentage that the corresponding phase takes in the estimated effort. The third column displays the estimated effort for each phase. The fourth column displays the percentage of the estimated schedule that is dedicated to the corresponding phase's completion. The fifth column displays the estimated schedule for phase completion. And the last column displays the estimated number of personnel needed for phase completion (FSWP).

Note: The programming phase has been broken down into two additional phases: "Detailed Design" and "Code and Unit Test." The detailed design is a follow-up to the product design phase. In this sub phase, those points developed in the product design are elaborated to a point necessary to breakdown agreed functions into units necessary for coding. The code and unit test sub phase houses the actually coding effort of the individual units of code. The testing of these units (upon completion) is also encompassed within this sub phase.

## 6.2.2 Plans and Requirements Module Phase Distribution

The plans and requirements phase distribution allows the user to view the components of this particular phase. When the Plans and Requirements distribution is chosen from the Module Phase distribution menu, the window shown in FIGURE 6-10 is displayed. This window displays the following

information: Module name, the total Module SLOC, the total estimated Module effort, and the total estimated Module schedule. In addition the window displays the activities requirements analysis, product design, programming, test planning, verification & validation, Project office, CM/QA, and manuals. These activity estimates are accompanied with a percentage of the phase effort that they encompass, the estimated effort, schedule and FSWP for the activity's completion as shown in FIGURE 6-10. To exit from this window click the OK button.

**FIGURE 6-10 Plans and Requirements Phase window for the overall Module**

Life Cycle Phase	Plans And Requirements			
Life Cycle Effort	24.408 Person Months			
Life Cycle Schedule	5.928 Months			
	PCNT	EFFORT (PM)	SCHEDULE	FSWP
Requirements Analysis	45.940	11.213	5.928	1.892
Product Design	17.030	4.157	5.928	0.701
Programming	4.560	1.113	5.928	0.188
Test Planning	3.530	0.862	5.928	0.145
Verification and Validation	7.030	1.716	5.928	0.289
Project Office	13.440	3.280	5.928	0.553
CM/QA	3.000	0.732	5.928	0.124
Manuals	5.470	1.335	5.928	0.225

### 6.2.3 Programming Module Phase Distribution

The programming phase distribution allows the user to view the components of this particular phase. When the Programming distribution is chosen from the Module Phase distribution menu, the window shown in FIGURE 6-11 is displayed. This window displays the following information: Module name, the total Module SLOC, the total estimated Module effort, and the total estimated Module schedule. In addition the window displays the activity's requirements analysis, product design, programming, test planning, verification & validation, Project office, CM/QA, and manuals. These activity estimates are accompanied with a percentage of the phase effort that they encompass, the estimated effort, schedule and FSWP for the activity's completion as shown in FIGURE 6-11. To exit from this window click the OK button.

FIGURE 6-11

Programming Phase window for the overall Module

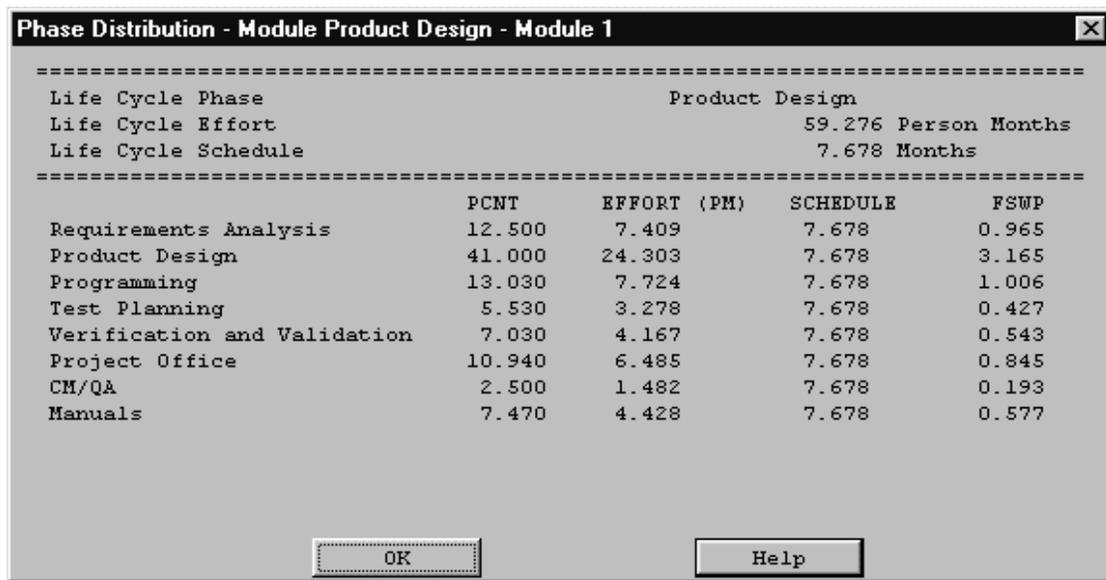
Life Cycle Phase		Programming			
Life Cycle Effort		201.604 Person Months			
Life Cycle Schedule		14.070 Months			
	PCNT	EFFORT (PM)	SCHEDULE	FSWP	
Requirements Analysis	4.000	8.064	14.070	0.573	
Product Design	8.000	16.128	14.070	1.146	
Programming	56.500	113.907	14.070	8.096	
Test Planning	5.030	10.141	14.070	0.721	
Verification and Validation	8.030	16.189	14.070	1.151	
Project Office	6.470	13.043	14.070	0.927	
CM/QA	6.500	13.104	14.070	0.931	
Manuals	5.470	11.027	14.070	0.784	

#### 6.2.4 Product Design Phase Distribution

The product design phase distribution allows the user to view the components of this particular phase. When the Product Design distribution is chosen from the Module Phase distribution menu, the window shown in FIGURE 6-12 is displayed. This window displays the following information: Module name, the total Module SLOC, the total estimated Module effort, and the total estimated Module schedule. In addition the window displays the activities requirements analysis, product design, programming, test planning, verification & validation, Project office, CM/QA, and manuals. These activity estimates are accompanied with a percentage of the phase effort that they encompass, the estimated effort, schedule and FSWP for the activity's completion as shown in FIGURE 6-12. To exit from this window click the OK button.

FIGURE 6-12

Product Design window for the overall Module

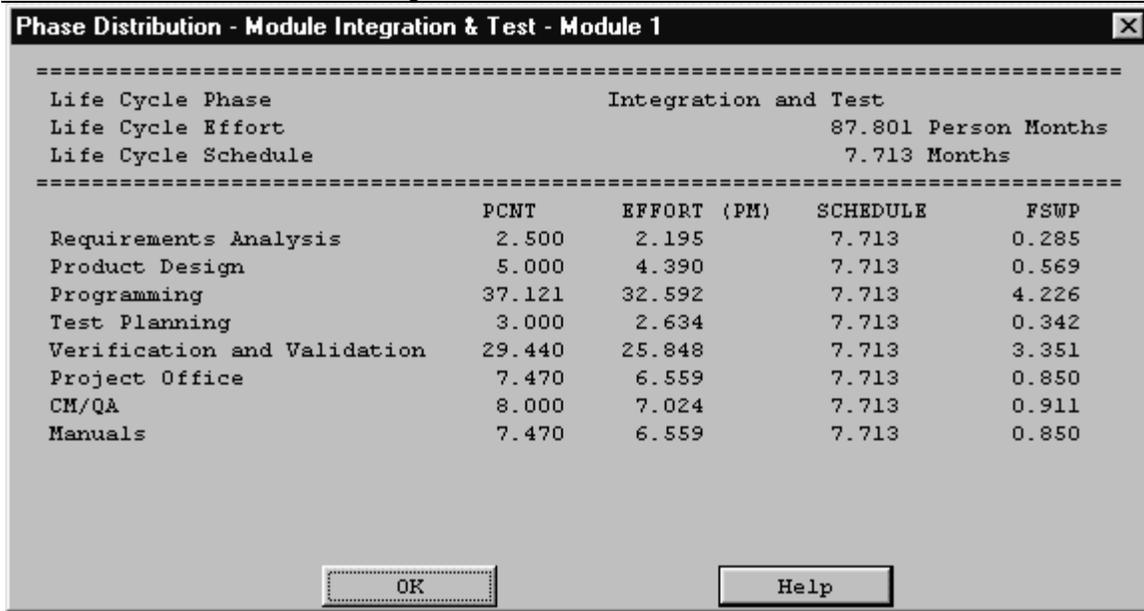


### 6.2.5 Integration and Test Module Phase Distribution

The integration & test phase distribution allows the user to view the components of this particular phase. When the Integration and Test distribution is chosen from the Module Phase distribution menu, the window shown in FIGURE 6-13 is displayed. This window displays the following information: Module name, the total Module SLOC, the total estimated Module effort, and the total estimated Module schedule. In addition the window displays the activities requirements analysis, product design, programming, test planning, verification & validation, Project office, CM/QA, and manuals. These activity estimates are accompanied with a percentage of the phase effort that they encompass, the estimated effort, schedule and FSWP for the activity's completion as shown in FIGURE 6-13. To exit from this window click the OK button.

FIGURE 6-13

Integration & Test window for the overall Module



## Chapter 7: Maintenance

Maintenance is one of the menu selections in the menu bar that can be accessed by either clicking upon "Maintenance" in the menu bar or pressing Meta+M. Its function is to calculate and display an estimate of the effort and cost necessary to maintain a post development software product for a user-defined number of years (maximum five years). Maintenance encompasses the process of modifying existing operational software while leaving its primary functions intact. This process excludes the following types of activities:

Major re-design and re-development (more than 50% new code) of a new software product performing substantially the same functions

Design and development of a sizeable (more than 20% of the source instructions comprising the existing product) interfacing software package which requires relatively little redesigning of the existing product

Data processing system operations, data entry, and modification of values in the database

Maintenance does include the following types of activities:

Re-design and re-development of small portions of an existing software product

Design and development of small interfacing software packages, which require some redesign of the existing software product

Modification of the software product's code, documentation, or database structure

Maintenance effort and costs are determined by essentially the same cost driver attributes used to determine the software development costs and effort (exceptions are the RELY, MODP, and SCED factors which will be discussed in greater detail later in this chapter). The maintenance calculations are heavily based upon the Maintenance change Factor (MCF) and the Maintenance Adjustment Factor (MAF). The MCF is similar to the Annual change Traffic in COCOMO81, except that maintenance periods other than a year can be used (see EQ 7-1).

### Maintenance Change Factor (EQ 7-1)

$$MCF = \frac{SizeAdded + SizeModified}{BaseCodeSize}$$

The initial maintenance size is obtained in one to two ways. The first equation in EQ 7-2 is used when the base code size is known and percentage of change to the base code is known. The second equation in EQ 7-2 is used when the fraction of code added or modified to the existing base code during the maintenance period is known.

$$\begin{aligned} \{Size\}_M &= \{BaseCodeSize\} \times MCF \times MAF \\ \{Size\}_M &= \{SizeAdded + SizeModified\} \times MAF \end{aligned}$$

### Initial Maintenance Size(EQ 7-2)

As shown in EQ 7-2, the initial maintenance size estimate is adjusted with a Maintenance Adjustment Factor (see EQ 7-3).

**Maintenance Adjustment Factor(EQ 7-3)**

$$MAF = 1 + \left( \frac{SU}{100} \times UNFM \right)$$

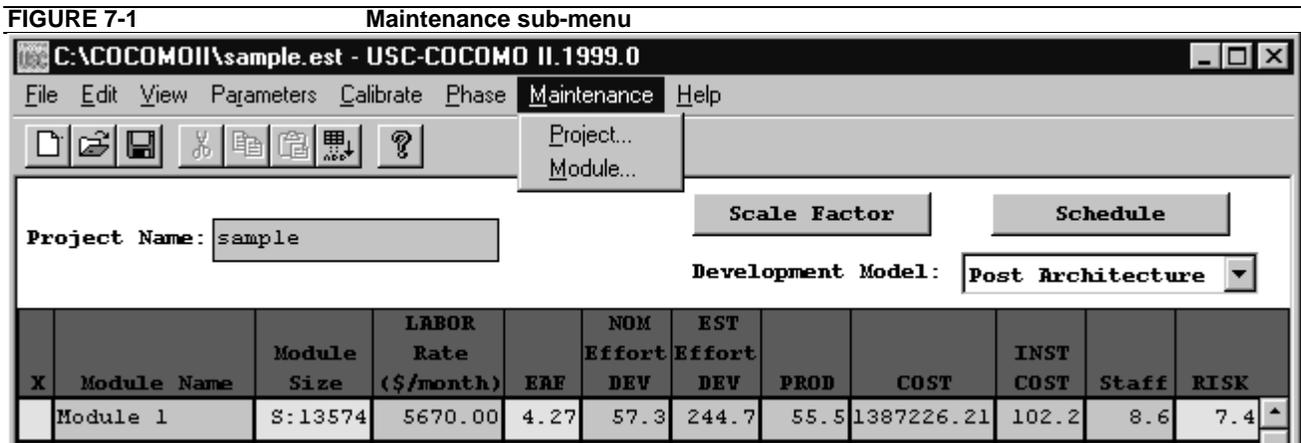
The resulting maintenance effort estimation formula is the same as the COCOMOII Post Architecture development model (see EQ 7-4).

**Maintenance Effort(EQ 7-4)**

$$PM_M = A \times (Size_M)^B \times \prod_{i=1}^{17} EM_i$$

As stated previously, some cost drivers for maintenance differ from development. Those cost drivers are software reliability (RELY) and schedule (SCED). The reason for the change in RELY is that increased investment in software reliability has a strong positive effect upon the maintenance stage. The SCED driver is no longer editable in the EAF window, but is calculated from the user input value for number of years when the maintenance function is engaged. For more information on these cost drivers please refer to the introduction of this manual.

The Maintenance menu option offers sub-menu for either a maintenance effort estimation based upon either an entire project or an individual module (see Figure 7-1). These separate options are discussed in sections 7.1 and 7.2



**7.1 Project Maintenance**

In order to view the maintenance estimation calculations for an entire project, the user can click on Project under the Maintenance menu (see Figure 7-1). Upon clicking this selection, a window will appear displaying the current project name, an EAF button, an editable labor rate field, editable

number of years of maintenance field, an editable percent of added source instructions field per year of maintenance and an editable percent of modified source instructions field per year of maintenance (see Figure7-2).

The EAF rate can be changed by clicking upon the corresponding button. This action

**FIGURE 7-2** Project Maintenance Dialog Box

EAF	1.00
Labor Rate	\$ 5500
Life Span	3 Year
Code Modified per Year	15 %
Code Added per Year	25 %
Software Understanding(SU): [0-50]	30
Unfamiliarity with Software(UNFM): [0.0-1.0]	0.4

Buttons: OK, Cancel, Help

will result in the appearance of an EAF dialog box where the cost driver ratings can be changed as described in the introduction (see Figure 7-3).

Upon completing the adjustment of the cost drivers click the OK button or click the Cancel button to return to the CLEF without viewing maintenance estimations.

After exiting the EAF dialog box with the OK button, you will be returned to the Project Maintenance Dialog box to continue inputting the editable values.

Click upon the OK button upon completion of editing the displayed fields or click upon the Cancel button if no changes are desired to the default values (if more assistance, the Help button is available to receive on-line assistance).

**FIGURE 7-3 Project Maintenance EAF Dialog Box**

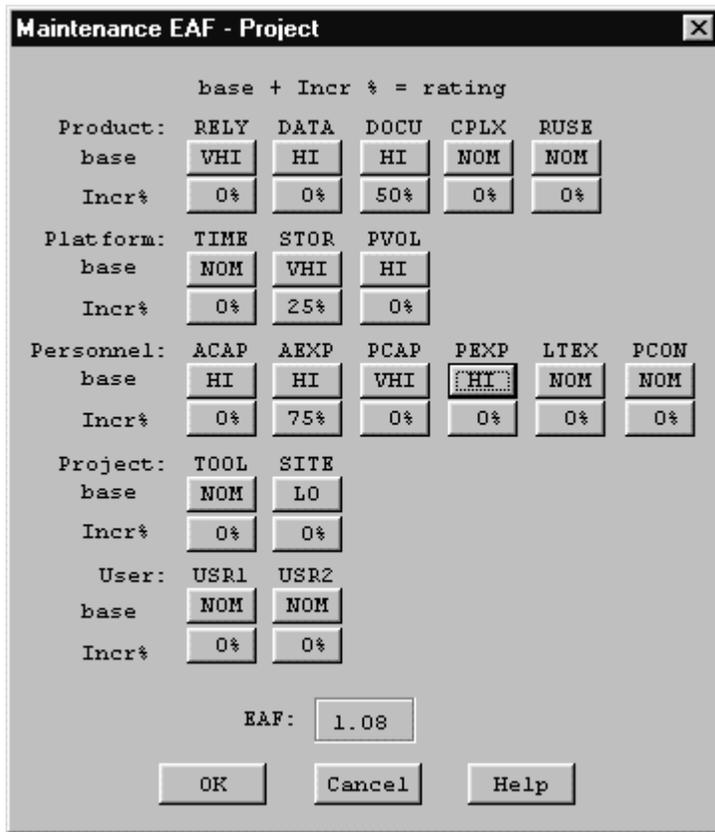
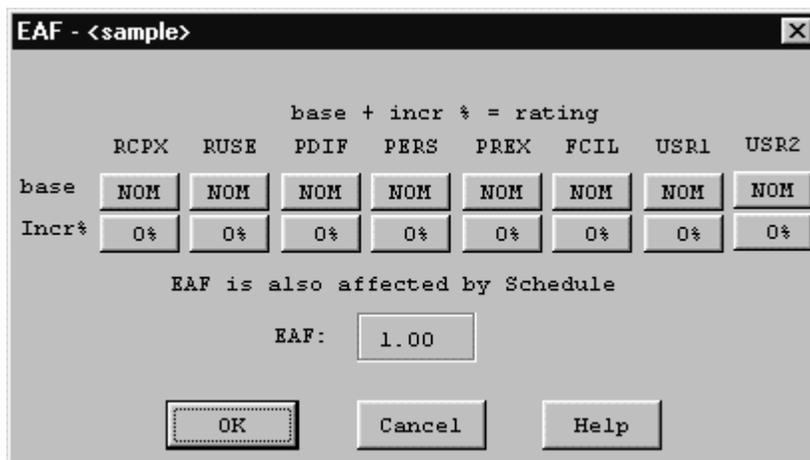


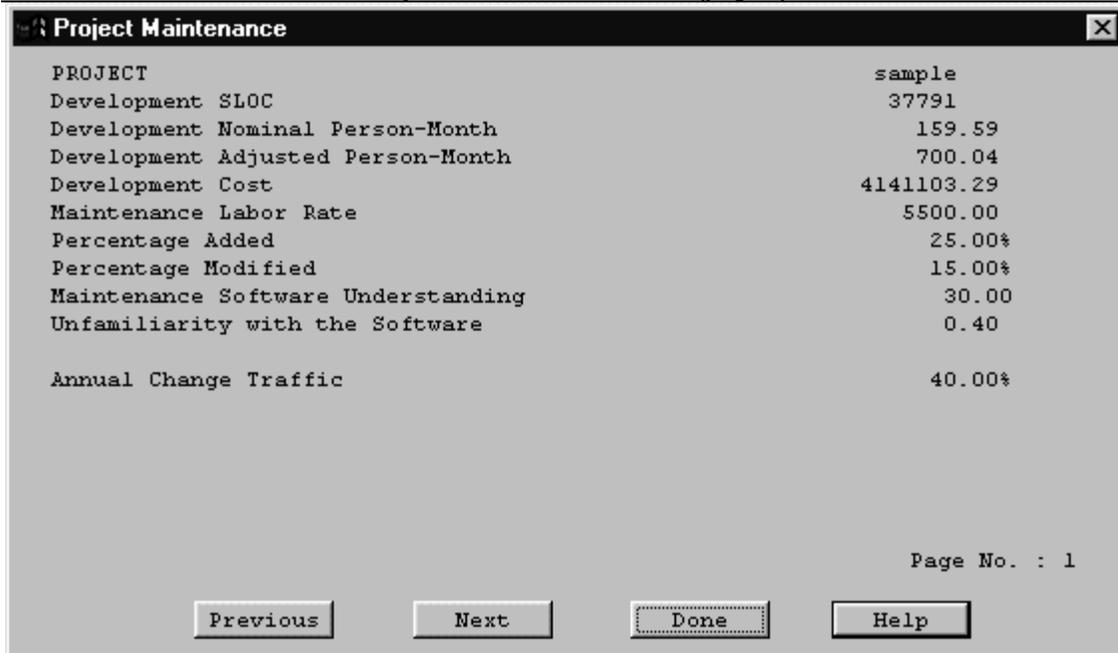
Figure 7-3 shows the EAF dialog box that is used in the Post Architecture Model. It contains sixteen EAF factors plus two user-defined factors. Figure 7-4 shows the EAF dialog box that is used in the Early Design model. Here there are only six EAF factors and two user defined factors.

**FIGURE 7-4 Early Design EAF Maintenance Factors Dialog Box**



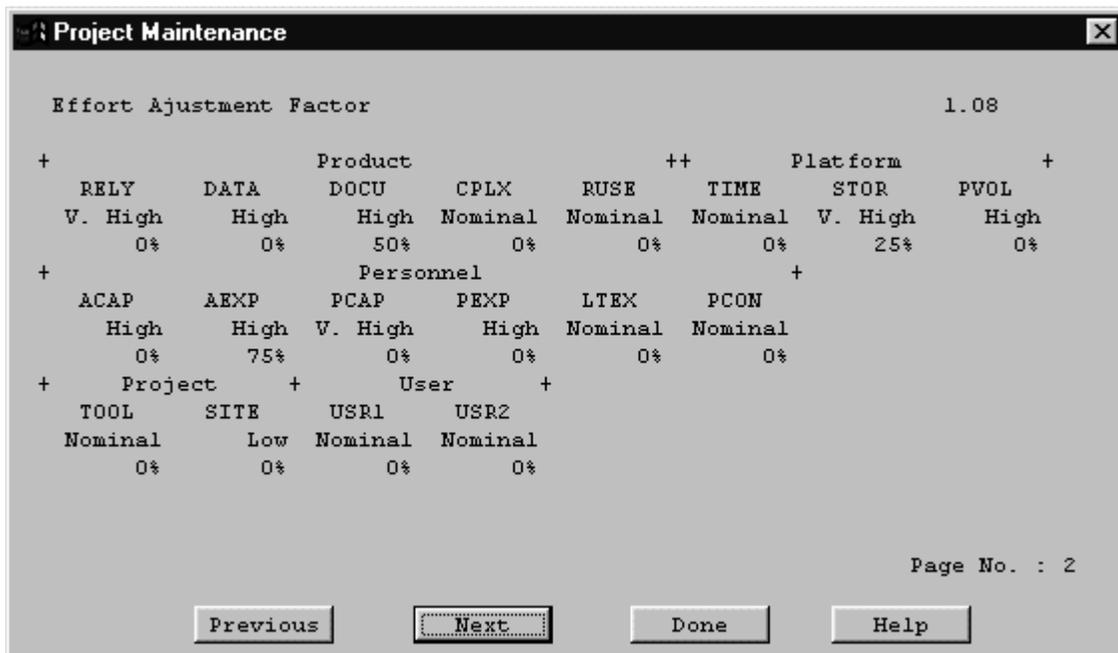
When the OK button is clicked in the Project Maintenance Dialog Box, a window displaying the first of four pages appears. It contains the project name, the current development mode, the total number of source instructions for development of the project (EDSI) that is loaded in the CLEF, the nominal effort of the project, the actual effort of the project, the development cost, the provided maintenance labor rate, the provided percent of code added during maintenance per year, the provided percent of code modified during maintenance per year (see Figure 7-5) and the calculated annual change traffic.

**FIGURE 7-5 Project Maintenance window (page 1)**



The second page of the maintenance window can be seen by clicking upon the Next button. It contains the settings for the 18 cost drivers, (see Figure 7-6).

**FIGURE 7-6 Project Maintenance window (page 2)**



The third page of the maintenance window contains the effort and cost estimation for the next N number of years (as defined by the user). With each year is listed the KSLOC (SLOC \* 10<sup>3</sup>), the nominal effort for development (PM nom), the actual effort for maintenance (PM maint), the number of full time software personnel necessary to maintain the project for the year (FSWP), the number of instructions that are to be maintained be per personnel (KSLOC/FSWP) and the total cost for maintenance for the year (see Figure 7-7).

**FIGURE 7-7** Project Maintenance window (page 3)

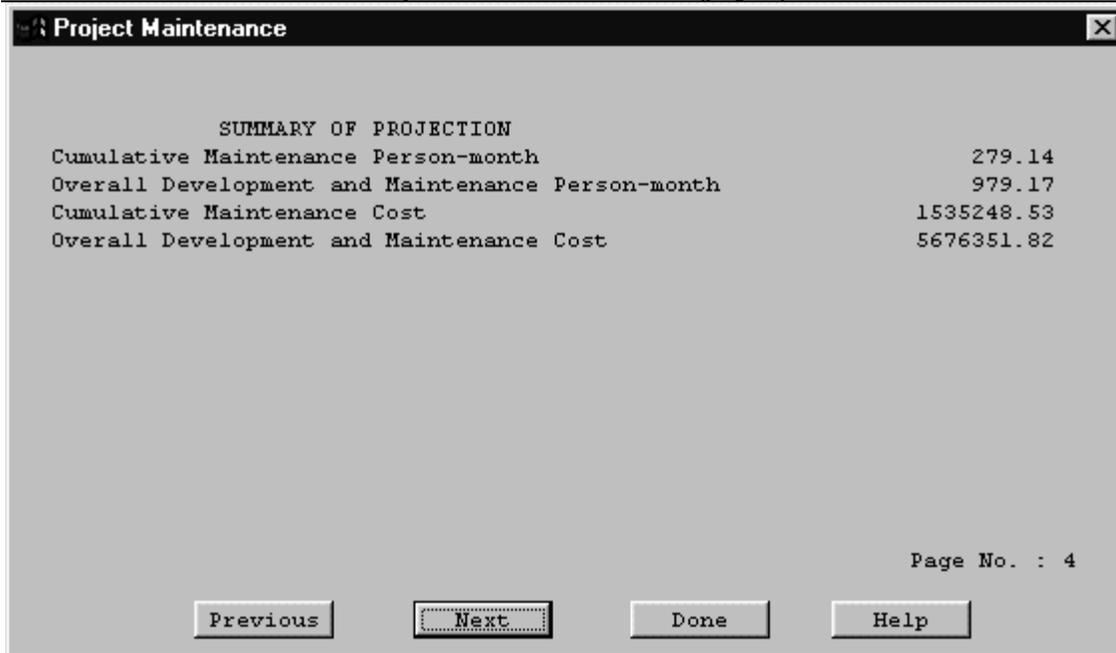
WORK LOAD PROJECTION FOR THE NEXT 3 YEARS						
Year	KSLOC	Effort_Nom	Effort_Maint	Staff	KSLOC/Staff	Cost
1	37.79	66.00	71.36	5.95	6.36	392474.64
2	47.24	84.35	91.21	7.60	6.22	501630.04
3	59.05	107.81	116.57	9.71	6.08	641143.85

Page No. : 3

Previous    **Next**    Done    Help

The fourth window of the maintenance window contains the cumulative figures for effort and cost for maintenance for the total number of years (see figure 7-8). This first displays the total amount of effort estimated for maintenance, then sums the effort of development and maintenance together. It also displays the total cost of maintenance of the project and then displays the summed total cost of development and maintenance for the entire project.

**FIGURE 7-8** Project Maintenance window (page 4)

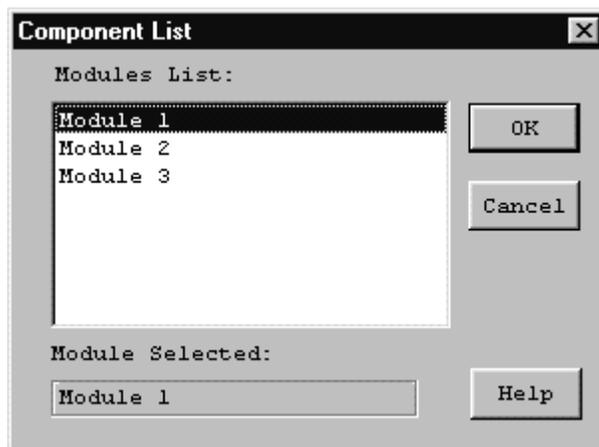


Note - Each individual page can be seen by cycling through the pages pressing either the Previous or Next buttons as needed.

## 7.2 Module Maintenance

In order to view the maintenance estimation calculations for a module, the user can click on Module under the Maintenance menu (see Figure 7-1). Upon clicking upon this selection a window will appear displaying the current module names. Choose only one of the modules by highlighting the appropriate module name and then clicking upon OK (see Figure 7-9).

**FIGURE 7-9** Module Selection window



Upon exiting the module selection window, another window will be appear that displays, the selected module name, an EAF button, an editable labor rate field, editable number of years of maintenance field, an editable percent of added source instructions field per year of maintenance and an editable percent of modified source instructions field per year of maintenance (see Figure 7-10).

**FIGURE 7-10** **Module Maintenance Dialog Box**

EAF	0.70
Labor Rate	\$ 4700
Life Span	3 Year
Code Modified per Year	20 %
Code Added per Year	25 %
Software Understanding (SU) : [0-50]	30
Unfamiliarity with Software (UNFM) : [0.0-1.0]	0.4

Buttons: OK, Cancel, Help

The EAF rate can be changed by clicking upon the corresponding button. This action will result in the appearance of an EAF dialog box where the cost driver ratings can be changed as described in the introduction. The EAF factors for the Post Architecture Model are shown in Figure 7-11 and for the Early Design Model in Figure 7-12.

**FIGURE 7-11** Module Maintenance EAF Dialog Box

base + Incr % = rating

Product:	RELY	DATA	DOCU	CPLX	RUSE	
base	LO	NOM	HI	NOM	NOM	
Incr%	0%	0%	25%	0%	0%	
Platform:	TIME	STOR	PVOL			
base	NOM	VHI	HI			
Incr%	0%	50%	0%			
Personnel:	ACAP	AEXP	PCAP	PEXP	LTEX	PCON
base	HI	HI	HI	VHI	HI	NOM
Incr%	25%	75%	0%	0%	50%	0%
Project:	TOOL	SITE				
base	HI	NOM				
Incr%	50%	0%				
User:	USR1	USR2				
base	NOM	NOM				
Incr%	0%	0%				

EAF: 0.70

OK Cancel Help

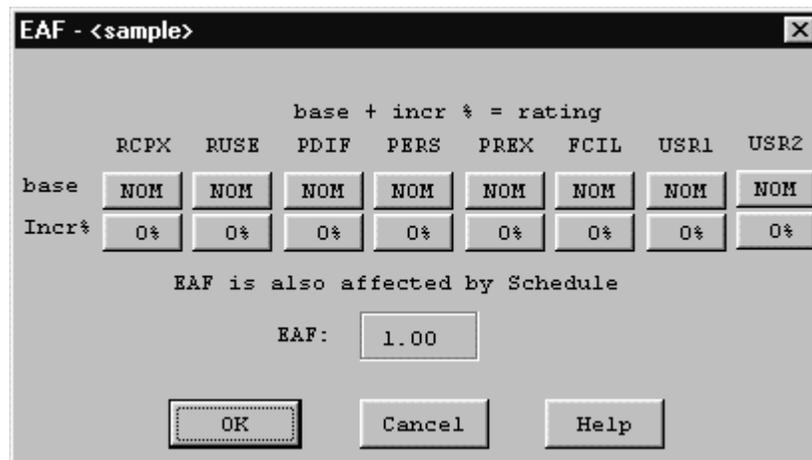
Upon completing the adjustment of the cost drivers click the OK button or click the Cancel button to return to the CLEF without viewing maintenance estimations.

After exiting the EAF dialog box, you will be returned to the Module Maintenance Dialog box to continue inputting the editable values.

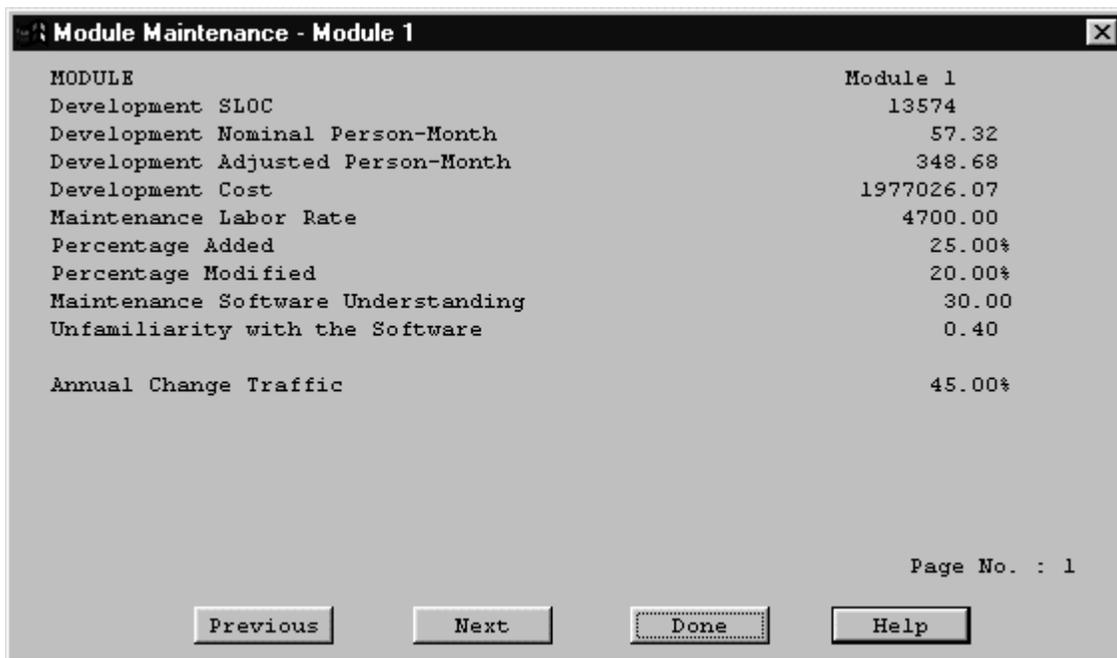
Click upon the OK button upon completion of editing the displayed fields or click upon the Cancel button if no changes are desired to the default values (if more assistance, the Help button is available to receive on-line assistance).

When the OK button is clicked in the Module Maintenance Dialog Box, a window displaying the first of four pages that contains the module name, the current development mode, the total number of source lines of code for development of the module (SLOC) that is loaded in the CLEF, the nominal effort of the module, the actual effort of the module, the development cost, the inputted maintenance labor rate, the inputted percent of code added during maintenance per year, the inputted percent of code modified during maintenance per year (see figure 7-13) and the calculated annual change traffic.

**FIGURE 7-12** Module Maintenance Dialog Box, Early Design

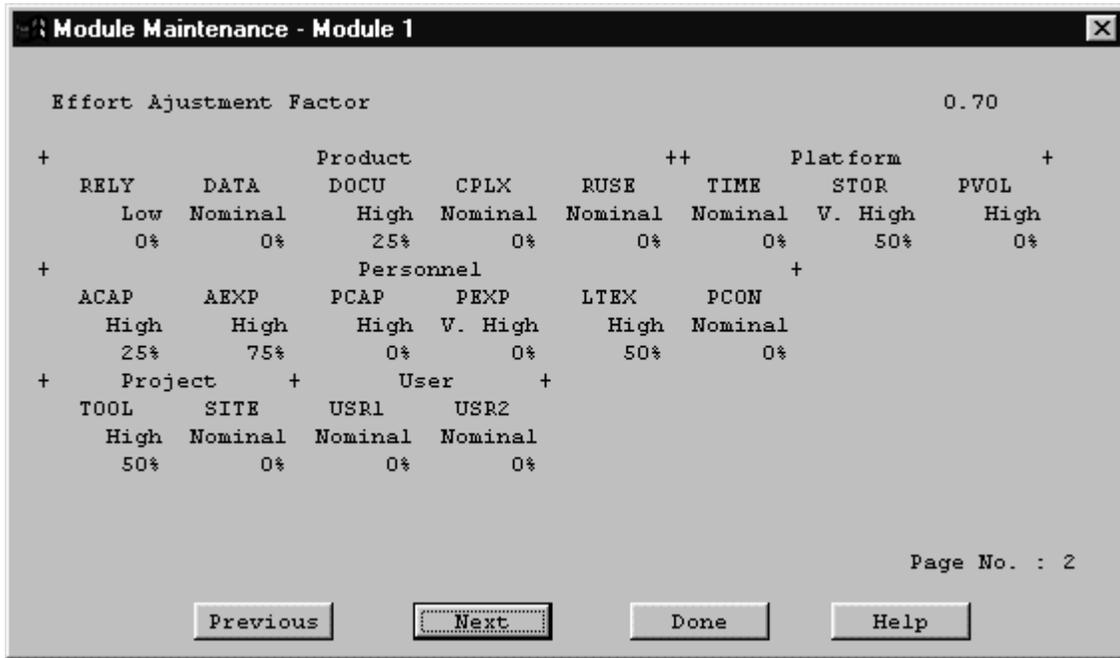


**FIGURE 7-13** Module Maintenance window (page 1)



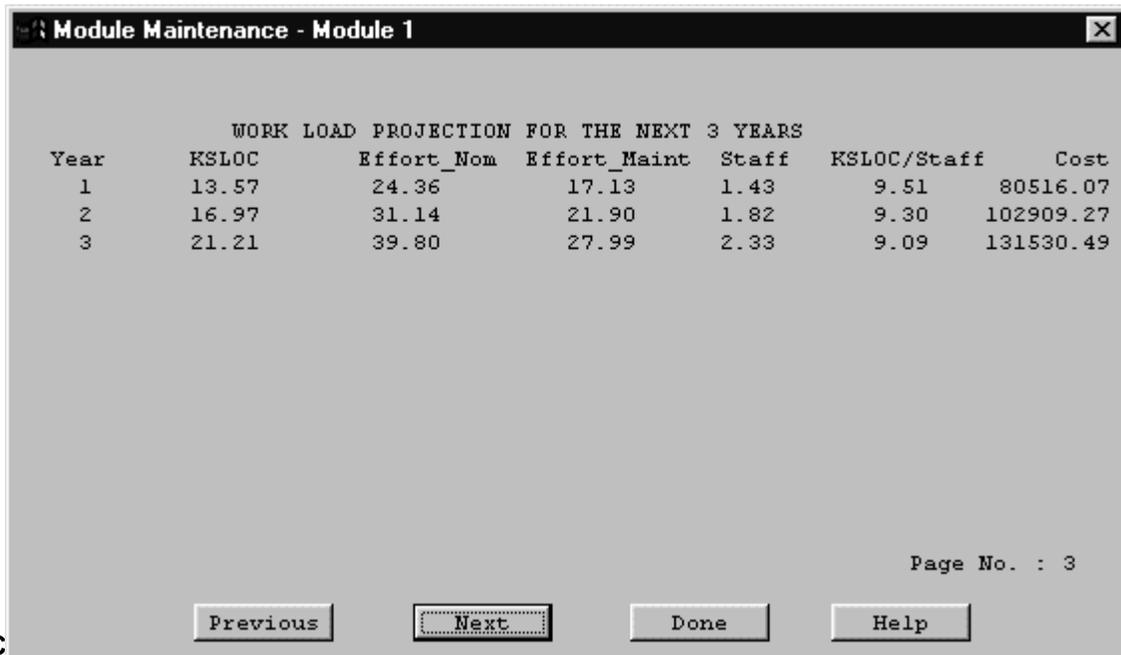
The second page of the maintenance window can be seen by clicking upon the Next button. It contains the settings for the 16 cost drivers, SCED is not applicable (see Figure 7-14).

**FIGURE 7-14** Module Maintenance window (page 2)

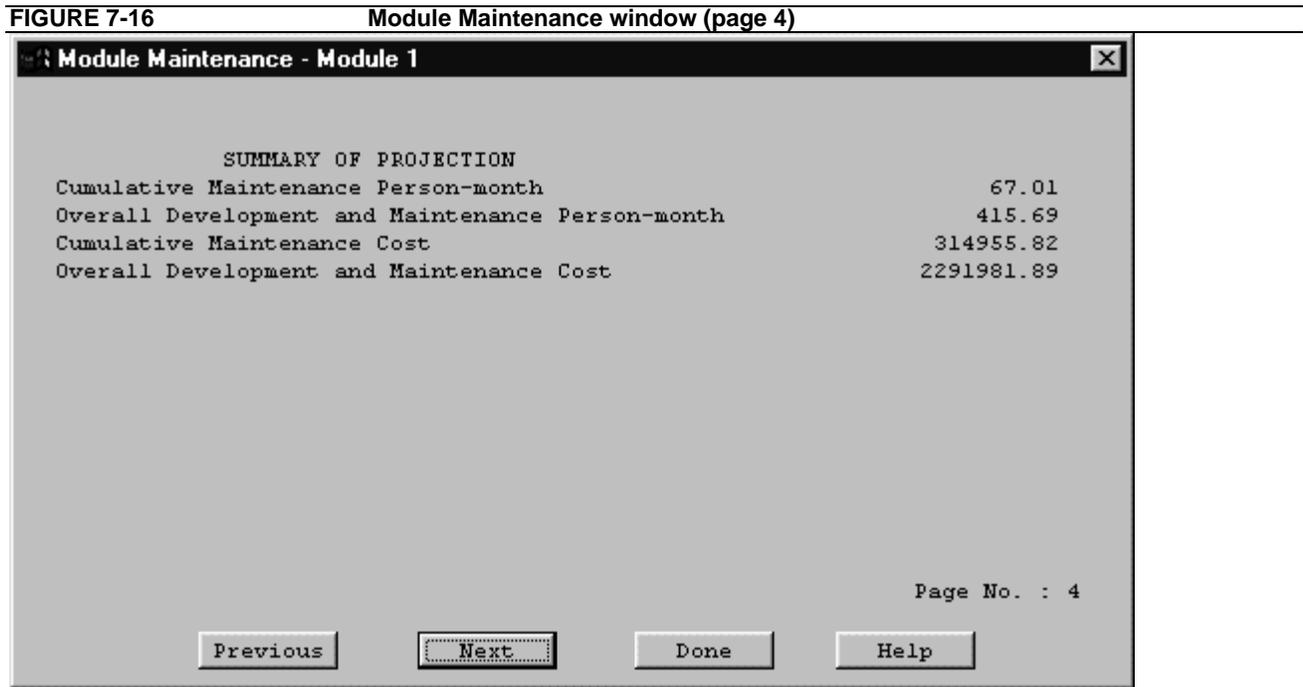


The third page of the maintenance window contains the effort and cost estimation for the next N number of years (as defined by the user). With each year is a listed the KSLOC (SLOC \* 10<sup>3</sup>), the nominal effort for development (PM nom), the estimated effort for maintenance (PM maint), the number of full time software personnel necessary to maintain the module for the year (FSWP), the number of instructions that are to be maintained be per personnel (KSLOC/FSWP) and the total cost for maintenance for the year (see Figure 7-15).

**FIGURE 7-15** Module Maintenance window (page 3)



The fourth window of the maintenance window contains the cumulative figures for effort and cost for maintenance for the total number of years (see Figure 7-16). This first displays the total amount of effort estimated for maintenance, then sums the effort of development and maintenance together. It also displays the total cost of maintenance of the module and then displays the summed total cost of development and maintenance for the entire module.



Note - Each individual page can be seen by cycling through the pages pressing either the Previous or Next buttons as needed.

## References

- Amadeus (1994), Amadeus Measurement System User's Guide, Version 2.3a, Amadeus Software Research, Inc., Irvine, California, July 1994.
- Behrens, C. (1983), "Measuring the Productivity of Computer Systems Development Activities with Function Points," *IEEE Transactions on Software Engineering*, November 1983.
- Boehm, B. (1981), *Software Engineering Economics*, Prentice Hall.
- Boehm, B. and W. Royce (1989), "Ada COCOMO and the Ada Process Model," *Proceedings, Fifth COCOMO Users' Group Meeting*, Software Engineering Institute, Pittsburgh, PA, November 1989.
- Boehm et al. (1995), "Cost Models for future Software Life Cycle Process: COCOMO 2.0", *Annals of Software Engineering Special Volume on Software Process and Product Measurement*, J.D Arther and S.M. Henry, Eds., J.C. Baltzer AG, Science Publishers, Amsterdam, The Netherlands, Vol 1, pp. 45 - 60.
- Chidamber, S. and C. Kemerer (1994), "A Metrics Suite for Object Oriented Design," *IEEE Transactions on Software Engineering*, (to appear 1994).
- Goethert, W., E. Bailey, M. Busby (1992), "Software Effort and Schedule Measurement: A Framework for Counting Staff Hours and Reporting Schedule Information." CMU/SEI-92-TR-21, Software Engineering Institute, Pittsburgh, PA.
- IEPUG (1994), *IFPUG Function Point Counting Practices: Manual Release 4.0*, International Function Point Users' Group, Westerville, OH.
- Kunkler, J. (1985), "A Cooperative Industry Study on Software Development/Maintenance Productivity," Xerox Corporation, Xerox Square --- XRX2 52A, Rochester, NY 14644, Third Report, March 1985.
- Madachy, J. Raymond (1997), "Heuristic Risk Assessment Using Cost Factors," *IEEE Software*, May/June 1997, pp. 51-59.
- Park R. (1992), "Software Size Measurement: A Framework for Counting Source Statements," CMU/SEI-92-TR-20, Software Engineering Institute, Pittsburgh, PA.
- Selby, R., A. Porter, D. Schmidt and J. Berney (1991), "Metric-Driven Analysis and Feedback systems for Enabling Empirically Guided Software Development," *Proceedings of the Thirteenth International Conference on Software Engineering (ICSE 13)*, Austin, TX, May 13-16, 1991, pp. 288-298.

## Appendix A: Accelerator Keys

	<b>Sparc</b>		<b>Windows</b>	
<b>File</b>	F	Meta+F		Alt+F
New	N	Ctrl+N	N	Ctrl+N
Load Project	L	Ctrl+L	L	Ctrl+L
Save Project	S	Ctrl+S	S	Ctrl+S
Save As Project	A	Ctrl+A	A	Ctrl+A
Load Model			O	
Save Model			V	
Save As Model			E	
Make Report			R	
Exit	X		X	
<b>View</b>	No			Alt+V
<b>Edit</b>	E	Meta+E		Alt+E
Add Module	No		A	Ctrl+A
Clear All Modules	C		L	
Snapshot	S		S	
Undo		Ctrl+Z	U	Ctrl+Z
Cut		Ctrl+X	T	Ctrl+X
Copy		Ctrl+C	C	Ctrl+C
Paste		Ctrl+V	P	Ctrl+V
<b>Parameters</b>	P	Meta+P		Alt+R
Product	P		P	
Platform	L		L	
Personnel	E		N	
Project	J		J	
User EAF	S		U	
Scale Factor	C		S	
Equation	E		E	
Reset	R		R	
<b>Calibrate</b>	C	Meta+C		Alt+C
File Load	L		L	
File Save	S		S	
File Save As	A		A	
Project	P		P	
Compute	C		C	
<b>Phase</b>	D	Meta+P		Alt+P
<b>Project</b>			P	
Overall Phase	O		O	
Plans & Requirement	R		R	
Programming	P		P	
Product Design	D		D	
Integration & Test	I		I	

<b>Module</b>		M
Overall Phase	O	O
Plans & Requirement	R	R
Programming	P	P
Product Design	D	D
Integration & Test	I	I
<b>Help</b>	H      Meta+H	Alt+H
On Application	A	No
On Version	V	No
COCOMOII User's Manual	No	C
Using Help	No	U
About USC-COCOMOII	No	A

**A**

AA 9  
AAF 9  
ACAP 10, 45  
ACT PM 18, 57  
Adaptation 9  
Adaptation Adjustment Factor 13  
Add Module 19  
ADSI 9  
AEXP 10, 45  
ASLOC 9

**B**

Boehm 1  
Breakage 2

**C**

Calibrate 11, 43  
calibration 11  
Clear 37  
CLEF 11, 41  
CM 9  
CM/QA 58, 59, 60, 61, 62, 64, 66, 67  
COCOMO 1, 9, 11, 13, 21, 22, 24, 25, 27, 28, 29, 30, 36, 38, 43, 49, 57, 59  
COCOMO81 57  
COCOMOII 1, 3, 5, 51, 56, 87  
coefficient 51  
complexity level 8  
Computer 43  
Copy 41  
Cost 18  
cost drivers 16  
COTS 5  
CPLX 10, 43  
Cut 41

**D**

DATA 9, 43  
Design 57  
development mode 16  
DI 8  
DM 9  
DOCU 9, 43

**E**

EAF 10, 18, 21, 51  
Edit 11, 37, 38, 40, 41  
EDSI 57  
Effort Adjustment Factor 16  
effort multipliers 10  
EM 10  
Equations 43  
EST PM 16, 18  
Exit 36  
exponent 51

**F**

File 11  
Files 25  
FLEX 48  
FSWP 16, 17, 58, 60, 61, 62, 64, 66, 57  
Function point 8  
functionality 5, 8

**G**

GFS 5

## **H**

Help 11

## **I**

IM 9

Instruction Cost 17

integration & test 57

integration & test phase 62, 68

## **J**

Java 87

Java COCOMO 87

## **K**

KSLOC 4, 9

## **L**

Labor Rate 16

labor rate 21

LEXP 10, 45

## **M**

Main Menu bar 11

Maintenance 11, 71

Microhelp 16

Module 12, 77

module 1, 16, 41, 43, 57, 71

model file 21, 22, 27, 28, 30

Module Phase Distribution 64

## **N**

New 22

NOM PM 18, 57

## **O**

overall phase 65

## **P**

Parameters menu 43, 45, 46, 48, 49

Paste 41

PCAP 10, 45

PCON 10, 45

Percent of Code Modification (CM) 9

Percent of Design Modification (DM) 9

Percent of Integration Required for Modified Software (IM) 9

Personnel 43, 45

Personnel attributes 10

personnel attributes 9, 21

PEXP 10, 45

Phase Distribution 11, 57

phase distribution 57

Plan & Requirements 57

plans & requirements 57

plans and requirements phase 66

Platform 16, 44

platform attribute 9, 21

Platform attributes 10

PM 10, 57

PMAT 48

PREC 48

predictor factors 21

PROD 18

Product 16, 43

Product attributes 10

product attributes 9, 21

product design phase 61, 67

Productivity 18

Programming 57  
programming 57  
programming phase 61, 67  
Project 43, 46, 73  
project 1, 25, 39, 43, 51, 54, 57, 71  
project attribute 10, 21  
Project attributes 10  
Project File 23  
project file 21, 22, 24, 26, 28  
Protocol 87  
PVOL 10, 44

## **R**

RELY 9, 43  
report 21  
report file 21, 31  
Reset 43, 49  
RESL 48  
Risk 16  
risk level 16  
RUSE 10, 43

## **S**

Scale Factors 48  
scale factors 4  
SCED 10  
Schedule 16  
SEI 5  
SITE 10, 46  
SLOC 9, 13, 16, 17, 21, 61, 62, 66  
Snapshot 38, 39  
software architecture 57  
STOR 10, 44  
SU 9

## **T**

TCP/IP 87  
TEAM 48  
TIME 10, 44  
TOOL 10, 46

## **U**

Unadjusted Function Points 8  
Undo 40  
UNFM 9  
User Defined EAF 10, 47

## **V**

V&V 59, 64  
Virtual COCOMOII 87

## **W**

waterfall model 1

## **X**

x 12