USC COCOMO

Reference Manual

University of Southern California

This manual is compatible with USC COCOMO81a.

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.

Acknowledgments

Version 1:
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

Version 1.1:
Principal Investigator - Dr. Ellis Horowitz

Student Designers, Testers and Programmers Ing-Jer Huang

Version 10.0:
Principal Investigator - Dr. Ellis Horowitz

Student Designers, Testers and Programmers - M. Susan Lane, Ping Luo, Lorna Zorman

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

Note - This manual has been written for the X Windows and MS Windows 3.0 versions of COCOMO. Though there are some inherent differences between these systems, the text and related examples are compatible to both versions.

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.
This research is sponsored by the Advanced Projects Research Agency (ARPA) through Rome Laboratory under contract F30602-94-C-0195, and by the Affiliates of the USC Center for Softrware Engineering. The current Affiliates are
Aerospace Corp.,
Air Force Cost Analysis Agency,
AT&T Bell Laboratories,
Bellcore,
DISA,
E-Systems,
Electronic Data Systems,
Hughes Aircraft Company,
Institute for Defense Analysis,
Interactive Development Environments,
Jet Propulsion Laboratory,
Litton Data Systems,
Lockheed Martin Corp,
Loral,
Motorola,
Northrop Grumman Corp.,
Rational Inc.,
Rockwell International,
Science Appications International Corp.,
Software Engineering Institute,
Software Productivity Consortium,
Sun Microsystems,
Teledyne Inc.,
TRW,
U.S. Air Force Rome Laboratory,
U.S. Army Research Laboratory,
Xerox

Chapter

1

Introduction

1.1 What is COCOMO?

COCOMO (Constructive Cost Model) is a screen-oriented, interactive software package that assists in budgetary planning and schedule estimation of a software project prior to any work beginning. In addition, COCOMO also assists in the budgetary planning of the maintenance effort after delivery of the software package up to a maximum of a five year period. 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 after the potential software package has been completed.

The COCOMO software package is based solely upon the intermediate version of the Constructive Cost Model first published by Dr. Barry Boehm in his book Software Engineering Economics, Prentice-Hall (1981). Its publication followed a development and trial period during Dr. Boehm's tenure with TRW from 1976-1979. During this period, the number of source instructions (called the equivalent delivered source instructions or EDSI), total development time, and the total effort necessary for 40 aerospace industry software projects were studied. From this information, the estimation formulas for COCOMO were created to calculate estimates for the time of development and the amount of effort needed for software development. It was later discovered that the schedule and effort are influenced by certain factors related to the difficulty of the project. The level of difficulty (or familiarity) is broken down into 3 modes: organic, semi-detached and embedded.

The modes are defined as follows:

- **Organic mode** is used to calculate the effort for a project where constraints upon development are mild. In addition, the given project has been pre-dated by a number similar projects, that could assist in defining the agenda of development.

- **Semi-detached mode** is used for a project where the constraints on the project are greater than organic mode, but there still remains some flexibility. The project may only be pre-dated by a few similar projects.

- **Embedded mode** is used for a project that has very tightly defined constraints. The project as a whole is a trailblazer and therefore cannot rely upon previous projects completed.

For each mode of effort estimation, the effort result is given in units of Person-Month (PM). PM is the number of months one person would need to develop a given project. The schedule estimation is given in the actual number of months needed for development by a properly staffed full-time development team. The equations for computing effort and time for each of the three modes is given in table 1-1. The value of PM used in the Schedule equation is divided by SCED, the effort multiplier.

TABLE 1-1 Effort and Schedule Estimation

MODE	EFFORT ESTIMATION	SCHEDULE ESTIMATION
Organic Mode	$PM_{nominal} = 3.2\ (KDSI)^{1.05}$	$TIME_{dev} = 2.5\ (PM)^{0.38}$
Semi-detached Mode	$PM_{nominal} = 3.0\ (KDSI)^{1.12}$	$TIME_{dev} = 2.5\ (PM)^{0.35}$
Embedded Mode	$PM_{nominal} = 2.8\ (KDSI)^{1.20}$	$TIME_{dev} = 2.5\ (PM)^{0.32}$

The different modes alone were found not to be the only contributing factor 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, computer attributes, personnel attributes, and project attributes.

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

- Required software reliability (RELY)
- Database size (DATA)
- Product complexity (CPLX)

Computer 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)
- Virtual machine volatility (VIRT)
- Computer turnaround time (TURN)

Personnel attributes refer to the level of skill that is 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)
- Virtual machine experience (VEXP)
- Programming language experience (LEXP)

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

- Modern programming practices (MODP)
- Use of software tools (TOOL)
- Required development schedule (SCED)

These 15 factors (or multipliers) are incorporated into calculating an estimated effort and schedule. Each of the factors has associated with it up to five ratings. These ratings are *very low, low, nominal, high,* and *very 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 multipliers are incorporated into the schedule and effort estimation formulas by multiplying them together. The numerical value of the i^{th} adjustment factor (there are 15 of them) is called EM_i and their product is called the estimated adjustment factor or EAF. The actual effort, PM_{total} is the product of the nominal effort times the EAF (see figure 1-1).

FIGURE 1-1 **Estimate Development Effort**

$$PMtotal = PMnominal \times \prod_{i=1}^{15} EMi$$

As stated above, COCOMO's accuracy was improved with the incorporation of the effort adjustment factors into the schedule and effort estimation formulas. COCOMO accuracy is reflected in figures 1-2 and 1-3.

FIGURE 1-2 **Intermediate COCOMO estimates vs. project actuals**

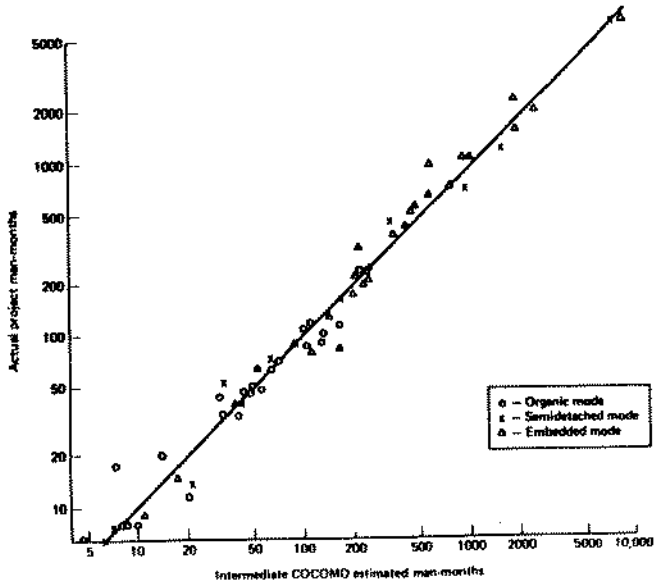
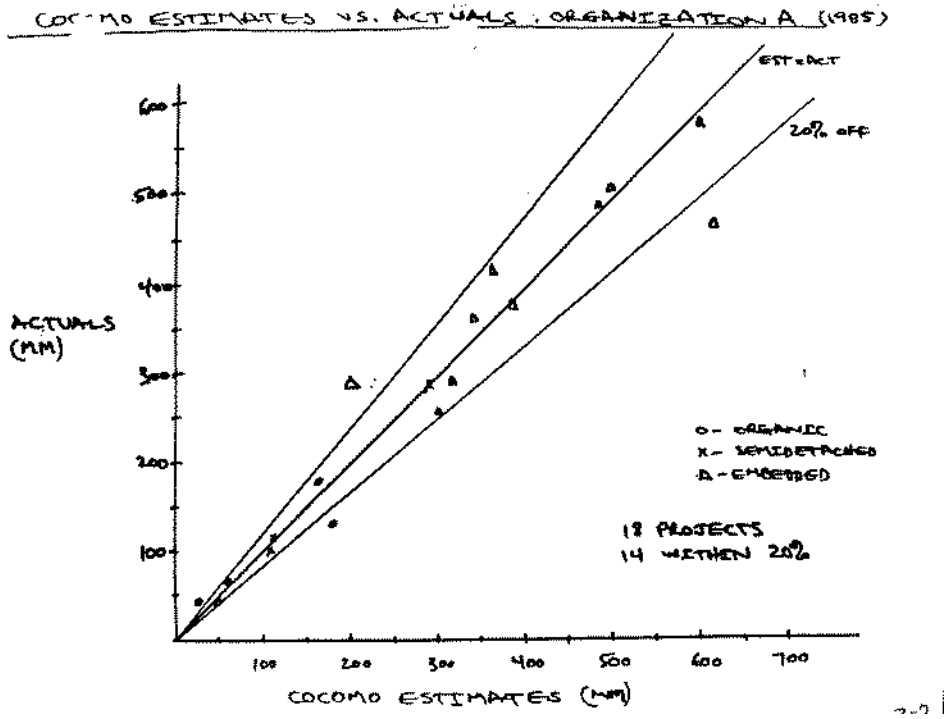


FIGURE 8-3 Intermediate COCOMO estimates vs. project actuals

FIGURE 1-3 **COCOMO Estimates vs. actuals - Organization A (1985)**



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 building upon already existing code. Adaptation considerations have also been incorporated into COCOMO, where an estimate for KDSI will be calculated. This value will be substituted in place of the DSI 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 Delivered Source Instructions (ADSI). The number of delivered source instructions 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.

These components are brought together in the equations found in figure 1-4. The AAF of figure 1-4 is the adaptation adjustment factor. The AAF is the calculated degree to which the adapted software will affect overall development.

FIGURE 1-4 **COCOMO Adaptation Estimating Equations**

$$AAF = 0.40(DM) + 0.30(CM) + 0.30(IM)$$

Maintenance

With COCOMO, one can estimate the annual change traffic (ACT) that would occur to already existing code up to maximum of five years. This function provides an estimate for the effort and schedule of maintaining post development code for a given period of time, in the units of years.

This function is described in more detail in chapter 6.

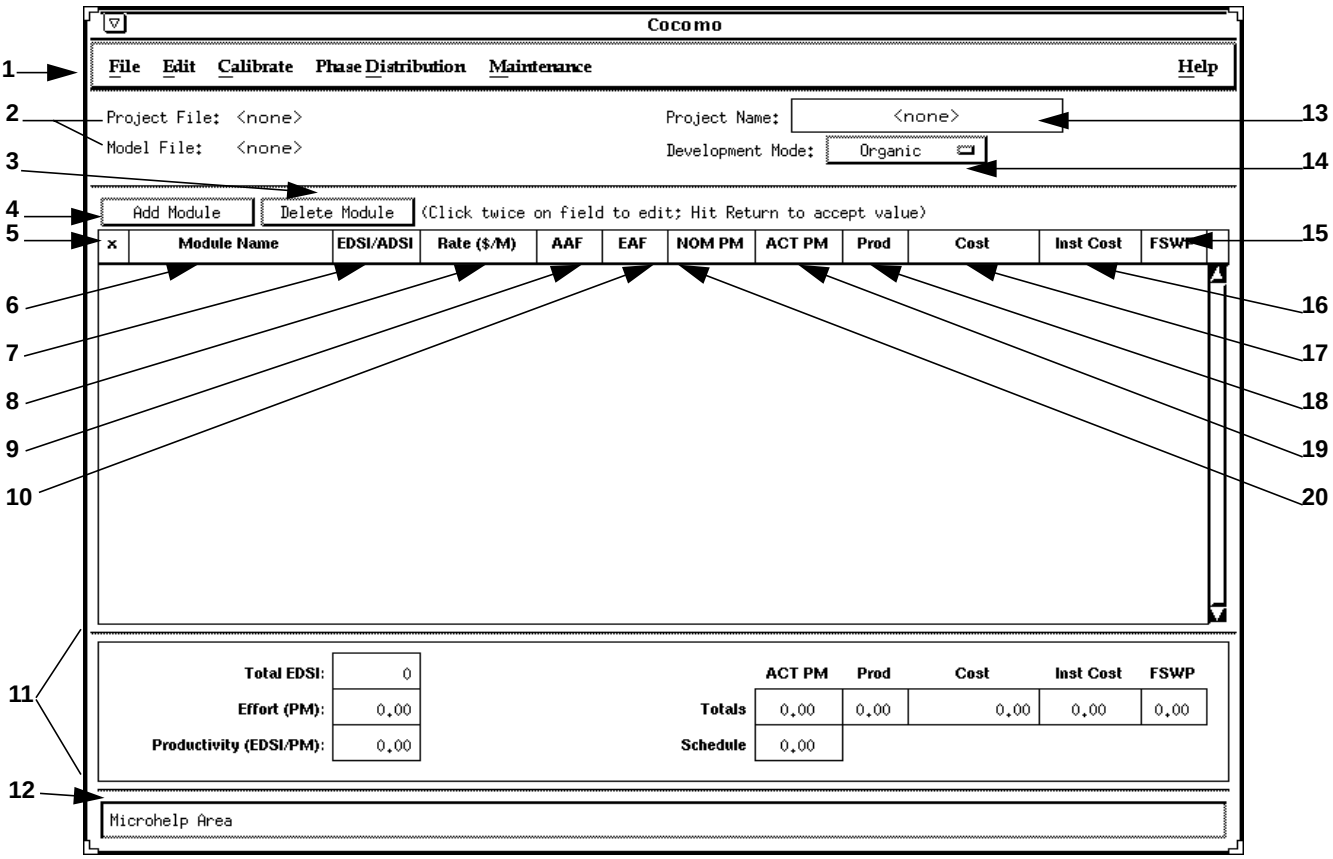
1.2 Navigating COCOMO

In order to efficiently use COCOMO, you must become familiar with the Component Level Estimating Form (CLEF) that is pictured in figure 1-5. The different sections 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 1-5 and their descriptions are as follows:

1. **Main Menu bar** - This area houses the menu selection of the main functions of COCOMO. These selections are *File*, *Edit*, *Calibrate*, *Phase Distribution*, *Maintenance* and *Help*. File, Edit, Calibrate, Phase Distribution, and Maintenance are discussed in chapters 2, 3, 4, 5 and 6 respectively. Help is the selection used to receive on-line assistance with the available functions.

FIGURE 1-5 **COCOMO CLEF**



2. **Project Reference Names** - These fields denoted by this arrow contain the following elements: *Project File and Model File*.

- Project File displays the name of the current project file loaded into COCOMO. The default name is "(none)" as found in figure 1-5.
- Model File displays the name of the current model file that has been loaded into COCOMO. The default name is "(none)" as found in figure 1-5.

3. **Delete Module** - This button is used to delete a module that has been marked with an "x" (see number 5).

4. **Add Module** - This button when depressed adds a new module to the project that is currently being worked upon by the user.

5. **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.

6. **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".

7. **Adapted/Equivalent Delivered Source Instructions (ADSI/EDSI)Column** - This column is used to house the ADSI/EDSI of each module located in the Module Area. The value for ADSI can be changed by clicking on the AAF field and editing the components of AAF (see AAF). The value for EDSI can be changed by clicking on the EDSI box of the desired module and entering the new numeric value within this field. Upon completion press "Return". There is a limit to the range of input. The inputted value for both ADSI and EDSI must be within the range 0 - 999,999.

Note - COCOMO is not calibrated for EDSI < 2000.

8. **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.

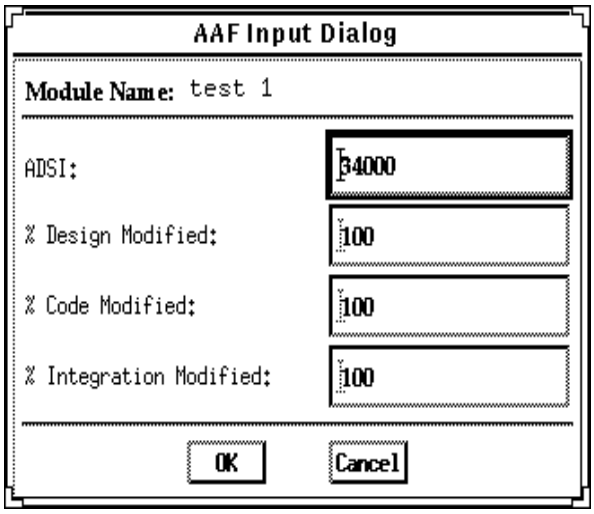
9. **Adaptation Adjusting Factor (AAF) Column** - This column houses the value for AAF for each module. The value can be changed by clicking on the corresponding box to the module. This will result in a window (see figure 1-6). From this window you can change the *ADSI, Percent design modified, Percent code modified* and *Percent integration modified*. Changes can be initiated by clicking on the appropriate box and enter changes directly in the boxes of this window. Changes are confirmed and implemented by clicking on the OK button. To exit the window without making any changes click the CANCEL button.

10. **Effort Adjustment Factor (EAF) Column** - This column displays the product of the 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 and their current ratings. The cost drivers are divided into the groupings: *Product, Computer, Personnel* and *Project*. The ratings for each multiplier can be changed by cycling through the available ratings using either the "+" or "-" buttons 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.

11. **Totals Area** - This area houses the calculated results of all of the modules combined. Within this area is the total EDSI count for the module, the total productivity, the total actual effort (ACT PM), the total instruction cost, the total project cost, the total FSWP and the total estimated schedule for project completion (see each individual column for more information).

FIGURE 1-6 **AAF Dialog Box - Sample**



12. **Microhelp Window** - 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.

13. **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.

14. **Development Mode Button** - This button displays the current mode of estimation. In addition it allows the user to toggle between the three modes of estimation organic, semidetached and embedded "on the fly."

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

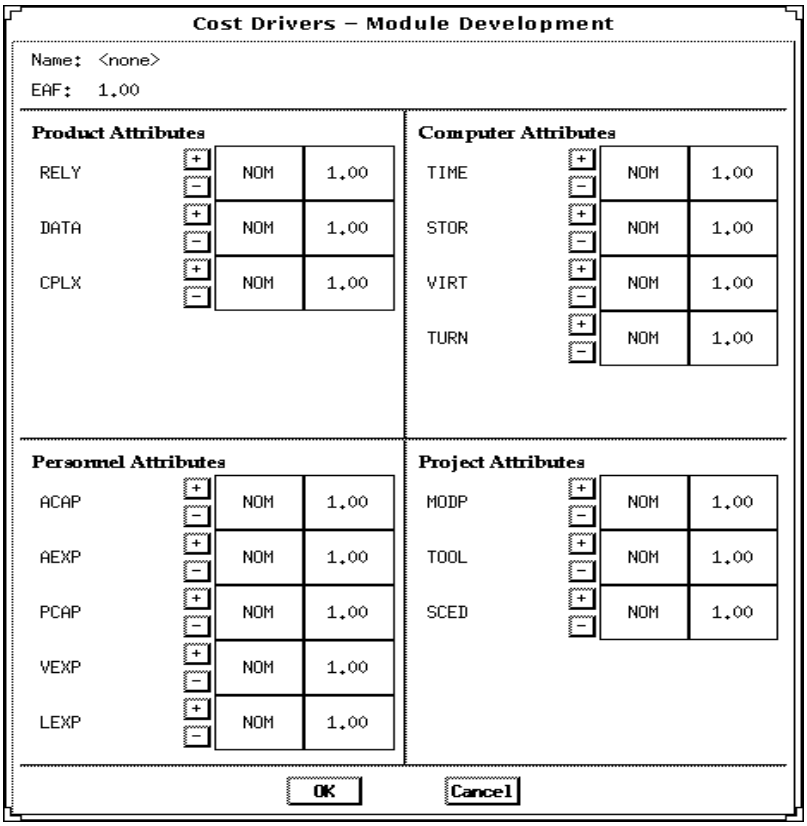
16. **Instruction Cost Column** - This column contains the cost of each instruction. This figure is calculated from EDSI/Cost * Number of modules.

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

18. **Productivity (PROD) Column** - This column contains the calculated result of the individual EDSI divided by the calculated actual schedule estimate in person-months.

19. **Actual Person-Month Development (ACT PM DEV) Column** - This column holds the calculated schedule estimate including the adjusting factors.

FIGURE 1-7 **EAF Dialog Box**

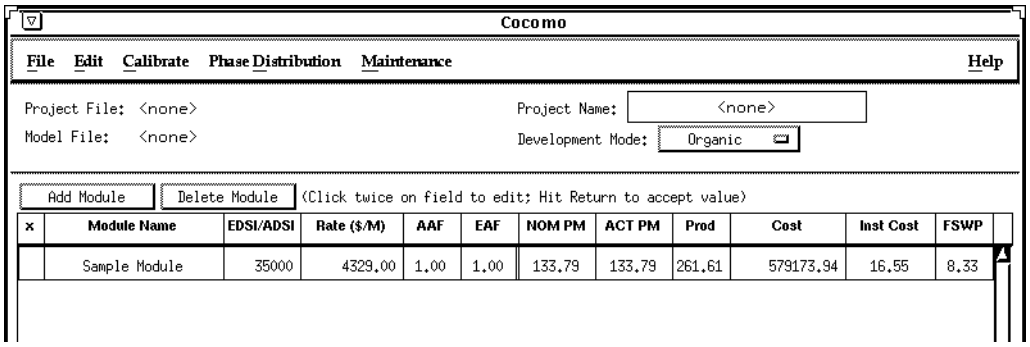


20. **Nominal Person-Month Development (NOM PM DEV) Column** - This column holds the calculated schedule estimate without incorporating the adjusting factors.

1.3 Begin Using COCOMO

To begin entering a new module, click on the "Add Module" button. 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 EDSI and Labor rate may also be given by clicking on the respective field and editing appropriately (see figure 1-8).

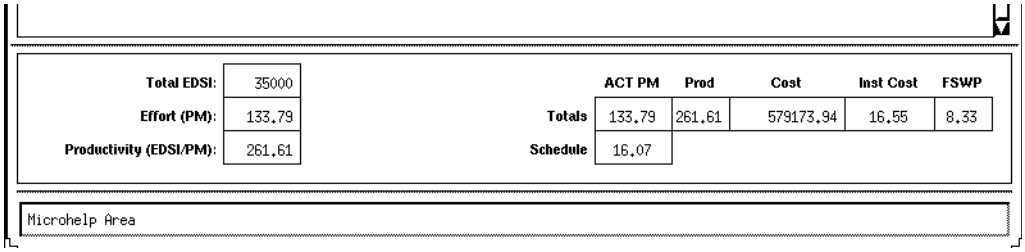
FIGURE 1-8 **Create Sample Module and give values to EDSI and Labor Rate**



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 hit the "Return" key.

All of the final results can be found at the bottom of the CLEF in the Totals area (see figure 1-9).

FIGURE 1-9 **Totals area after calculations have been completed**



Chapter

2

File Menu

The COCOMO file type includes: project file, model file, and report file.

The project file in COCOMO stores a project's data, which include project name, project mode, module name, EDSI, labor rate, adaptation adjustment factor (AAF), values of cost drivers, and COCOMO related calculation results. The COCOMO system gives all project files with ".est" extensions.

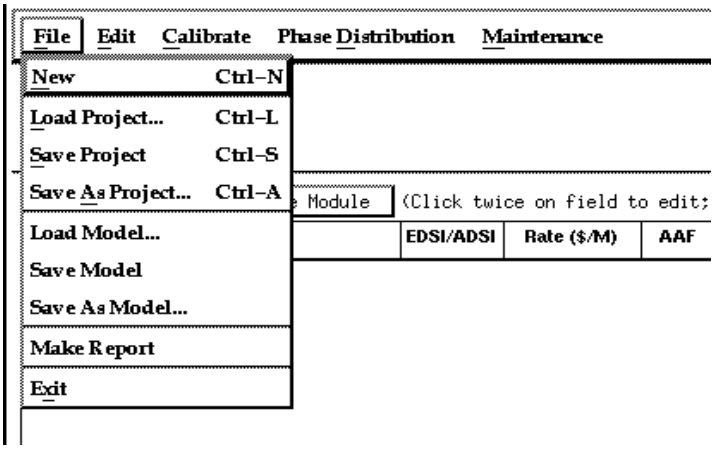
Regarding the model file, as we mentioned in chapter one, COCOMO incorporates 15 predictor factors, or cost driver attributes, which are grouped into four categories: software product attributes, computer attributes, personnel attributes, and project attributes. Each of these cost drivers attributes determines a multiplying factor which estimates the effect of the attribute in software development effort. Besides these cost drivers, COCOMO also has three software development modes, which are organic mode, semidetached mode, and embedded mode. These three modes have different effort and schedule estimating equations respectively. These multiplying factors and effort estimating equations consist of the model of a project. As we said previously, COCOMO has assigned default values and equations for 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 multiplying factors and effort/schedule estimating equations. 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.is a particular model. 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 edited and calculated values of a project. 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, press Meta+F or 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 on COCOMO working window. It also clears previous project file of its contents if there is one on the working window.

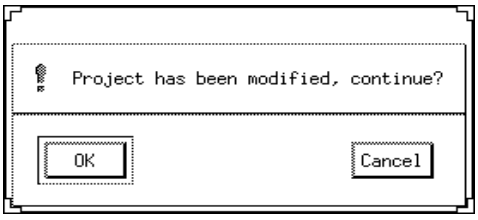
To Create a New Working File

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 OK, otherwise choose Cancel. If the Cancel is selected, a File Save dialog will appear. (see Project Save and Model Save respectively)

2.2 Project Load

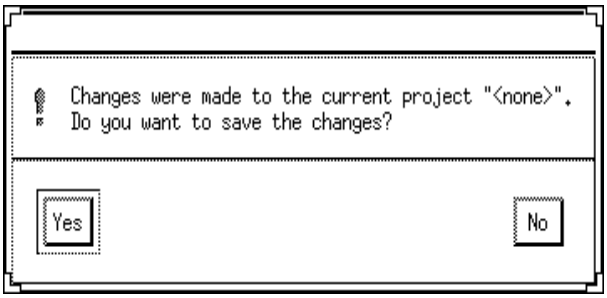
The Project Load option is used to retrieve a project file as well as loading it on the working window.

To Retrieve or Load a Project File

1. Choose Project Load from the File menu with the mouse.

2. If a previous project file has been modified in the working window, the following dialog box will appear.

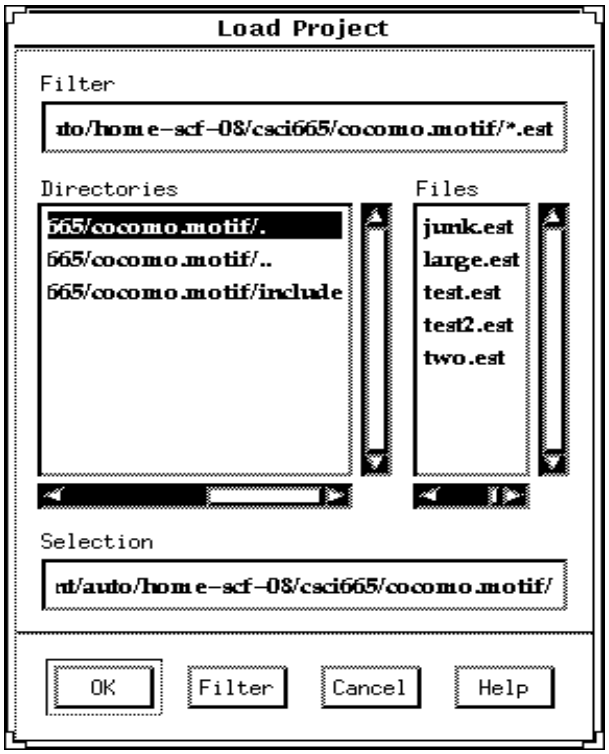
FIGURE 2-3 **Warning Dialog Box**



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

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

FIGURE 2-4 **Project Load Dialog Box**



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 choose an appropriate directory.

5. Choose desired directory for file loading

Look at the filter input box. The path found in this box represents the path to be searched for loading a project file. This path will be changed after each directory change. To select the desired directory, click the appropriate directory choice from the Directories scroll list, then click the "Filter" button. At this point, the files located under this directory will be listed in the Files scroll list.

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 are beyond the window scope, the scroll bar can be used to look at all items.

2.3 Project Save

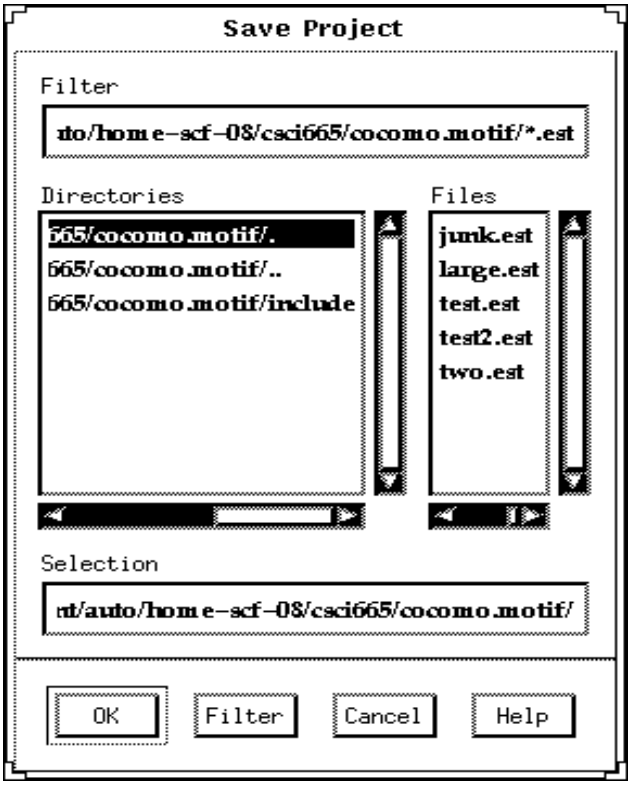
The Project Save option is used to store the results of the current COCOMO project as a file with ".est" extension.

To Store the Results of Current Project

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

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

2.4 Project Save As

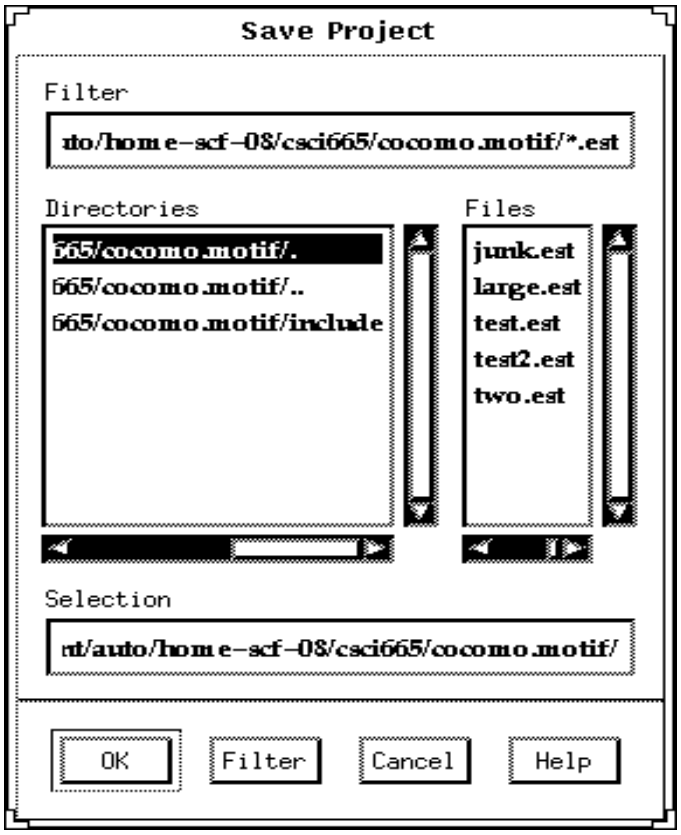
The Project Save As option is to store the current project as a COCOMO project file, which has a file name different from current file.

To Store Current Project With different File Name

1. Choose Project Save As from the File menu with mouse.

2. The Project Save dialog box will appear, as seen in Figure 2-6.

FIGURE 2-6 **Project Save 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 the filename in the SELECTION box.

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

2.5 Model Load

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

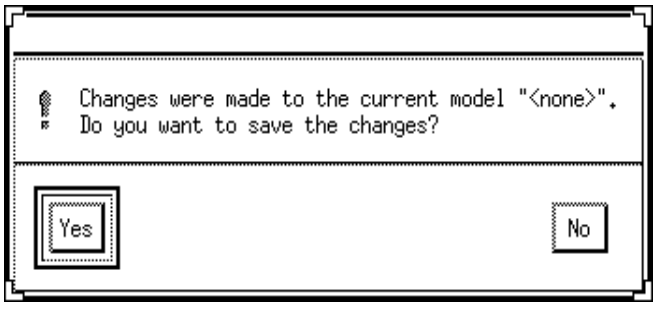
The Model Load option is used to retrieve a model file as well as loading it on current project.

To Retrieve or Load a Model File

1. Choose Model Load from the File menu.

2. If a previous model has been modified in current project, the following dialog box will appear.

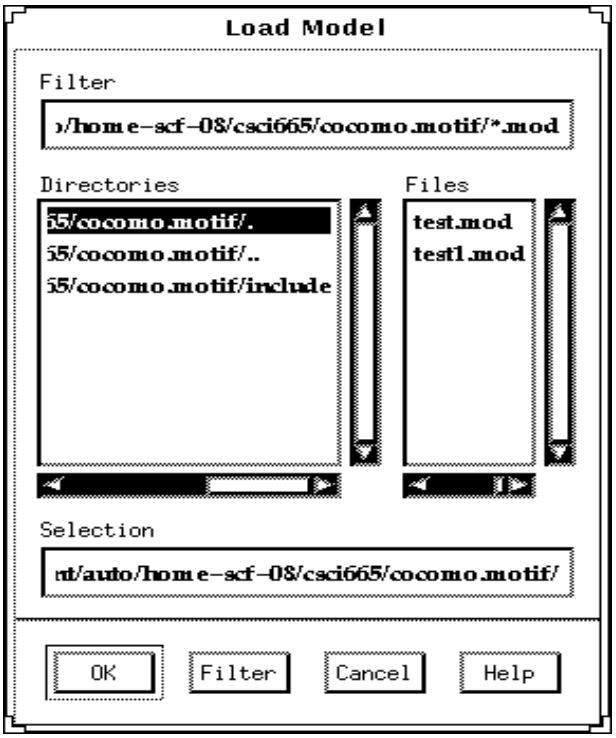
FIGURE 2-7 **Warning Dialog Box**



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

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

FIGURE 2-8 **Model Load 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, it is necessary to choose an appropriate directory.

5. Choose desired directory for file loading

Look at the filter input box. The path found in this box represents the path to be searched for loading a model file. This path will be changed after each directory change. To select the desired directory, click the appropriate directory choice from the Directories scroll list, then click the "Filter" button. At this point, the files located under this directory will be listed in the Files scroll list.

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 working window.

2.6 Model Save

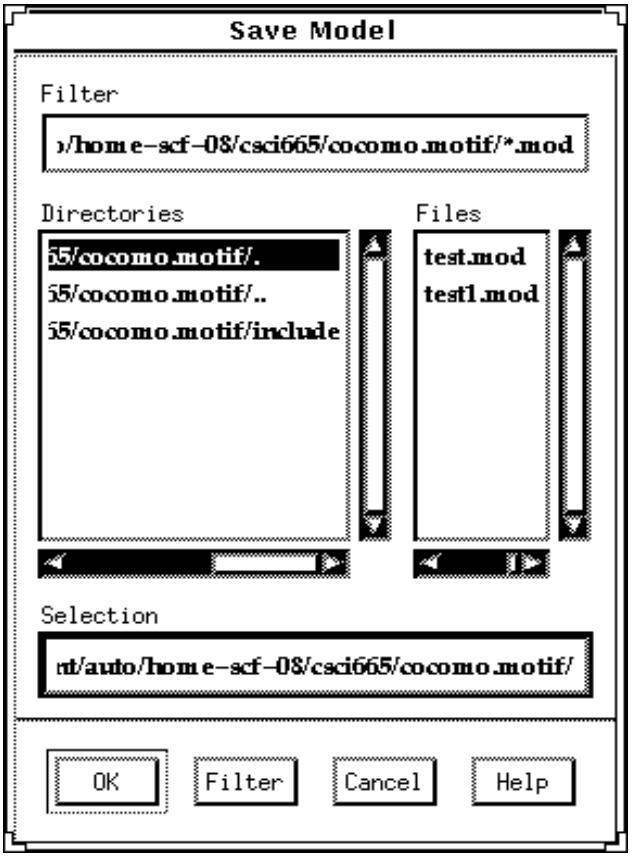
The Model Save option is used to store the results of the current COCOMO model as a file with ".mod" extension.

To Store the Results of Current Model

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

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

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



3. Look at the Files scroll window. If the file saving is to update (overwrite) a existing

model 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 model file, choose the desired directory, then type in the filename.

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

2.7 Model Save As

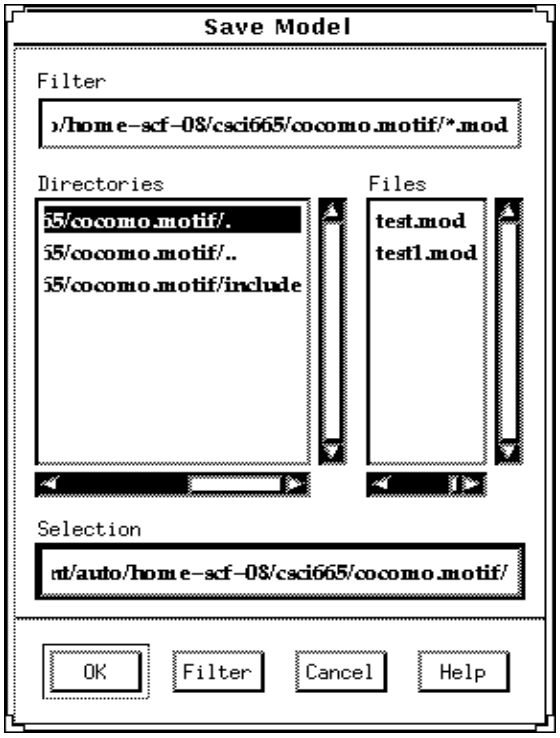
The Model Save As option is to store the current model as a COCOMO model file, which has a file name different from current model.

To Store Current Model With different File Name

1. Choose Model Save As from the File menu, or press Meta+E.

2. The Model Save dialog box will appear, as seen in Figure 2-10.

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



3. Look at the Files scroll window. If the file saving is to update (overwrite) a existing model 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 model file, choose the desired directory, then type in the filename in the SELECTION box.

5. After the desired filename is selected or inputted, 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

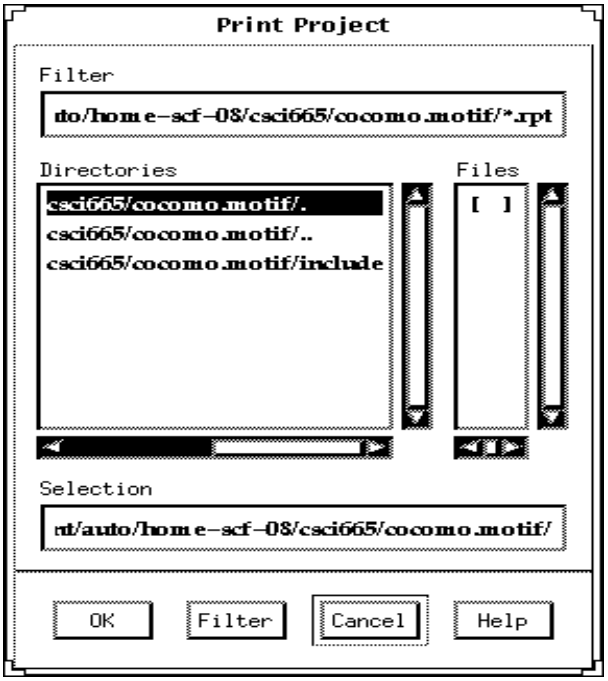
The Make Report option creates a COCOMO report in the form of a text file for printing.

To Create Project Report

1. Choose Make Report from the File menu, or press R.

2. The Make Report dialog box will appear, as seen in Figure 2-11.

FIGURE 2-11 **Make Report Dialog Box**



3. Look at the Files scroll window. If the file saving is to update (overwrite) a existing report 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 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, executing the local commands for your system in order to send the file for printing.

2.9 Exit

The Exit option leaves COCOMO system.

To Leave COCOMO System

1. Choose Exit from the File menu with mouse.

2. Leave COCOMO system.

Chapter

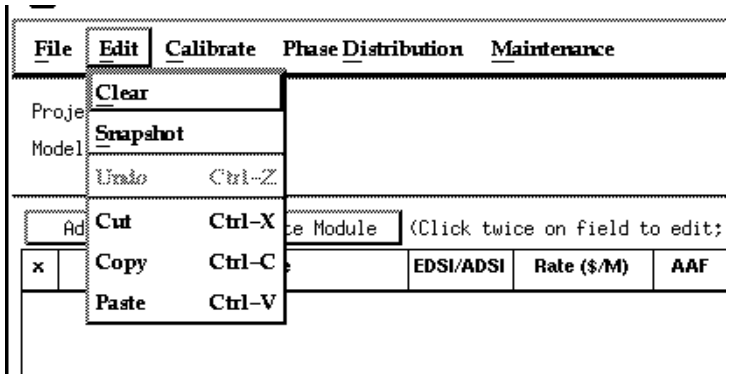
3

Edit Menu

The Edit Menu option supplies several useful commands which will enable you to establish a project more conveniently. With the Edit Menu options, the entering of modules will be easier.

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

FIGURE 3-1 **Edit Menu**



3.1 Clear

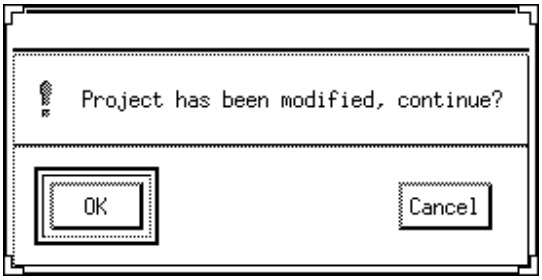
The Clear option erases all modules of the current project on the working window.

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.

FIGURE 3-2 **Warning Dialog Box**



2. If you really want to clear, click OK. If not, click CANCEL.

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

3.2 Snapshot

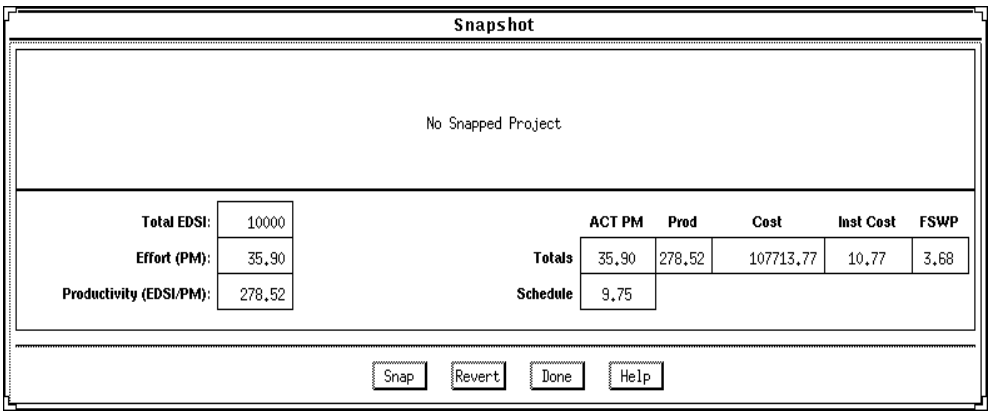
The Snapshot option enables users to compare the effort estimation change for a project so that he/she can decide to apply the change or not. This function makes COCOMO more convenient and powerful for software development effort estimation.

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 3-4.

FIGURE 3-3 **Snapshot Dialog Box**



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.

3. 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.

4. When finished, click Done button.

3.3 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.

2. The changed module will go back to previous status.

3.4 Cut

The Cut option copies a module into the cut buffer and removes it from 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 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.

3. The cut module disappears

3.5 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 CLEF area.

2. Press Ctrl-C, or choose Copy from the Edit menu with mouse.

3. The cross sign in the check box disappears.

3.6 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 where 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. Press Ctrl-V, or choose Paste from the Edit menu with 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

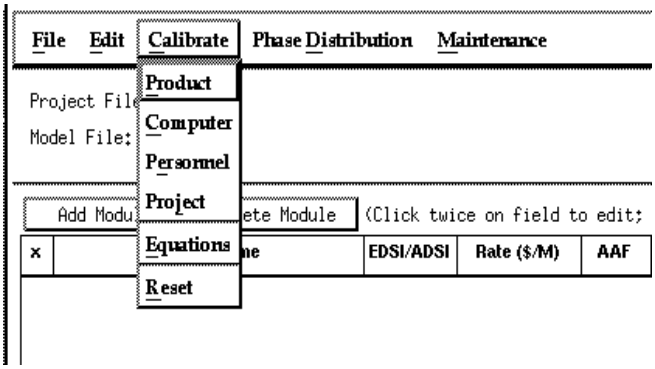
4

Calibrate Menu

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

To choose the Calibrate menu and its options, press Meta+C or click on Calibrate with the mouse. The Calibrate menu will appear as Figure 4-1.

FIGURE 4-1 **Calibrate Menu**

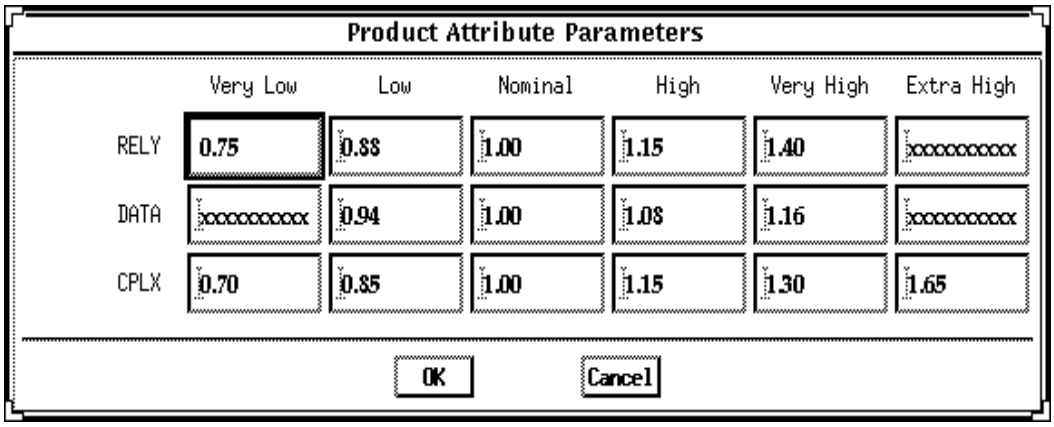


4.1 Product

The Product option displays three cost drivers: RELY, DATA, and CPLX, and their corresponding ratings and multiplier values.

Select Product from the Calibrate menu with mouse. The Product Dialog Box will appear as Figure 4-2.

FIGURE 4-2 **Product Dialog Box**



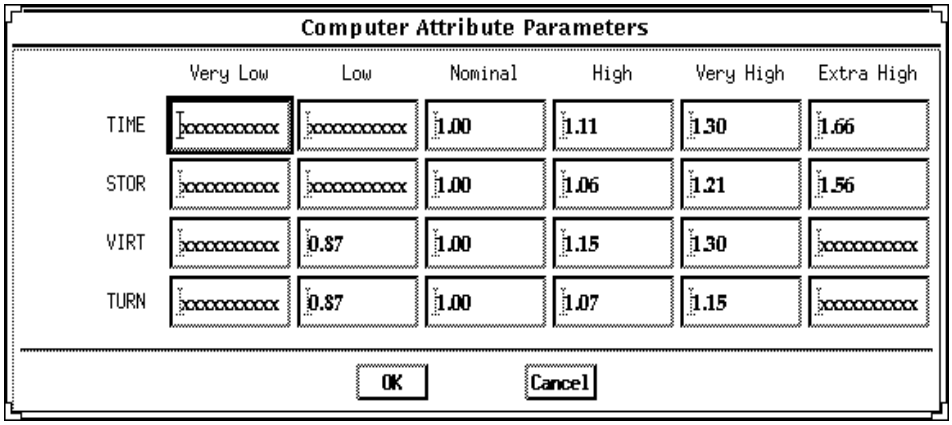
To modify these values, go straight to those edit boxes and type new values. When finishing the modification, click the OK button

4.2 Computer

The Computer option displays four cost drivers: TIME, STOR, VIRT, and TURN, and their corresponding ratings and multiplier values.

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

FIGURE 4-3 **Computer Dialog Box**



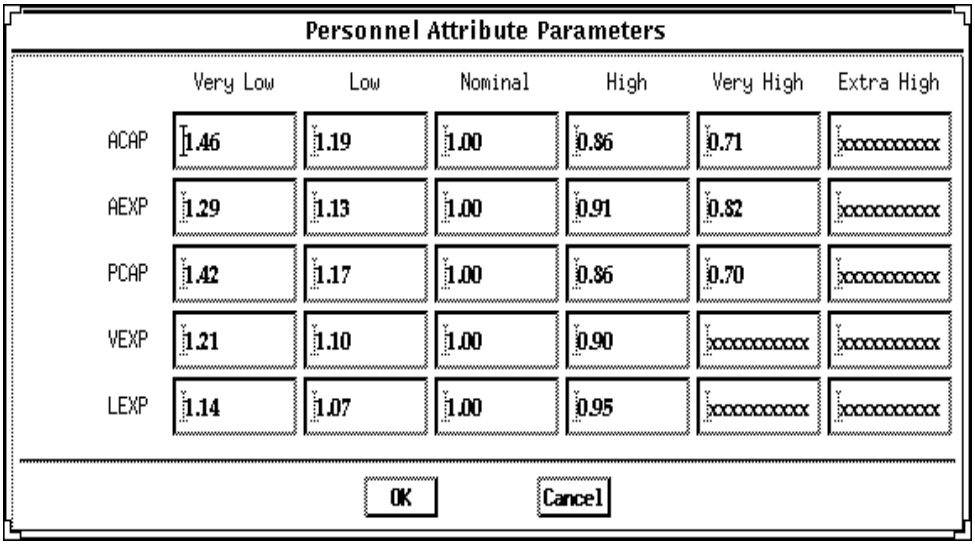
To modify these values, go straight to those edit boxes and type new values. When finishing the modification, click the OK button.

4.3 Personnel

The Personnel option displays five cost drivers: ACAP, AEXP, PCAP, VEXP, and LEXP, and their corresponding ratings and multiplier values.

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

FIGURE 4-4 **Personnel Dialog Box**



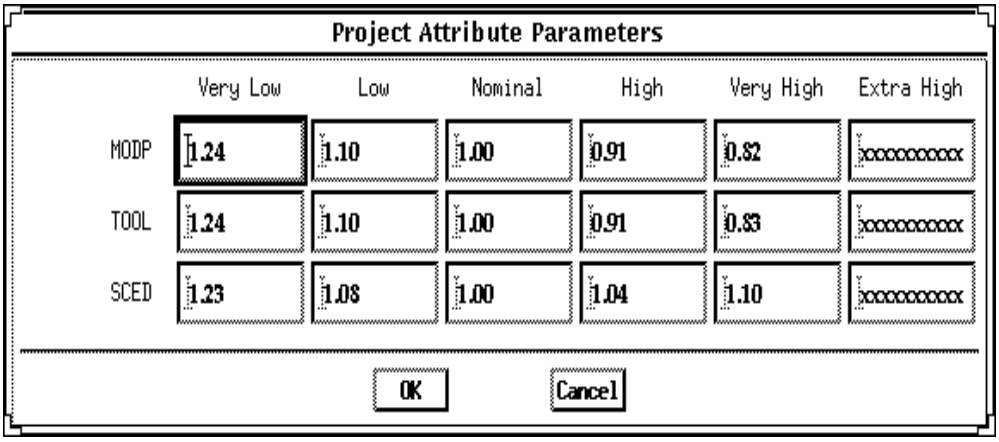
To modify these values, go straight to those edit boxes and type new values. When finishing the modification, click the OK button.

4.4 Project

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

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

FIGURE 4-5 **Project Dialog Box**



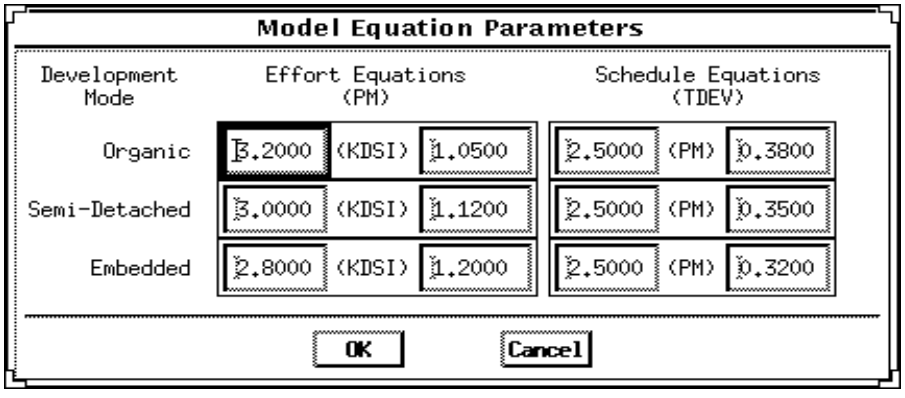
To modify these values, go straight to those edit boxes and type new values. When finishing the modification, click the OK button.

4.5 Equation

The Equation option displays effort and schedule equations of three development modes, organic, semi-detached, and embedded modes.

Select Equation from the Calibrate menu with mouse. The Equation Dialog Box will appear as Figure 4-6.

FIGURE 4-6 **Equation Dialog Box**



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

4.6 Reset

The Reset option reset the values of multiplying factors and effort/schedule estimating equations of current project.

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

Chapter

5

Phase Distribution

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

Plan & 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.

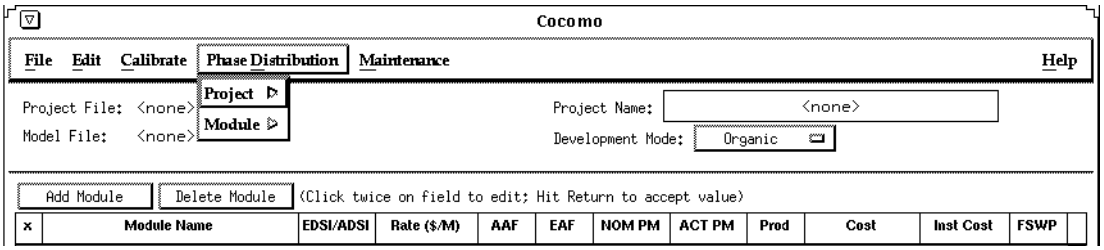
Design - In this phase, a hardware/software architecture, control structure and data structure for the product are defined. A draft of the users' manual and test plans are also created during this phase.

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 5.1 and 5.2 in this chapter.

FIGURE 5-1 **Phase Distribution Sub-menu**



5.1 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 5-1). 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 5.1.1 - 5.1.4, 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

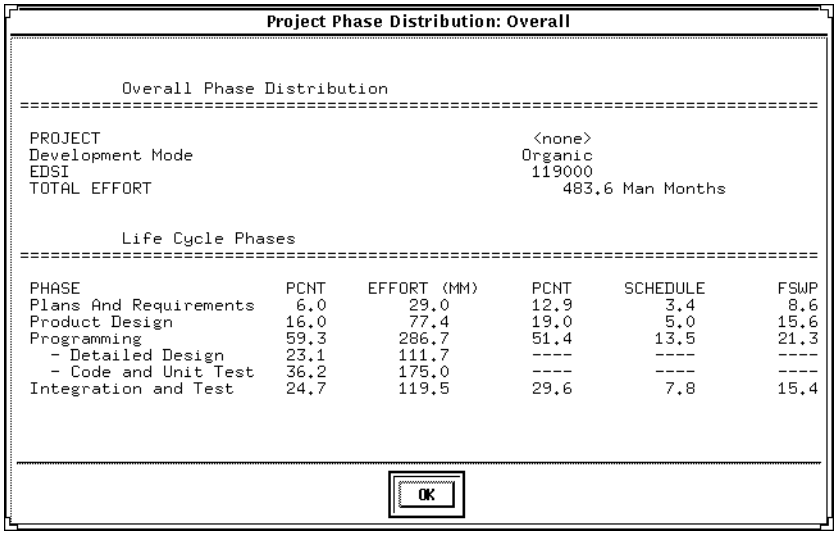
program support library, development and monitoring of end item acceptance plan. Quality assurance includes development and monitoring of project standards, and technical audits of software products and processes.

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

5.1.1 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 5-2). This window displays the project name, project EDSI, and the total estimated effort for the project. Looking at figure 5-1, this information can be seen in the upper left corner of the window.

FIGURE 5-2 **Phase Distribution window displaying a sample project's overall phase distribution**



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 5-2, 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

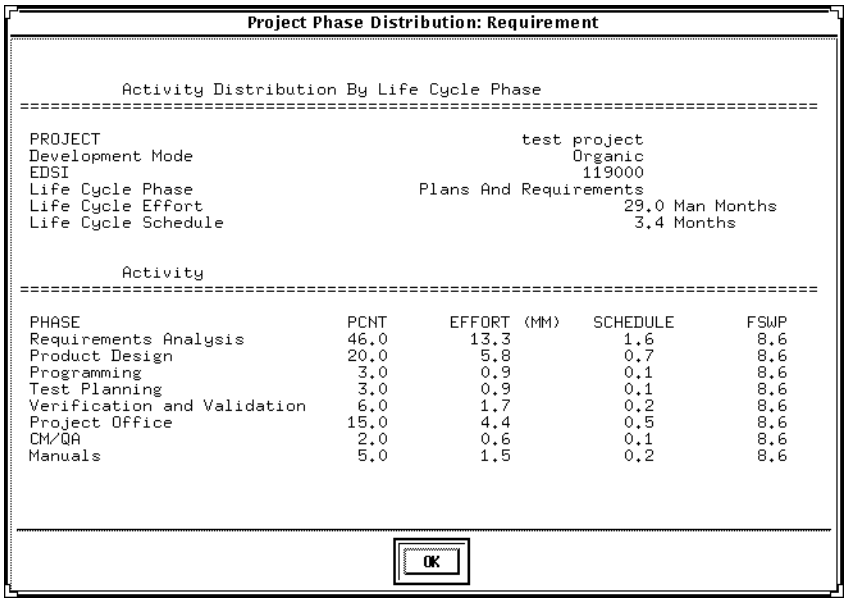
column displays the estimated number of personnel needed for phase completion (FSWP).

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.

5.1.2 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 5-3 is displayed. This window displays the following information: project name, the total project EDSI, the total estimated project effort, the total estimated project schedule. In addition the window displays the sub-phases requirements analysis, product design, programming, test planning, verification & validation, project office, CM/QA, and manuals. These sub-phases are accompanied with a percentage of the phase effort that they encompass, the estimated effort, schedule and FSWP for the sub-phases' completion as shown in figure 5-3. To exit from this window click the OK button.

FIGURE 5-3 **Plans and Requirements Phase window for the overall project**

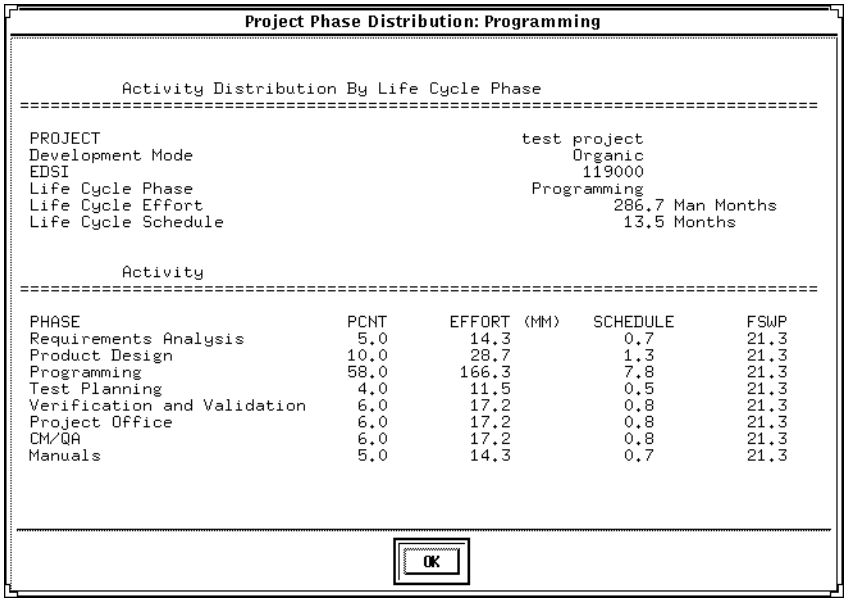


5.1.3 Programming Project Phase

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 5-4 is displayed. This window displays the following information: project name, the total project EDSI, the total estimated project effort, the total estimated project

schedule. In addition the window displays the sub-phases requirements analysis, product design, programming, test planning, verification & validation, project office, CM/QA, and manuals. These sub-phases are accompanied with a percentage of the phase effort that they encompass, the estimated effort, schedule and FSWP for the sub-phases' completion as shown in figure 5-4. To exit from this window click the OK button.

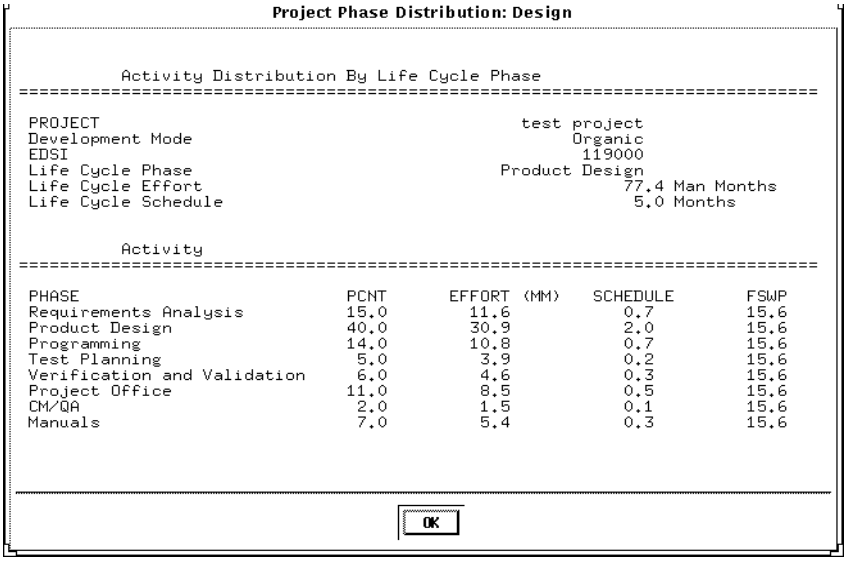
FIGURE 5-4 **Programming Phase window for overall project**



5.1.4 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 5-5 is displayed. This window displays the following information: project name, the total project EDSI, the total estimated project effort, the total estimated project schedule. In addition the window displays the sub-phases requirements analysis, product design, programming, test planning, verification & validation, project office, CM/QA, and manuals. These sub-phases are accompanied with a percentage of the phase effort that they encompass, the estimated effort, schedule and FSWP for the sub-phases' completion as shown in figure 5-5. To exit from this window click the OK button.

FIGURE 5-5 **Product Design window for overall project**



Project Phase Distribution: Design

Activity Distribution By Life Cycle Phase

PROJECT	test project
Development Mode	Organic
EDSI	119000
Life Cycle Phase	Product Design
Life Cycle Effort	77.4 Man Months
Life Cycle Schedule	5.0 Months

Activity

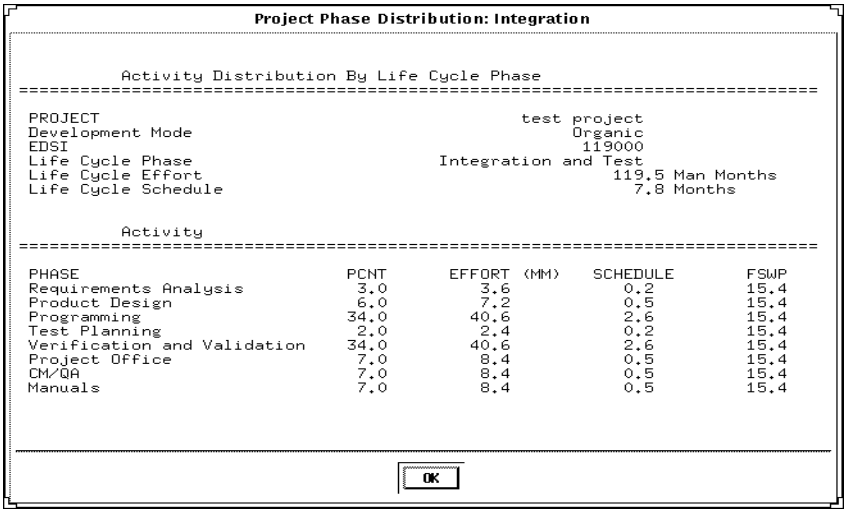
PHASE	PCNT	EFFORT (MM)	SCHEDULE	FSWP
Requirements Analysis	15.0	11.6	0.7	15.6
Product Design	40.0	30.9	2.0	15.6
Programming	14.0	10.8	0.7	15.6
Test Planning	5.0	3.9	0.2	15.6
Verification and Validation	6.0	4.6	0.3	15.6
Project Office	11.0	8.5	0.5	15.6
CM/QA	2.0	1.5	0.1	15.6
Manuals	7.0	5.4	0.3	15.6

OK

5.1.5 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 5-6 is displayed. This window displays the following information: project name, the total project EDSI, the total estimated project effort, the total estimated project schedule. In addition the window displays the sub-phases requirements analysis, product design, programming, test planning, verification & validation, project office, CM/QA, and manuals. These sub-phases are accompanied with a percentage of the

FIGURE 5-6 **Integration and Test window for overall project**



Project Phase Distribution: Integration

Activity Distribution By Life Cycle Phase

PROJECT	test project
Development Mode	Organic
EDSI	119000
Life Cycle Phase	Integration and Test
Life Cycle Effort	119.5 Man Months
Life Cycle Schedule	7.8 Months

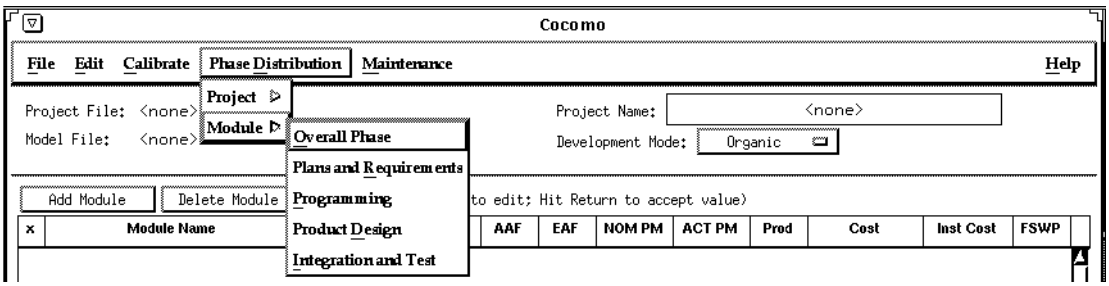
Activity

PHASE	PCNT	EFFORT (MM)	SCHEDULE	FSWP
Requirements Analysis	3.0	3.6	0.2	15.4
Product Design	6.0	7.2	0.5	15.4
Programming	34.0	40.6	2.6	15.4
Test Planning	2.0	2.4	0.2	15.4
Verification and Validation	34.0	40.6	2.6	15.4
Project Office	7.0	8.4	0.5	15.4
CM/QA	7.0	8.4	0.5	15.4
Manuals	7.0	8.4	0.5	15.4

OK

phase effort that they encompass, the estimated effort, schedule and FSWP for the sub-phases' completion as shown in figure 5-6. To exit from this window click the OK button.

FIGURE 5-7 **Phase Distribution Module Sub-menu**



5.2 Module Phase Distribution

Four formats for viewing will appear in another menu: overall phase, plan & requirements, programming, and integration & test (see figure 5-7). Each of these menu selections will be discussed in sections 5.2.1 - 5.2.4, 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, Module 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?"

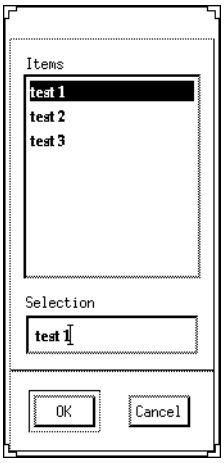
Module Office Functions: Module level management functions. Includes Module level planning and control, contract and subcontract management, and customer interface.

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 5-8). 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.

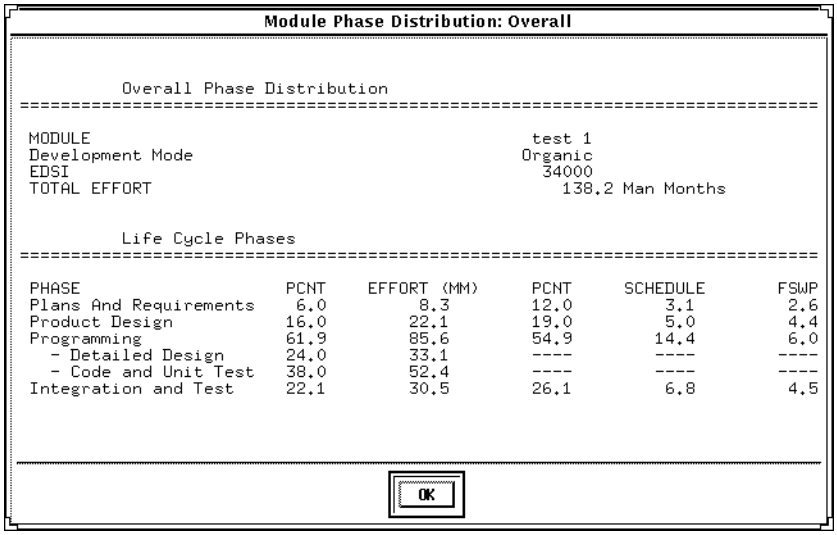
FIGURE 5-8 **Module selection window**



5.2.1 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 5-9). To exit from this window click the OK button.

FIGURE 5-9 **Phase Distribution window displaying a sample Module's overall phase distribution**



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 5-9, 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.

5.2.2 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 5-10 is displayed. This window displays the following information: Module name, the total Module EDSI, the total estimated Module effort, the total estimated Module schedule. In addition the window displays the sub-phases requirements analysis, product design, programming, test planning, verification & validation, Module office, CM/QA, and manuals. These

sub-phases are accompanied with a percentage of the phase effort that they encompass, the estimated effort, schedule and FSWP for the sub-phases' completion as shown in figure 5-10. To exit from this window click the OK button.

5.2.3 Programming Module Phase

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 5-11 is displayed. This window displays the following information: Module name, the total Module EDSI, the total estimated Module effort, the total estimated Module schedule. In addition the window displays the sub-phases requirements analysis, product design, programming, test planning, verification & validation, Module office, CM/QA, and manuals. These sub-phases are accompanied with a percentage of the phase effort that they encompass, the estimated effort, schedule and FSWP for the sub-phases' completion as shown in figure 5-11. To exit from this window click the OK button.

FIGURE 5-10 **Plans and Requirements Phase window for the overall Module**

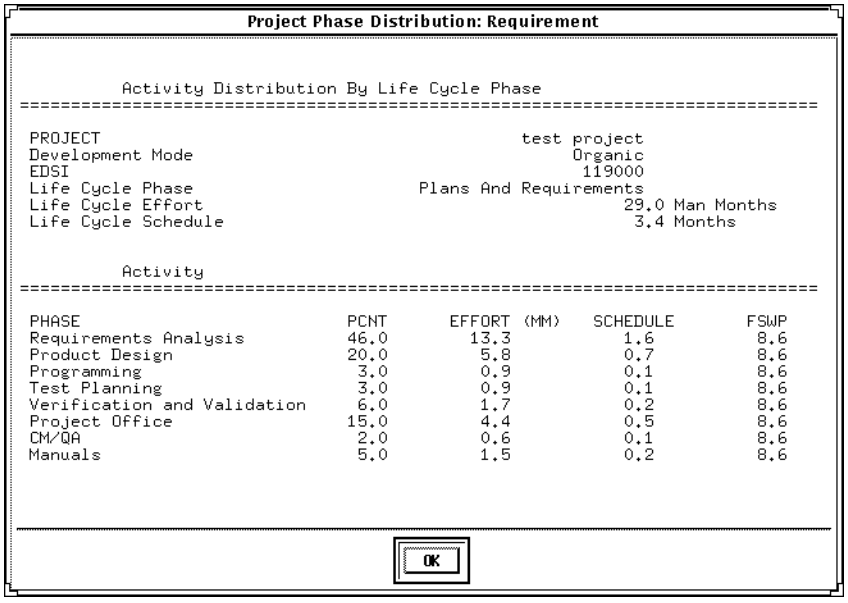
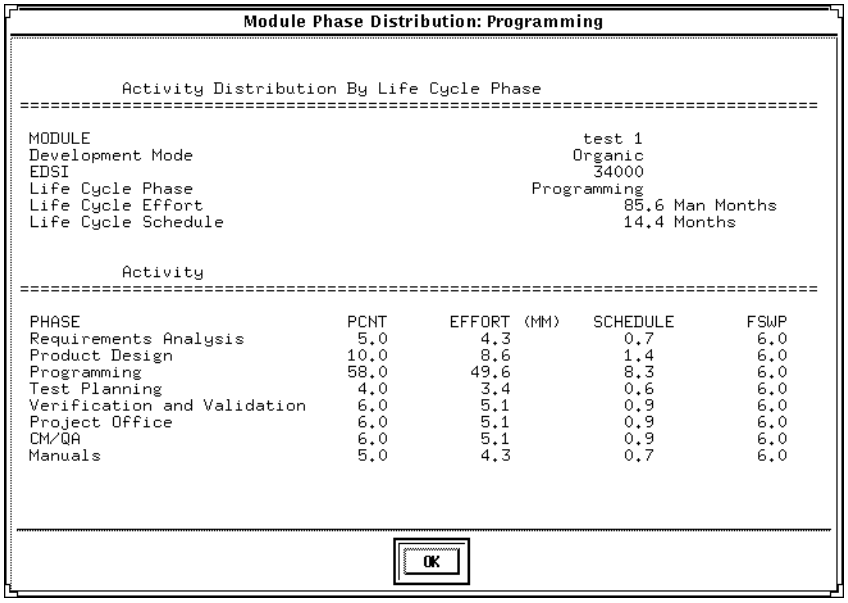


FIGURE 5-11 **Programming Phase window for overall Module**



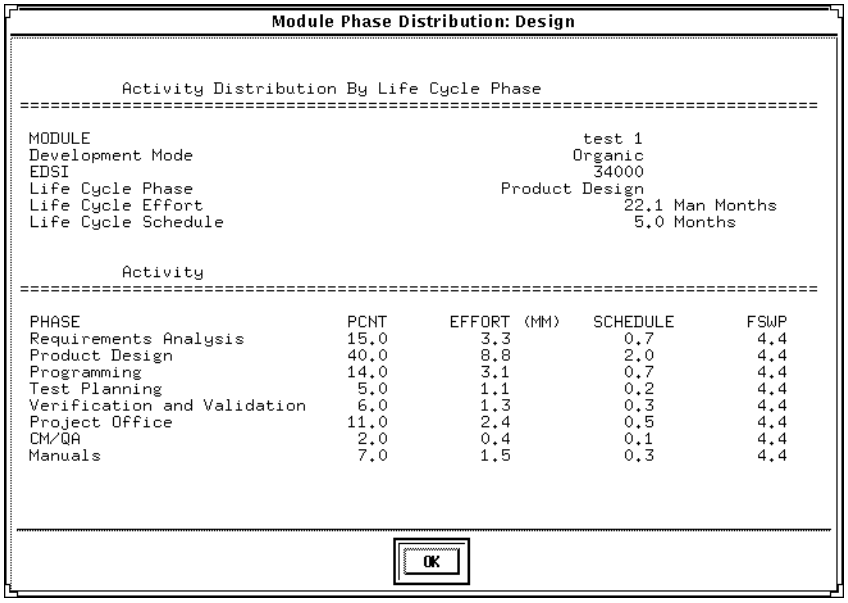
5.2.4 Product Design Phase

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 Module Phase distribution menu, the window shown in figure 5-12 is displayed. This window displays the following information: Module name, the total Module EDSI, the total estimated Module effort, the total estimated Module schedule. In addition the window displays the sub-phases requirements analysis, product design, programming, test planning, verification & validation, Module office, CM/QA, and manuals. These subphases are accompanied with a percentage of the phase effort that they encompass, the estimated effort, schedule and FSWP for the sub-phases' completion as shown in figure 5-12. To exit from this window click the OK button.

5.2.5 Integration and Test Module 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 Module Phase distribution menu, the window shown in figure 5-13 is displayed. This window displays the following information: Module name, the total Module EDSI, the total estimated Module effort, the total estimated Module schedule. In addition the window displays the sub-phases requirements analysis, product design, programming, test planning, verification & validation, Module office, CM/QA, and manuals. These subphases are accompanied with a percentage of the phase effort that they encompass, the estimated effort, schedule and FSWP for the sub-phases' completion as shown in figure 5-13. To exit from this window click the OK button.

FIGURE 5-12 **Product Design window for overall Module**



Module Phase Distribution: Design

Activity Distribution By Life Cycle Phase

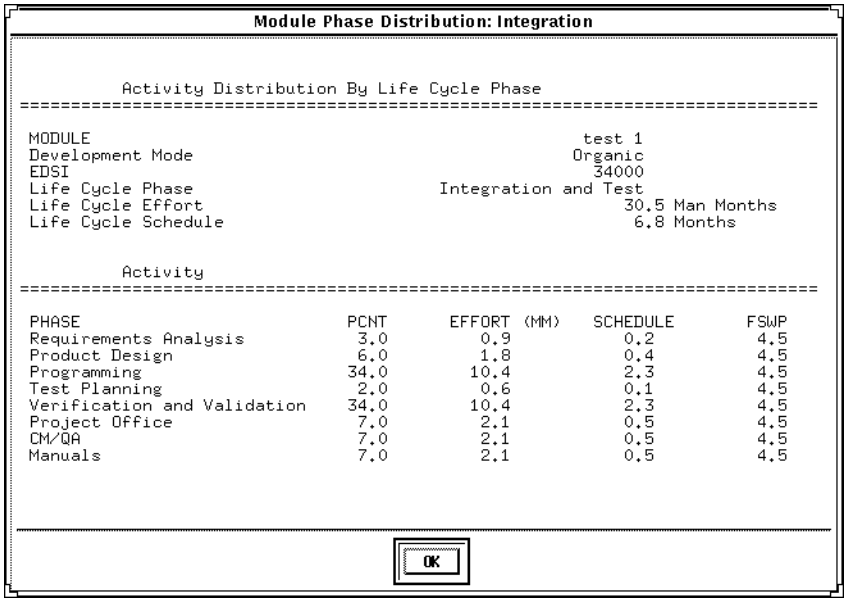
MODULE	test 1
Development Mode	Organic
EDSI	34000
Life Cycle Phase	Product Design
Life Cycle Effort	22.1 Man Months
Life Cycle Schedule	5.0 Months

Activity

PHASE	PCNT	EFFORT (MM)	SCHEDULE	FSWP
Requirements Analysis	15.0	3.3	0.7	4.4
Product Design	40.0	8.8	2.0	4.4
Programming	14.0	3.1	0.7	4.4
Test Planning	5.0	1.1	0.2	4.4
Verification and Validation	6.0	1.3	0.3	4.4
Project Office	11.0	2.4	0.5	4.4
CM/QA	2.0	0.4	0.1	4.4
Manuals	7.0	1.5	0.3	4.4

OK

FIGURE 5-13 **Integration and Test window for overall Module**



Module Phase Distribution: Integration

Activity Distribution By Life Cycle Phase

MODULE	test 1
Development Mode	Organic
EDSI	34000
Life Cycle Phase	Integration and Test
Life Cycle Effort	30.5 Man Months
Life Cycle Schedule	6.8 Months

Activity

PHASE	PCNT	EFFORT (MM)	SCHEDULE	FSWP
Requirements Analysis	3.0	0.9	0.2	4.5
Product Design	6.0	1.8	0.4	4.5
Programming	34.0	10.4	2.3	4.5
Test Planning	2.0	0.6	0.1	4.5
Verification and Validation	34.0	10.4	2.3	4.5
Project Office	7.0	2.1	0.5	4.5
CM/QA	7.0	2.1	0.5	4.5
Manuals	7.0	2.1	0.5	4.5

OK

Chapter

6

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, SCED and MODP factors which will be discussed in greater detail later in this chapter). The maintenance calculations are heavily based upon the annual change traffic (ACT). The ACT is fraction of the software product’s source instructions that are changed per year, where the changes are categorized into code added and code modified (see figure 6-1). This change takes the form of either additions or modifications.

FIGURE 6-1 **Annual Change Traffic Equation**

$$ACT = \frac{Added\dot{}\,(percent) + Modified\,(percent)}{100}$$

The maintenance equation that calculates the maintenance effort is displayed in figure 6-2. This equation contains four components: the number of years of maintenance, the ACT value, and the nominal number of person-months needed for the development of the software product

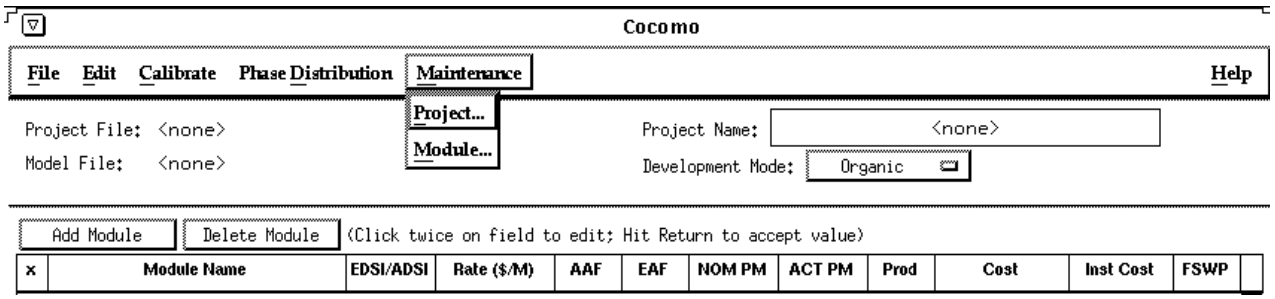
FIGURE 6-2 **Maintenance Effort Equation**

$$(PM)_{AM} = (\#\ of\ years)*(ACT)*(PM)_{NOM}*(EAF)_{M}$$

and the product of the cost drivers for maintenance. As stated previously, three cost drivers for maintenance differ from development. Those cost drivers are software reliability (RELY), modern programming practices (MODP) and schedule (SCED). The reason for the change in MODP, RELY is that increased investment in software reliability and use of modern programming practices during software development have a strong positive effect upon the maintenance stage. The SCED attribute is controlled by the number of years value entered by the user. As a result the SCED driver is no longer editable in the EAF window, but is calculated from the user inputted 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 upon either an entire project or an individual module (see figure 6-3). These separate options are discussed in section 6.1 and 6.2

FIGURE 6-3 **Maintenance sub-menu**

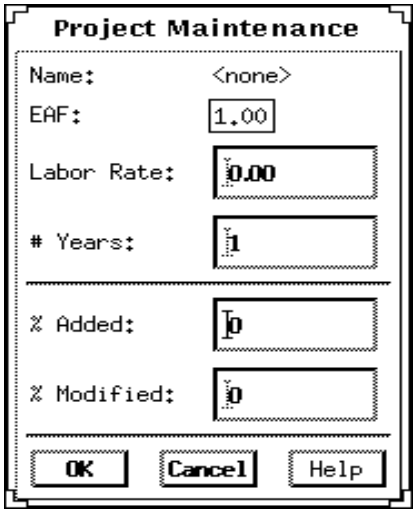


6.1 Project Maintenance

In order to view the maintenance estimation calculations for an entire project, the user can click on Project under the Maintenance menu (see figure 6-3). Upon clicking upon 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 figure 6-4).

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

FIGURE 6-4 **Project Maintenance Dialog Box**



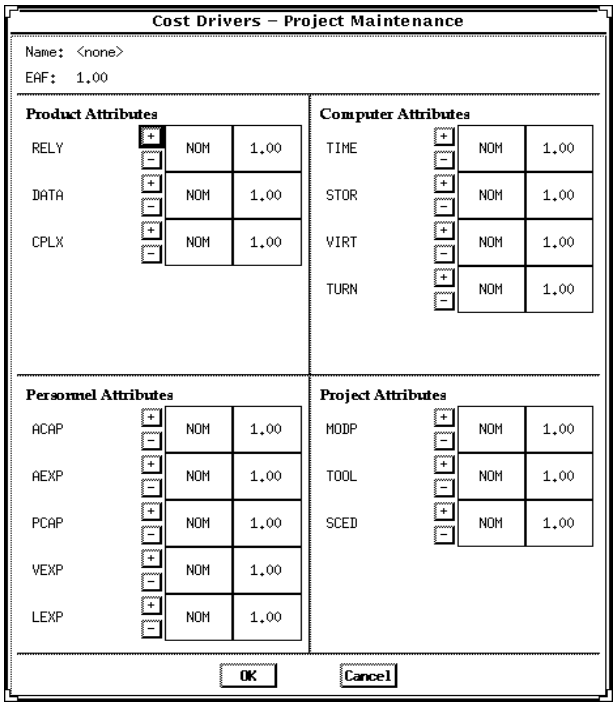
will result in the appearance of an EAF dialog box where the cost driver ratings can be changed as described in the introduction (see figure 6-5).

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 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 6-5 **Project Maintenance EAF Dialog Box**



When the OK button is clicked in the Project Maintenance Dialog Box, a window displaying the first of four pages that contains the project name, the current development mode, the total number of source instructions for development of the project (EDSI) hat is loaded in the CLEF, the nominal effort of the project, the actual effort of the project, 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 6-6) and the calculated annual change traffic.

The second page of the maintenance window can be seen by clicking upon the Next button. It contains the settings for the 14 cost drivers, SCED is not applicable (see figure 6-7).

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 KDSI (EDSI * 10^3), 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(KDSI/FSWP) and the total cost for maintenance for the year.

FIGURE 6-6 **Project Maintenance window (page 1)**

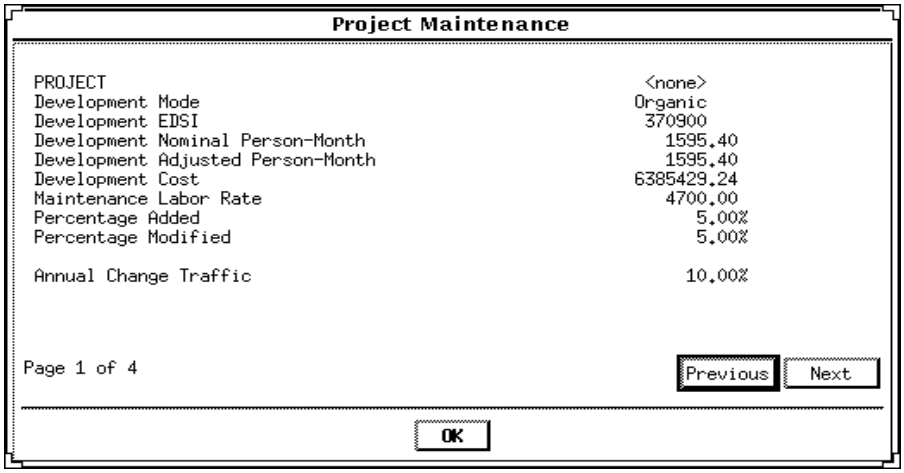
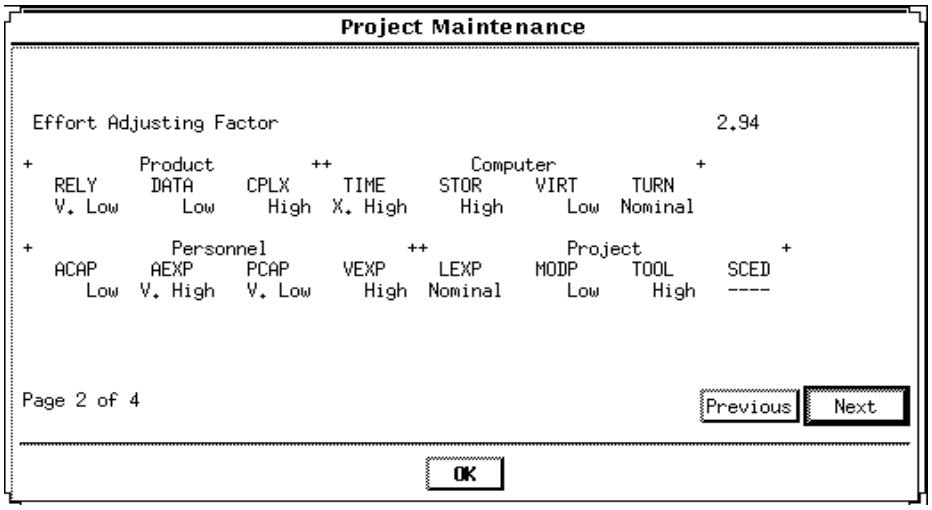


FIGURE 6-7 **Project Maintenance window (page 2)**



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 6-9). This first displays the total number 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 6-8 **Project Maintenance window (page 3)**

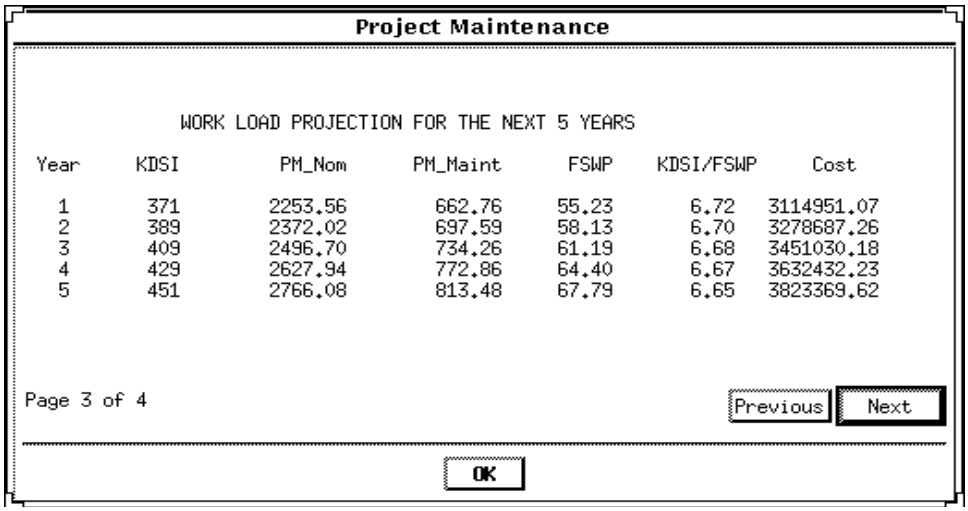
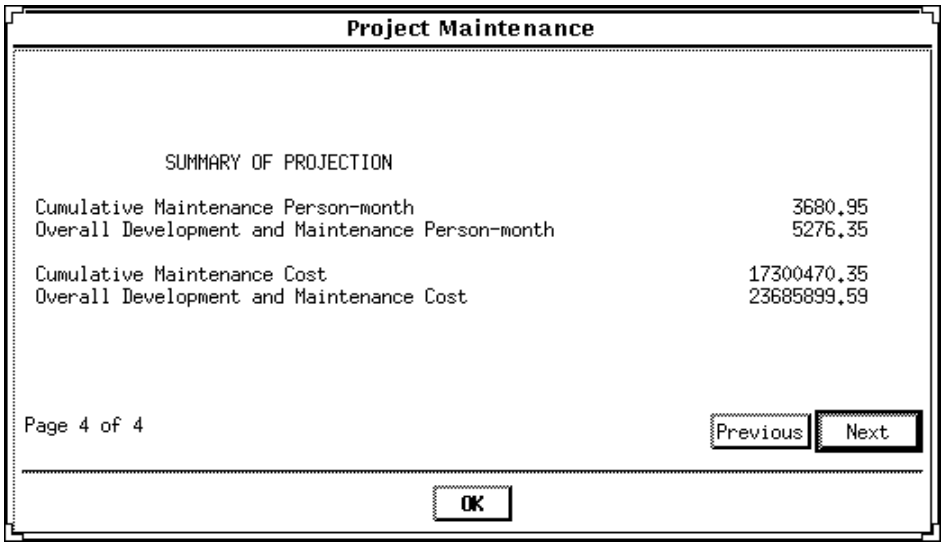
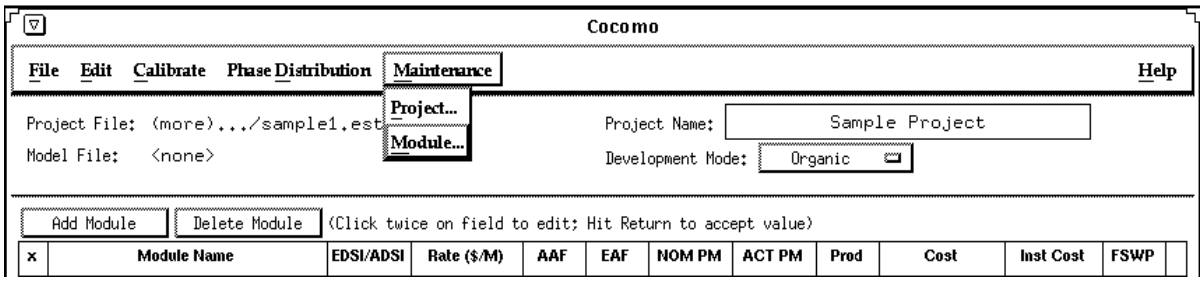


FIGURE 6-9 **Project Maintenance window (page 4)**



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

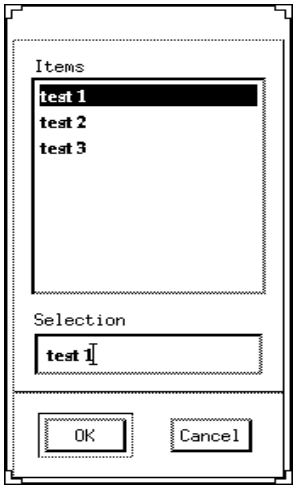
FIGURE 6-10 **Maintenance Sub-menu**



6.2 Module Maintenance

In order to view the maintenance estimation calculations for an entire module, the user can click on Module under the Maintenance menu (see figure 6-10). 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 6-11).

FIGURE 6-11 **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 6-12).

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 (see figure 6-13).

FIGURE 6-12 **Module Maintenance Dialog Box**

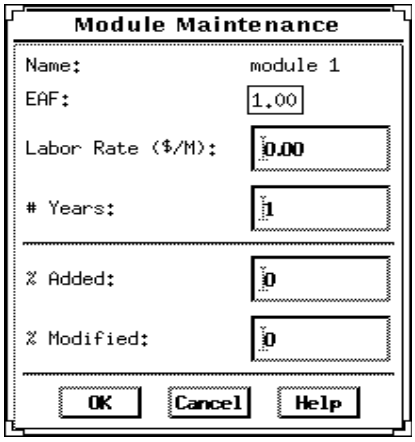
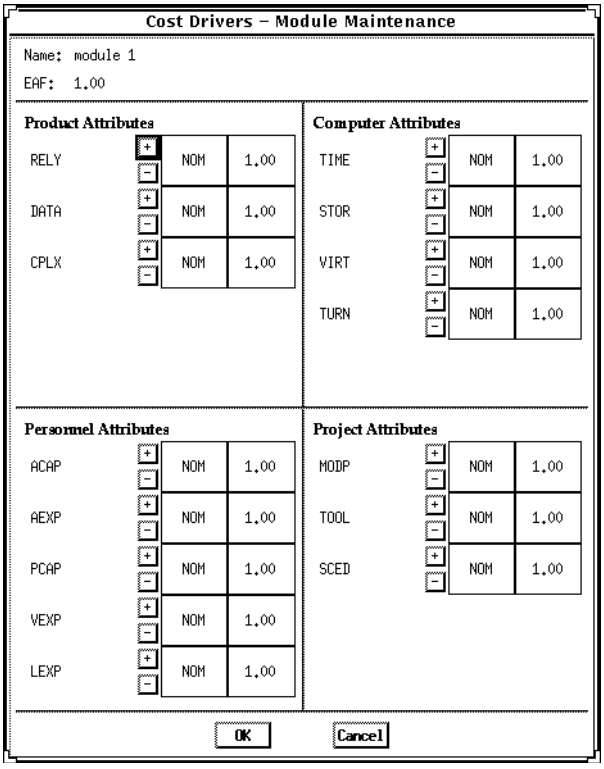


FIGURE 6-13 **Module Maintenance EAF Dialog Box**



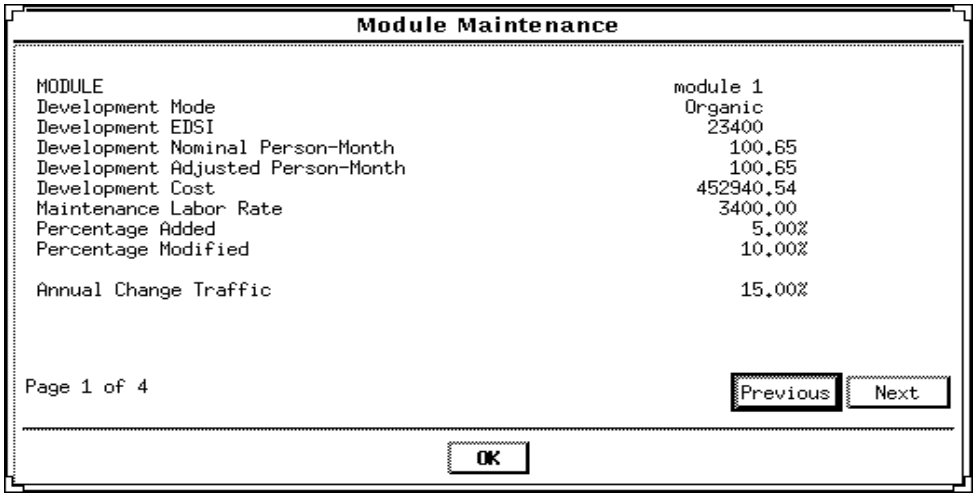
Upon completing the adjustment of the cost drivers click the OK button or click the Cancel button to return to the CLEF without viewing maintenance estimations.

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).

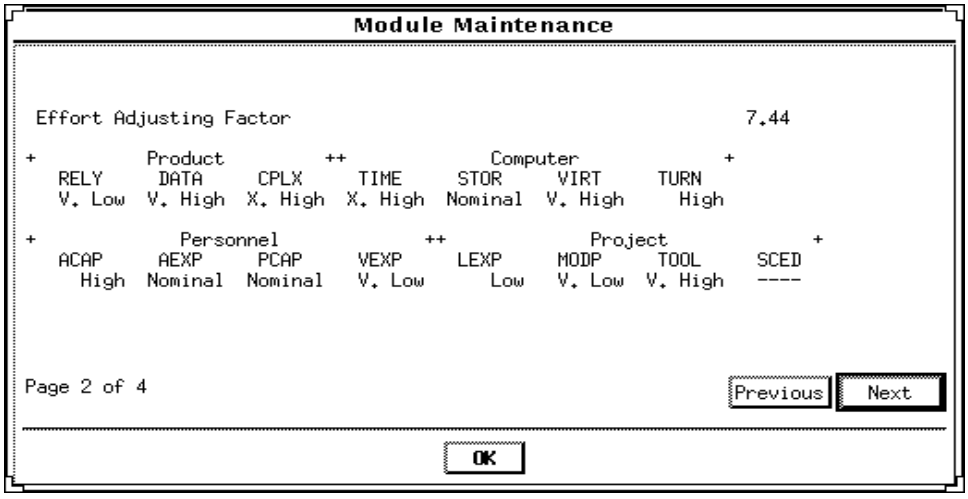
When the OK button is clicked in the Module Maintenance Dialog Box, a window displaying the first of four pages that contains the module name, the current development mode, the total number of source instructions for development of the module (EDSI) hat is loaded in the CLEF, the nominal effort of the 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 6-14) and the calculated annual change traffic.

FIGURE 6-14 **Module Maintenance window (page 1)**



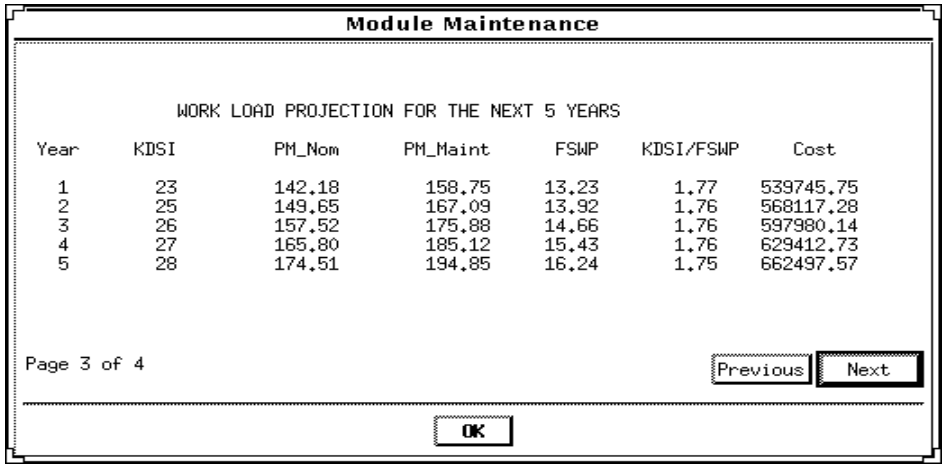
The second page of the maintenance window can be seen by clicking upon the Next button. It contains the settings for the 14 cost drivers, SCED is not applicable (see figure 6-15).

FIGURE 6-15 **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 KDSI (EDSI * 10^3), 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 module for the year (FSWP), the number of instructions that are to be maintained be per personnel(KDSI/FSWP) and the total cost for maintenance for the year.

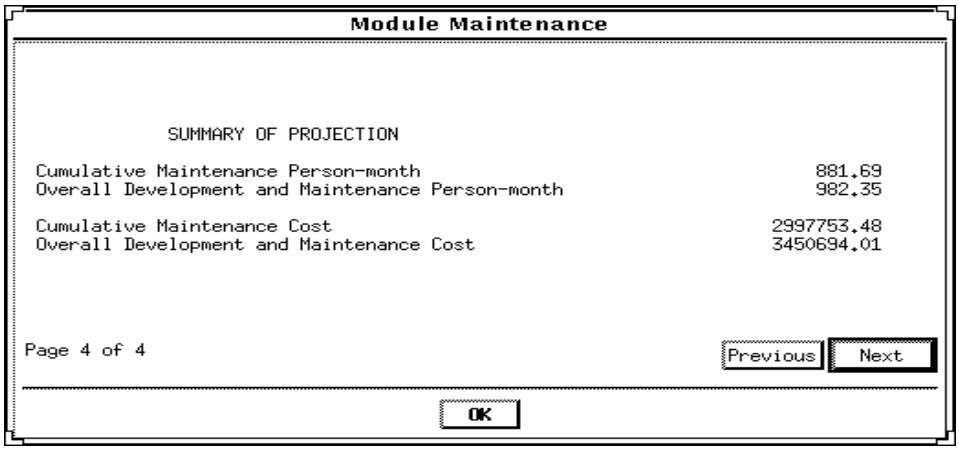
FIGURE 6-16 **Module Maintenance window (page 3)**



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 6-17). This first

displays the total number 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

FIGURE 6-17 **Module Maintenance window (page 4)**



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.