

LIST OF FIGURES:	30
LIST OF TABLES:	31
2.2.3. DATA COLLECTION:	32
2.2.3.1. COCOMOII definition	32
2.2.3.2. COCOMOII Models	32
2.2.3.3. Development Effort Estimates	33
2.2.3.4. Scaling Drivers	33
2.2.3.5. Post-Architecture Model Cost drivers	34
2.2.3.6. Early Design Model Cost drivers	35
2.2.3.7. Object Oriented Project sizing:	36
2.2.3.8. Object Oriented Function Point Counting Procedure:	37
2.2.3.9. Example:	39
2.2.3.10. Line Of Code (LOC) estimation:	41
2.3. REQUIREMENTS SPECIFICATION:	42
2.3.1. REQUIREMENT DOCUMENT	43

List of figures:

Figure 1: Perspectives and measures in the software development process 36
Figure 2: Example to count the OOFP 39

List of tables:

Table 1: Scale Factors for COCOMO II Early Design and Post-Architecture Models 33
Table 2: Scale Factors Explanation..... 34
Table 3: Post-Architecture cost drivers rating values 35
Table 4: Early Design cost drivers..... 35
Table 5: IFPUG Complexity matrix for ILFs and ELF..... 37
Table 6: IFPUG Unadjusted Function Point Table..... 38
Table 7: TFPCP’s Complexity matrix for external input (EIs)..... 38

2.2.3.

Data Collection:

2.2.3.1.COCOMOII definition

COCOMO II is an effort to update the well-known COCOMO (Constructive Cost Model) software cost estimation model originally published in Software Engineering Economics by Dr. Barry Boehm in 1981. It focuses on issues such as non-sequential and rapid-development process models; reuse-driven approaches involving commercial-off the-shelf (COTS) packages, reengineering, applications composition, and application generation capabilities; object-oriented approaches supported by distributed middleware; software process maturity effects and process-driven quality estimation.[5]

2.2.3.2.COCOMOII Models

COCOMO II provides the following three-stage series of models for estimation of Application Generator, System Integration, and Infrastructure software projects:

1. The earliest phases or spiral cycles will generally involve prototyping, using the Application Composition model capabilities. **The COCOMO II Application Composition model** supports these phases, and any other prototyping activities occurring later in the life cycle.
2. The next phases or spiral cycles will generally involve exploration of architectural alternatives or incremental development strategies. To support these activities, COCOMO II provides an early estimation model called **the Early Design model**. This level of detail in this model is consistent with the general level of information available and the general level of estimation accuracy needed at this stage.
3. Once the project is ready to develop and sustain a fielded system, it should have a life- cycle architecture, which provides more accurate information on cost driver inputs, and enables more accurate cost estimates. To support this stage, COCOMO II provides **the Post-Architecture model**. [34]

2.2.3.3. Development Effort Estimates

In COCOMO II effort is expressed as Person Months (PM). person month is the amount of time one person spends working on the software development project for one month.

Equation I-1 is the base model for the Early Design and Post-Architecture cost estimation models. The inputs are the Size of software development, a constant, A, and a scale factor, B. The size is in units of thousands of source lines of code (KSLOC). This is derived from estimating the size of software modules that will constitute the application program. It can also be estimated from unadjusted function points (UFP), converted to SLOC then divided by one thousand". M is the cost drivers of the project. There are 7 cost drivers in the Early Design model and 17 cost drivers in the Post-Architecture model.[2][5]

$$PM = A * (Size)^B * EM \quad \text{Eq.1.1}$$

2.2.3.4. Scaling Drivers

Equation I-2 defines the exponent, B, used in Equation I-1. Table 1 provides the rating levels for the COCOMO II 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, Wi, are summed across all of the factors, and used to determine a scale exponent, B, via the following formula:

$$B = 0.91 + 0.01 * \sum Exp_i \quad I = 1 \dots 5 \quad \text{Eq.1.2} \quad [5]$$

Wi	Very Low	Low	Nominal	High	Very High	Extra High
1-Precedentedness	4.05	3.24	2.42	1.62	0.81	0.00
2-Development/Flexibility	6.07	4.86	3.64	2.43	1.21	0.00
3-Architecture/Risk Resolution	4.22	3.38	2.53	1.69	0.84	0.00
4- Team Cohesion	4.94	3.95	2.97	1.98	0.99	0.00
5- Process Maturity	4.54	3.64	2.73	1.82	0.91	0.00

Table 1: Scale Factors for COCOMO II Early Design and Post-Architecture Models

Scale factor	Explanation
1-Precedentedness	Reflects the previous experience of the organization with this type of project. Very Low means no previous experience; Extra high means that the organization is completely familiar with this application domain.
2-Development/Flexibility	Reflects the degree of flexibility in the development process. Very low means a prescribed process is used; Extra high means that the client sets only general goals.
3-Architecture/Risk Resolution	Reflect the extent of risk analysis carried out. Very low means little analysis; Extra high means a complete and thorough risk analysis.
4- Team Cohesion	Reflects how well the development team know each other and work together. Very low means very difficult interaction; Extra high means an integrated and effective team with no communication problems.
5- Process Maturity	Reflects the process maturity of the organization.

Table 2: Scale Factors Explanation

2.2.3.5. Post-Architecture Model Cost drivers

The Post-Architecture model is the most detailed estimation model and it is intended to be used when a software lifecycle architecture has been developed. The Post-Architecture model includes a set of 17 cost drivers and a set of 5 factors determining the projects scaling component.

The 17 effort multipliers used in COCOMO II Post-Architecture model are grouped into four categories: product, platform, personnel, and project. [5]

Cost Driver	Description	Rating					
		Very Low	Low	Nominal	High	Very High	Extra High
<i>Product</i>							
RELY	Required software reliability	0.75	0.88	1.00	1.15	1.39	-
DATA	Database size	-	0.93	1.00	1.09	1.19	-
CPLX	Product complexity	0.70	0.88	1.00	1.15	1.30	1.66
RUSE	Required reusability		0.91	1.00	1.14	1.29	1.49
DOCU	Documentation		0.95	1.00	1.06	1.13	
<i>Platform</i>							
TIME	Execution time constraint	-	-	1.00	1.11	1.31	1.67

STOR	Main storage constraint	-	-	1.00	1.06	1.21	1.57
PVOL	Platform volatility	-	0.87	1.00	1.15	1.30	-
<i>Personnel</i>							
ACAP	Analyst capability	1.50	1.22	1.00	0.83	0.67	-
PCAP	Programmer capability	1.37	1.16	1.00	0.87	0.74	-
PCON	Personnel continuity	1.24	1.10	1.00	0.92	0.84	-
AEXP	Analyst experience	1.22	1.10	1.00	0.89	0.81	-
PEXP	Programmer experience	1.25	1.12	1.00	0.88	0.81	-
LTEX	Language and tool experience	1.22	1.10	1.00	0.91	0.84	
<i>Project</i>							
TOOL	Software Tools	1.24	1.12	1.00	0.86	0.72	-
SITE	Multisite development	1.25	1.10	1.00	0.92	0.84	0.78
SCED	Development Schedule	1.29	1.10	1.00	1.00	1.00	-

Table 3: Post-Architecture cost drivers rating values

For more details about Post-Architecture cost drivers refer to appendix A

2.2.3.6. Early Design Model Cost drivers

This Early Design model is used in the early stages of a software project when very little may be known about the size of the product to be developed, the nature of the target platform, the nature of the personnel to be involved in the project, or the detailed specifics of the process to be used.

The 7 effort multipliers used in Early Design Model are obtained by combining the Post-Architecture cost drivers shown in Table 4. [5]

Cost Driver	Description	Counterpart Combined Post-Architecture Cost Driver
RCPX	Product reliability and complexity	RELY, DATA, CPLX, DOCU
RUSE	Required reuse	RUSE
PDIF	Platform difficulty	TIME, STOR, PVOL
PERS	Personnel capability	ACAP, PCAP, PCON
PREX	Personnel experience	AEXP, PEXP, LTEX
FCIL	Facilities	TOOL, SITE
SCED	Schedule	SCED

Table 4: Early Design cost drivers

For more details about Early Design cost drivers rating values refer to appendix A

2.2.3.7.Object Oriented Project sizing:

Cost estimation is not one-time activity at project initiation. Estimates should be refined continually throughout a project. Thus, it is necessary to estimate size repeatedly throughout development. Most research on estimating size has dealt with traditional applications and traditional software development practices. Few methods have been proposed for object oriented software development. In this project we need an adaptation of the classical function point method to object oriented software.

Figure 5 show the main phases of an object oriented (OO) development process, and the measurements that can be obtained at different points in development. As we move through the phases of the process, the measurement perspective changes from that of the user to the designer. [10]

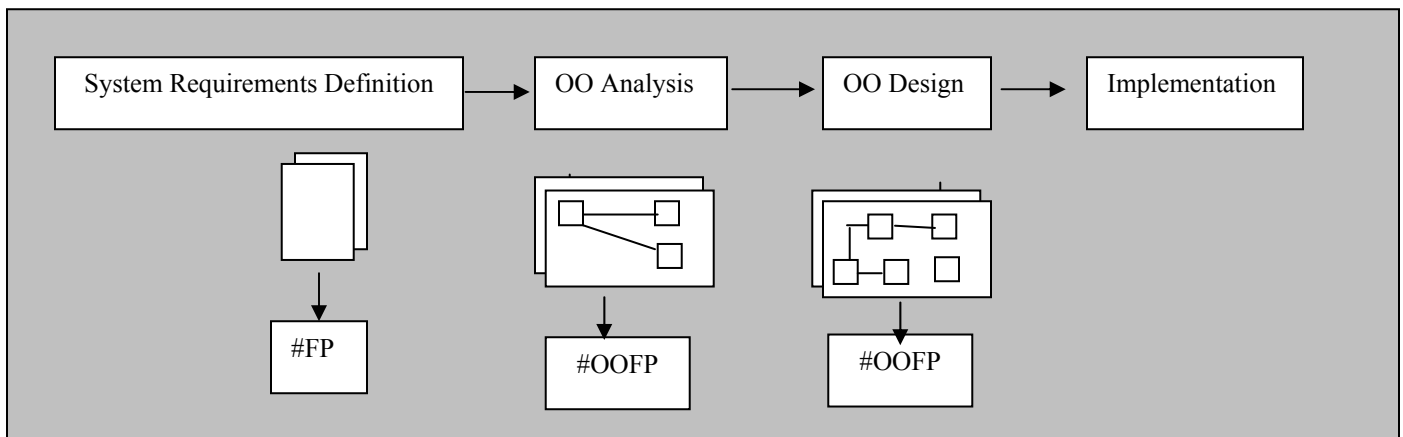


Figure 7: Perspectives and measures in the software development process

At the end of the requirement specification phase, the classical function points (FP) counting method is applied on the requirements document. The function point method takes the perspective of the end user. What are actually measured are the functions of the system that are visible to the end user. This measure is generally considered to be independent of any particular implementation.

Some aspects of (OO) system (e.g. inheritance, aggregation, association and polymorphism) are not included in the classical function point count. Nevertheless, they contribute to the final size of the system. If the objective of measuring functionality is to estimate the final size of an implementation of a system, and from that the effort and

duration of software project, these aspects should be taken into account. This changes the perspective from that of the user to that of the customer (the organization that develops and pays the development cost). More detailed information is available, as the system evolves from abstract specification to a concrete implementation. It should be possible to refine a size estimate repeatedly, as long as this project tend to develop a tool to automate the counting of OOFPs, recalculation is easy at any time. [10]

2.2.3.8.Object Oriented Function Point Counting Procedure:

In object oriented software design, all data elements those are visible to all methods of a class are called logical file. The methods in a class are called Transactional functions. At the design phase, Data in a class can be mapped to a logical file and each method to a transactional function.

Logical File

There are two types of logical files depending on the application boundary. Internal Logical File (ILF) is a Logical file within the application boundary and External Interface File (EIF) is a Logical file outside the application boundary.

The complexity of an ILF/EIF depends on the DETs and RETs it has. A DET is a simple data type such as int, char, etc. Object reference, a complex data type is considered as a RET. So, a RET provides the complexity due to aggregation.

The inherited data is visible to all the methods in a derived class. So, inherited data should be included to calculate the complexity of a derived class. After DETs and RETs are counted, IFPUG tables (Table 5 - 6) can be used to classify the logical files as low, average or high. [24]

		DET		
		1-19	20-25	>=51
RET	1	Low	Low	Average
	2-5	Low	Average	High
	>5	Average	High	High

Table 5: IFPUG Complexity matrix for ILFs and ELFs

Component	Function Levels		
	Low	Average	High
ILF	x 7	x 10	x 15
EIF	x 5	x 7	x 10

Table 6: IFPUG Unadjusted Function Point Table

Transactional Function

Object models rarely contain the information needed to tell whether a method performs an input, an output or is dealing with an enquiry [10]. So, no distinction is made among the methods. A method in a class operates on the data within that class, arguments and return values. Since, the data within a class is already considered as a logical file, it can be omitted while calculating the complexity of a method. So, the complexity of a method depends on the DETs and RETs.

The inherited methods will be coded only once in the base class. So, methods that are inherited from a base class should not be considered for estimating the complexity of a derived class. If any derived class overrides a method, its complexity should be considered for that derived class alone.

An abstract method will be declared in a base class and defined in its derived classes. The complexity of an abstract method should be considered only for derived classes.

Using the signature of a method, it is possible to identify the communicating objects. So, association should be considered for the method from where it invokes the required method(s). A single valued association is considered as a DET and a multi-valued association is considered as a RET.

If a method does not have any arguments and return type, then its complexity is considered as one DET.[24]

The method complexity is rated according to TFPCP's external input rate complexity table.

		DET		
		1-4	5-15	>=16
RET	<2	Low	Low	Average
	2	Low	Average	High
	>2	Average	High	High

Table 7: TFPCP's Complexity matrix for external input (EIs)

Component	Function Levels		
	Low	Average	High
EI	x 3	x 4	x 6

Table 8: IFPUG Unadjusted Function Point Table

2.2.3.9.Example:

The following example mentioned in [24]. It represents most of object-oriented properties. The design of the example drawn according to Object Modeling Technique (OMT).

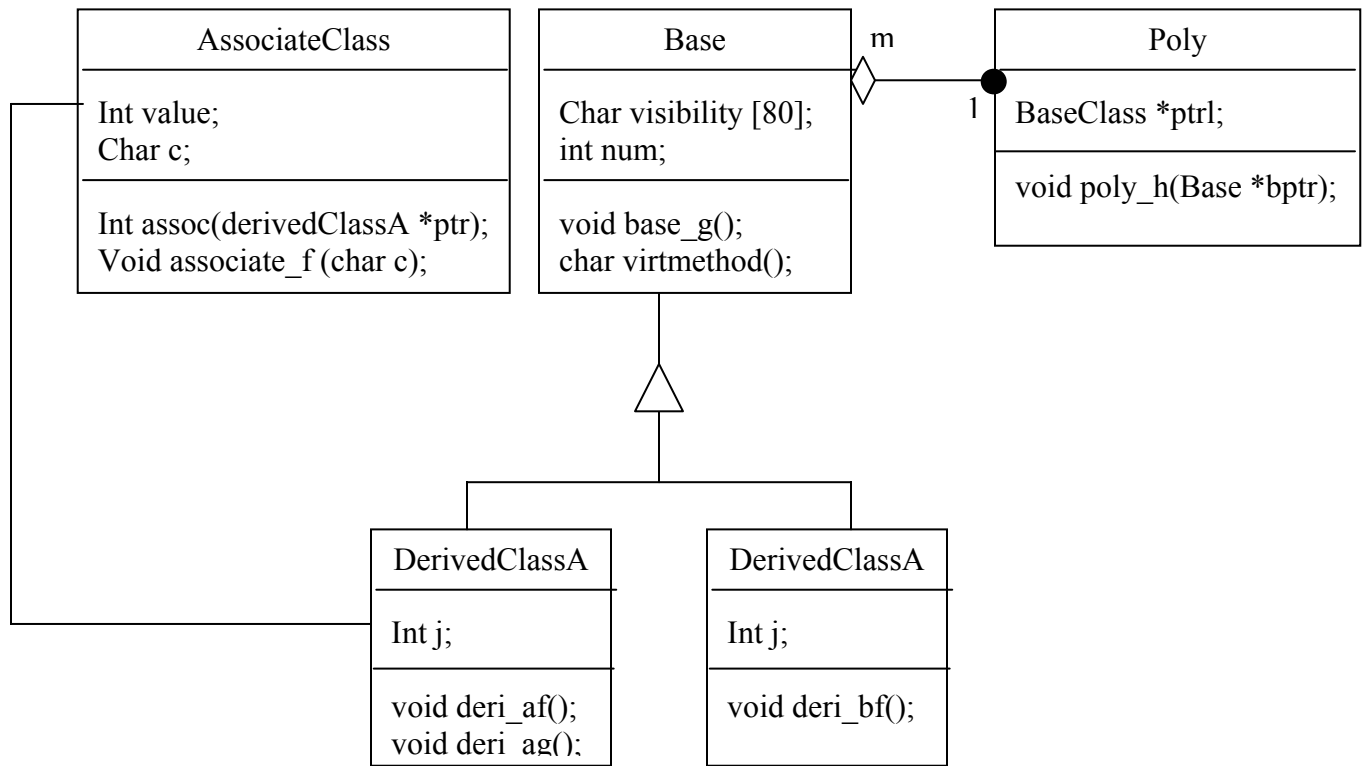


Figure 8: Example to count the OOFP

In the following table count the number of DETs and RETs of each logical file and method and the reason for the count.

Logical File /Transactional Functions	DETs	Reason	RETs	Reason
---------------------------------------	------	--------	------	--------

Base	2	2 simple data elements	1	Because of the class itself
Poly	0	No data elements	2	Because of the Base class reference and the class itself
AssociateClass	2	2 simple data elements	1	Because of the class itself
DerivedClassA	3	Two due to the inherited data from Base class, and one due to the data element it has.	1	Because of the class itself
DerivedClassB	3	Two due to the inherited data from Base class, and one due to the data element it has.	1	Because of the class itself
Base_g	1	The method dose not has any arguments and dose not return type.	0	The method dose not has reference to any class.
virtmethod	2	The method dose not has any arguments and it returns type.	0	The method dose not has reference to any class.
Poly_h	0	The method has argument and dose not return type.	2	One due to the Base class reference and other due to the multi-valued association with Base.
assoc	1	Due to the single valued association with DerivedClassA	1	Because of the DerivedClassA reference.
Associate_f	1	Due to the single valued association with DerivedClassA	0	Because it dose not has any reference.
Deri_af	1	Due to the single valued association with AssociateClass	0	Because it dose not has any reference.
Deri_ag	1	Due to the single valued association with AssociateClass	0	Because it dose not has any reference.
Deri_bf	1	The method dose not has any arguments and dose	0	The method dose not has reference to any class.

		not return type.		
--	--	-------------------------	--	--

Based on the Table 5 and Table 6, all logical files (classes) are Low complexity, so the total number of its UOOFP = $5 * 7 = 35$.

Based on the Table 7 and Table 8, all Transactional Functions (methods) are Low complexity, so the total number of its UOOFP = $8 * 3 = 24$.

The total number of UOOFP = $35+24=59$.

Adjusted OOFP can be estimated by using the Value Adjusted Factor (VAF). [4]

How we can compute (VAF).

The following 14 general system characteristics (GSCs) are summarized into the Value Adjusted Factor (VAF).

1. Backup and recovery
2. Data communication
3. Distributed processing function
4. Is performance critical?
5. Existing operating environment
6. On- line data entry
7. Input transaction built over multiple screens
8. Master files updated on-line
9. Complexity of inputs, outputs, files, inquiries
10. Complexity of processing
11. Code design for reuse
12. Are conversion/installation included in design?
13. Multiple installations
14. Application designed to facilitate change by the user.

The following steps provide the procedure to calculate the VAF:

Evaluate the 14 GSCs on a scale from 0(no influence) to 5 (strong influence) to determine the degree of influence (DI) for each of the GSC descriptions.

Add the DI for all 14 GSCs to produce the total degree of influence (TDI).

Use the application TDI in the following equation to compute the VAF:

$$\text{VAF} = (\text{TDI} * 0.01) + 0.65.$$

Then the OOFP could be calculated from the following equation:

$$\text{OOFP} = \text{VAF} * \text{UOOFP}$$

2.2.3.10. Line Of Code (LOC) estimation:

Traditionally size meant the length of the delivered program, in lines of code (LOC). There is a large body of experience now in estimating effort from LOC, using

models such as COCOMO, or knowledge of team / company dependent productivity factors.

Unfortunately, estimating LOC has proved to be just as difficult as estimating effort, especially early in development when the estimates are of most use.

The ideal is to identify size measure that can be used to estimate effort directly. But it takes time, and extensive data, to develop and validate direct estimation approaches. In the mean time, an alternative is to estimate LOC as an intermediate step (i.e. use other measures as a basis for estimating length in LOC), and then use existing LOC-based models to estimate effort. This provides immediate access to the body of LOC-based experience, which is still very widely used.

This project uses OOFPs to estimate LOC. Direct estimation of effort requires data that is not available, so look at estimating LOC as a step towards estimating effort and cost. Ultimately, we hope to investigate direct relationships between OOFPs and project effort and cost.

Several regression models were developed, to relate LOC to software metrics computed at the design stage (OOFPs). Models with small number of independent variables were preferred, for ease of use and interpretation, and because the number of data points is not large. Caldiera et al [10], compared robust regression techniques with the more traditional least squares line fitting . They found that the robust models do not gain much in explaining the data better. They found “ The best predictive accuracy (NMSE=0.337) was achieved by the rreg-logistic-G model with tuning parameter $u = :8$, corresponding to the linear predictor $LOC = 7183:4 + 25:6$ OOFP: (This model is very close to the basic linear model lm-G, whose equation is $LOC = 7435:1 + 25:2$ OOFP:)” [10]

Thus, only simple linear regression model will be investigated here.

2.3. Requirements Specification:

The software requirement document is the deliverable of software requirements specification. It should include both the user requirements and system requirements. This section will use IEEE template in the requirement document.

2.3.1. Requirement Document

1- Introduction

The requirements document should contain all the requirements developed for the project, as well as other information. The requirements document is one of the major deliverables during the first quarter of a project, and should be reviewed thoroughly in each subsequent quarter.

1.1 Purpose

The Software cost Estimation tool under construction tend to be very useful for any software house when they need to estimate the cost of their Object Oriented software projects written in C++ language as consol application. This tool enables the software house to feed the system with their historical object oriented projects data (at least 10 projects) then ask the system to calibrate the most famous and practical cost estimation model COCOMOII. The calibrated model can be used to estimate the cost for any new object oriented project at design phase by its class's properties (inheritance, aggregation, association and polymorphism). Also this tool enable the software house user to re-calibrate the model and re-estimate the project if there is any new data.

1.2 Scope

Application Proposed Name

English Long Name:50	<i>Object Oriented Software Cost Estimation Tool</i>	
English Short Name:4	<i>OOSCET</i>	
Arabic Name :50 Char		

Application description:... This application is an object-oriented software cost estimation tool that used by any software house. It constructs new cost model by calibrating COCOMOII Early Design Model according to the historical object oriented projects data of a specific software house. The calibrated model can be used to estimate the cost of a new object oriented project. Recalibration and re-estimation can be done at any time.

Application Goals :

The goal of this application is to present easy to use and powerful cost estimation tool that can be used by any software house.

Application Objectives:

1. To develop a software cost estimation model suitable for any organization by calibrating COCOMOII model.
2. To develop software cost database and tool support capabilities for continuous model improvement by the ability of recalibrate the model when more historical data are available.

1.3 References

The references listed at reference part.

2- Overall Description

In this section, describe the general factors that affect the product and its requirements. This section should contain background information, not state specific requirements.

Check [✓] an appropriate requirement.

2.1 Product Perspective

2.1.1 User Interfaces

2.1.1.1 Screen Description

Requirement No	R1
Screen No	S1
Screen name	Main screen
Input	Function button
Output	Function Screen
Description	This screen present main function of the system

Requirement No	R2
Screen No	S2

Screen name	Software House Edit/Entry Form
Input	Software house name, Telephone, Fax, Actual person per month (PM), Labor rate, process maturity.
Output	Save Software House information record in database.
Description	This screen enables the user to enter or update Software House information then save the record.

Requirement No	R3
Screen No	S3
Screen name	Required information screen
Input	Enter Required information (Basic information, 17 Effort Multipliers, exponent drivers, 14 General System Characteristics, Class's information, Method's information) For old or new project by using wizard screen. Help, Next and Back buttons.
Output	Update old project or new project database Go to Detailed Question & answers screen or Next or Back Required information
Description	This screen displays the required information for old or new project as a wizard screen.

Requirement No	R4
Screen No	S4
Screen name	Detailed Question & answers screen
Input	Enter Required information by select suitable answer from the answers list for old or new project Ok or Cancel buttons to confirm the answer.
Output	Update old project or new project database Display your answer in Required information screen
Description	This screen displays the required information in question form and its description. Also presents all possible answers and its description.

Requirement No	R5
Screen No	S5
Screen name	Custom Old Software Project screen
Input	Enter Old project information using custom form (Basic information, 17 Effort Multipliers, exponent drivers, 14 General System Characteristics, Class's information, Method's information) Function buttons to save/ Edit/Delete/Cancel/Close
Output	Update old project database
Description	This screen enables the user to enter /edit/ Delete Old project information

Requirement No	R6
Screen No	S6
Screen name	Custom Estimate New Software Project screen
Input	Enter New project information using custom form (Basic information, 7 Effort Multipliers, exponent drivers, 14 General System Characteristics, Class's information, Method's information) FUNCTION BUTTONS TO SAVE/ EDIT/DELETE/CANCEL/CLOSE ESTIMATE BUTTON TO ESTIMATE CURRENT PROJECT
Output	Update new project database (Add, update, delete) The result of the estimation.
Description	This screen enables the user to enter /edit/ Delete new project information to estimate its effort and cost.

2.1.1.2 Menu Structure

Requirement No: R7

- Menu driven*
- Button control* *[description in section3]*
- Command Line*
- Other*

2.1.1.3 Interface Colors

Requirement No	Item name	color
R8	<i>Background screen</i>	Picture
R9	<i>Function button</i>	Gray

2.1.1.4 User Interface Language: **Requirement No: R10**

- Arabic* *English* *Arabic/English*

2.1.1.4 Application Help: **Requirement No: R11**

- Online Help Screens* *User Manual* *Both*

2.1.2 Hardware Interfaces:

2.1.2.1 Hardware environment **Requirement No: R12**

- Stand alone PC* *Main Frame*
- Network* *Internet* *Other:*

2.1.2.2 Which devices are to be supported **Requirement No: R13**

- Modems* *Scanner*
- Code reader* *Light pen*
- touch Screen* *Plotters*
- printers* *Other: Non*

2.1.3 Software Interfaces:

2.1.3.1. Software requirements **Requirement No: R14**

Refer to
Project planning phase
Project resources – Software.

2.1.4 Memory Constraints: **Requirement No: R15**

I will not accept the system if the Memory size needed exceeds:

64K 1MB 5 MB Other:

I will not accept the system if it needs HDD size exceeds:

640K 1MB 5MB 10MB 1GB More

2.2 Product Functions:

System Functions

Requirement No	Function No	Function name	Description
R16	F1	Save	To save current record in database
R17	F2	Cancel	To cancel changes in current record
R18	F3	Close	To close current screen
R19	F4	Edit	To edit current record
R20	F5	Delete	To delete current record
R21	F6	New	To enter new record
R22	F7	Next	To go to next step in wizard screen
R23	F8	Back	To go to previous step in wizard screen
R24	F9	Help	To give help information
R25	F10	Project	To go to project information in the wizard

Requirement No	Function No	Function name	Description
R26	F11	Classes	To go to class's information in the wizard
R27	F12	Methods	To go to Methods information in the wizard
R28	F13	Another	To Enter another project, classes or methods information in the wizard
R29	F14	Estimate	To display the estimation result for current project.
R30	F15	OK	To confirm the user choice for required information.
R31	F16	Software house information	Open Software house information screen.
R32	F17	Old software project Wizard	Open Old software project Wizard screen.
R32	F18	Old software project Custom	Open Old software project Custom screen.
R33	F19	Calibrate COCOMO early design model	Execute Calibration procedure
R34	F20	Estimate new software project - Wizard	Open Estimate new software project –Wizard screen
R35	F21	Estimate new software project - Custom	Open Estimate new software project –Custom screen

2.3 User Characteristics:

<i>User group</i>	<i>Educational level</i>	<i>Experience</i>	<i>Technical expertise</i>
<i>Manager</i>	<i>Project management specialist</i>		
<i>Analysts</i>	<i>Software engineering specialist</i>		
<i>Designers</i>	<i>Object Oriented Projects specialist</i>		
<i>Public user</i>	<i>Data entry course</i>		

2.4 Constraints:

2.4.1 reliability requirements **Requirement No: R36**

I will not accept the system if the response time is more than:

2 Seconds 5 Seconds 10 Seconds

2.4.2 Criticality of the application **Requirement No: R37**

high medium low

2.4.3 Security: **Requirement No: R38**

None Single Password Multilevel Non

2.4.4 Higher-order language requirements **Requirement No: R39**

English Arabic English/Arabic

2.4.5 Programming language used **Requirement No: R40**

VBasic Delphi C++ Power Builder
 VBScript Other

3 Specific Requirements

This section should describe all software requirements at a sufficient level of detail for designers to design a system satisfying the requirements and testers to verify that the system satisfies requirements.

3.1 External Interface Requirements

3.1.1 User Interfaces

3.1.1.1 *Menu Structure*

Button control

Requirement No	Button control number	Button control name	Related function#	Related screen#
R41	B1	Software house information	F16	S2
R42	B2	Old software project (Wizard)	F17	S3
R43	B3	Old software	F18	S5

Requirement No	Button control number	Button control name	Related function#	Related screen#
		project (Custom)		
R44	B4	Calibrate COCOMOII Early Design Model	F19	S1
R45	B5	Estimate new software project (Wizard)	F20	S3
R46	B6	Estimate new software project (Custom)	F21	S6
R47	B7	Save	F1	S2, S5, S6
R48	B8	Close	F3	S2, S5, S6
R49	B9	Cancel	F2	S2, S4, S5, S6
R50	B10	New	F6	S5, S6
R51	B11	Edit	F4	S5, S6
R52	B12	Delete	F5	S5, S6
R53	B13	Next	F7	S3
R54	B14	Back	F8	S3
R55	B15	OK	F15	S4
R56	B16	Estimate	F14	S6
R57	B17	Project	F10	S3
R58	B18	Classes	F11	S3
R59	B19	Methods	F12	S3
R60	B20	Another	F13	S3
R61	B21	Help	F9	S3

3.1.1.2 Screens

Requirement No		R1									
Screen No		S1									
Screen name		Main screen									
Purpose		This screen present main function of the system									
Input Items	Source of input	Valid range	Input mask	Format	Related input/output items		Output Items	Source of output	Format	Related input/output items	
B1,B2, B3,B4, B5, B6	S1	-	-	-	-		S3, S5, S1, S6	-	-	-	
<i>Screen format</i>											
<table border="1"> <tr> <td> Main options B1 B2 B3 B4 B5 </td> </tr> </table>											Main options B1 B2 B3 B4 B5
Main options B1 B2 B3 B4 B5											

Requirement No		R2								
Screen No		S2								
Screen name		Software House Edit/Entry Form								
Purpose		This screen enables the user to enter /edit/ Delete Old project information								
Input Items	Source of input	Valid range	Input mask	Format	Related input/output items		Output Items	Source of output	Format	Related input/output items
Software house name, Telephone, Fax, Actual (PM), Labor rate, process maturity.	S2	-	-	-	-		-	-	-	-
<i>Screen format</i>										

<p><u>Software House Edit/Entry Form</u> Software house name----- Telephone----- Fax----- Actual (PM)----- Labor rate----- process maturity-----</p>
--

Requirement No			R3								
Screen No			S3								
Screen name			Required information screen								
Purpose			This screen displays the required information for old or new project as a wizard screen.								
Input Items	Source of input	Valid range	Input mask	Format	Related input/output items		Output Items	Source of output	Format	Related input/output items	
Required information. Back Next Cancel Help	-	-	-	-	-		Previous required info. Next info Cancel info S4	-	-	-	
<i>Screen format</i>											
<table border="1"> <tr> <td style="text-align: center;"> <p>Required information <input type="text"/> Help Description Back Next Cancel</p> </td> </tr> </table>											<p>Required information <input type="text"/> Help Description Back Next Cancel</p>
<p>Required information <input type="text"/> Help Description Back Next Cancel</p>											

Screen No			S4							
Screen name			Detailed Question & answers screen							
Purpose			This screen displays the required information in question form and its description. Also presents all possible answers and its description.							
Input Items	Source of input	Valid range	Input mask	Format	Related input/output items		Output Items	Source of output	Format	Related input/output items

Answer selection	-	-	-	-	-		-	-	-
Help						Help for Q or A			
OK						Confirm Answer			
Cancel						Cancel answer			

Screen format

Required information

Question? Help

Answer List Help

OK Cancel

Requirement No			R5						
Screen No			S5						
Screen name			Custom Old Software Project screen						
Purpose			This screen enables the user to enter /edit/ Delete Old project information						
Input Items	Source of input	Valid range	Input mask	Format	Related input/output items	Output Items	Source of output	Format	Related input/output items
Old project information B7 B8 B9 B10 B11 B12		-	-	-	-	Save record in DB Close screen Cancel changes New record Edit record Delete record from DB	-	-	-
<i>Screen format</i>									

<p><u>Old project information</u> Basic info GSCs info EMs Clases info Exps Methods info</p> <p>New Save Edit Delete Cancel Close</p>

Requirement No		R6							
Screen No		S6							
Screen name		Custom Estimate New Software Project screen							
Purpose		This screen enables the user to enter /edit/ Delete new project information to estimate its effort and cost.							
Input Items	Source of input	Valid range	Input mask	Format	Related input/output items	Output Items	Source of output	Format	Related input/output items
New project information B7 B8 B9 B10 B11 B12 B16		-	-	-	-	Save record in DB Close screen Cancel changes New record Edit record Delete record from DB Display the result of estimation	-	-	-

<i>Screen format</i>															
<table border="1"> <tr> <td colspan="2"><u>new project information</u> Basic info GSCs info EMs Clases info Exps Methods info</td> <td style="border: 1px solid black; text-align: center; vertical-align: middle;">Estimation result</td> </tr> <tr> <td colspan="2">New Save Edit Delete Cancel Close</td> <td>Estimate</td> </tr> </table>										<u>new project information</u> Basic info GSCs info EMs Clases info Exps Methods info		Estimation result	New Save Edit Delete Cancel Close		Estimate
<u>new project information</u> Basic info GSCs info EMs Clases info Exps Methods info		Estimation result													
New Save Edit Delete Cancel Close		Estimate													

3.1.1.3 Interface Colors

Requirement No	Item #	Item name	Color
R8	1	Background screen	Picture
R9	2	Function button	Gray

3.1.2 Hardware Interfaces:

Requirement No	Hardware No	Hardware Name	H/W Specification	How S/W communicate with H/W
R13			Non	

3.1.3 Software Interfaces:

3.1.2.1 *Software requirements*

Requirement No	Software name	S/W type	Version	Purpose	Related function	Description
R62	Ms Access	Database	2000	Store data	Insert delete update retrieve	
R63	Microsoft power point	MS office	2000	Diagrams	Drawing diagrams	
R64	Microsoft Word	MS office	2000	Documentation	Project Documentation	
R65	Microsoft Project	Management		Project Planning	Project Planning and management	
R66	Install shield			CD installation	Project SW tool installation	

3.1.2 Software Operating System

Requirement No	Operating System name	version	Special configuration
R67	Microsoft Windows XP	2000	

3.2 Software Product functions

Requirement No	R16
Function No	F1

Function name		Save					
Description		To save current record in database					
Input	Source	Valid range of input	Error handling message	Processing Sequence of operation	Formulas used	Effects of parameters	Output
B7	S2, S5, S6						Save record in DB

Requirement No		R17					
Function No		F2					
Function name		Cancel					
Description		Cancel changes in current record					
Input	Source	Valid range of input	Error handling message	Processing Sequence of operation	Formulas used	Effects of parameters	Output
B9	S2, S4, S5, S6						Cancel current changes on record

Requirement No		R18					
Function No		F3					
Function name		Close					
Description		Close current screen					
Input	Source	Valid range of input	Error handling message	Processing Sequence of operation	Formulas used	Effects of parameters	Output
B8	S2, S5, S6						Close current screen

Requirement No		R19					
Function No		F4					
Function name		Edit					

Description		Edit current record					
Input	Source	Valid range of input	Error handling message	Processing Sequence of operation	Formulas used	Effects of parameters	Output
B11	S5, S6						Edit current Record

Requirement No		R20					
Function No		F5					
Function name		Delete					
Description		Delete current record					
Input	Source	Valid range of input	Error handling message	Processing Sequence of operation	Formulas used	Effects of parameters	Output
B12	S5, S6						Delete current Record

Requirement No		R21					
Function No		F6					
Function name		New					
Description		Enter new record					
Input	Source	Valid range of input	Error handling message	Processing Sequence of operation	Formulas used	Effects of parameters	Output
B10	S5, S6						Enter new Record

Requirement No		R22					
Function No		F7					
Function name		Next					
Description		Go to next step in wizard screen					

Input	Source	Valid range of input	Error handling message	Processing Sequence of operation	Formulas used	Effects of parameters	Output
B13	S3						Display next required info

Requirement No		R23					
Function No		F8					
Function name		Back					
Description		Go to previous step in wizard screen					
Input	Source	Valid range of input	Error handling message	Processing Sequence of operation	Formulas used	Effects of parameters	Output
B14	S3						Display previous required info

Requirement No		R24					
Function No		F9					
Function name		Help					
Description		Give help information					
Input	Source	Valid range of input	Error handling message	Processing Sequence of operation	Formulas used	Effects of parameters	Output
B21	S3						S4

Requirement No		R25					
Function No		F10					
Function name		Project					
Description		Go to project information in the wizard					
Input	Source	Valid range of input	Error handling message	Processing Sequence of operation	Formulas used	Effects of parameters	Output

B17	S3						S3
-----	----	--	--	--	--	--	----

Requirement No		R26					
Function No		F11					
Function name		Classes					
Description		Go to Class's information in the wizard					
Input	Source	Valid range of input	Error handling message	Processing Sequence of operation	Formulas used	Effects of parameters	Output
B18	S3						S3

Requirement No		R27					
Function No		F12					
Function name		Methods					
Description		Go to Method's information in the wizard					
Input	Source	Valid range of input	Error handling message	Processing Sequence of operation	Formulas used	Effects of parameters	Output
B19	S3						S3

Requirement No		R28					
Function No		F13					
Function name		Another					
Description		Enter another project, classes or methods information in the wizard					
Input	Source	Valid range of input	Error handling message	Processing Sequence of operation	Formulas used	Effects of parameters	Output
B20	S3						S3

Requirement No		R29					
Function No		F14					
Function name		Estimate					

Description		Display the estimation result for current project.					
Input	Source	Valid range of input	Error handling message	Processing Sequence of operation	Formulas used	Effects of parameters	Output
B16	S6						S6

Requirement No		R30					
Function No		F15					
Function name		OK					
Description		Confirm the user choice for required information.					
Input	Source	Valid range of input	Error handling message	Processing Sequence of operation	Formulas used	Effects of parameters	Output
B15	S4						S3

Requirement No		R31					
Function No		F16					
Function name		Software house information					
Description		Open Software house information screen					
Input	Source	Valid range of input	Error handling message	Processing Sequence of operation	Formulas used	Effects of parameters	Output
B1	S1						S2

Requirement No		R32					
Function No		F17					
Function name		Old software project Wizard					
Description		Open Old software project Wizard screen.					
Input	Source	Valid range of input	Error handling message	Processing Sequence of operation	Formulas used	Effects of parameters	Output

B2	S1						S3
----	----	--	--	--	--	--	----

Requirement No		R33					
Function No		F18					
Function name		Old software project Custom					
Description		Open Old software project Custom screen.					
Input	Source	Valid range of input	Error handling message	Processing Sequence of operation	Formulas used	Effects of parameters	Output
B3	S1						S5

Requirement No		R34					
Function No		F19					
Function name		Calibrate COCOMOII early design Model					
Description		To execute Calibration procedure.					
Input	Source	Valid range of input	Error handling message	Processing Sequence of operation	Formulas used	Effects of parameters	Output
B4	S1						S1

Requirement No		R35					
Function No		F20					
Function name		Estimate new software project - Wizard					
Description		Open Estimate new software project – Wizard screen					
Input	Source	Valid range of input	Error handling message	Processing Sequence of operation	Formulas used	Effects of parameters	Output
B5	S1						S3

Requirement No		R36					
Function No		F21					
Function name		Estimate new software project - Custom					

Description		Open Estimate new software project – Custom screen					
Input	Source	Valid range of input	Error handling message	Processing Sequence of operation	Formulas used	Effects of parameters	Output
B6	S1						S6

3.3 Design Constraints

Specify requirements imposed by standards, hardware limitations, etc.

Requirement No	Standard <u>must</u> be followed	Hardware limitation		
		Hard disk size	Memory limit	Other resources
R68	Yes	5GB	500Mb	-

3.4 Software System Attributes

The following items provide a partial list of system attributes that can serve as requirements that should be objectively verified.

Other possible options include scalability, portability, robustness, recoverability, etc.

3.4.1 Reliability:

3.4.1.1 Hardware reliability

Requirement No: R69

- I will not accept the system if the hardware failing In a*

Year month week

Is more than:

one tow three other

- I will not accept the system if the hardware component take to repair more than*

one hour one day one week one month

3.4.1.2 Software reliability

Requirement No: R70

- I will not accept the system if the S/W components produce an incorrect output In a*

Year month week

More than:

one error tow errors ten errors errors

3.4.1.3 Operator reliability

Requirement No: R71

Number of errors made by the operator after training should not exceeds

[] one error/day [] tow errors/day [].....errors/day

5.3 Logical Database Requirements

List of all information that is to be placed into a database:

Req . No	Data entity	Data item	Type	Size	Valid range	Unique Yes/No	null Yes/No	Secure Yes/No
R71	Software house info E1	SW_name	Text	50		Yes	No	No
		Telephone	Number		Long int	Yes	Yes	No
		Fax	Text	50		Yes	Yes	No
		Hours_in_PM	Number		Long int	Yes	Yes	No
		Labor_rate	Currency			Yes	Yes	No
		Exp5	Number		Double	Yes	Yes	No
R72	Old_Projects E2	Project_Number	AutoNumber			Yes	No	No
		Project_Name	Text	50		No	Yes	No
		Actual_PM	Number		Double	No	Yes	No
		Actual_SR_Cost	Currency			No	Yes	No
		Actual_KLOC_Size	Number		Long int	No	Yes	No
		EM1	Number		Double	No	Yes	No
		EM2	Number		Double	No	Yes	No
		EM3	Number		Double	No	Yes	No
		EM4	Number		Double	No	Yes	No
		EM5	Number		Double	No	Yes	No
		EM6	Number		Double	No	Yes	No
		EM7	Number		Double	No	Yes	No
		EM8	Number		Double	No	Yes	No
		EM9	Number		Double	No	Yes	No
		EM10	Number		Double	No	Yes	No
		EM11	Number		Double	No	Yes	No
		EM12	Number		Double	No	Yes	No
EM13	Number		Double	No	Yes	No		
EM14	Number		Double	No	Yes	No		
EM15	Number		Double	No	Yes	No		
EM16	Number		Double	No	Yes	No		
EM17	Number		Double	No	Yes	No		

Req . No	Data entity	Data item	Type	Size	Valid range	Unique Yes/No	null Yes/No	Secure Yes/No
		Exp1	Number		Double	No	Yes	No
		Exp2	Number		Double	No	Yes	No
		Exp3	Number		Double	No	Yes	No
		Exp4	Number		Double	No	Yes	No
		Communications	Number		Integer	No	Yes	No
		DBprocessing	Number		Integer	No	Yes	No
		Performance configuration	Number		Integer	No	Yes	No
		Transaction	Number		Integer	No	Yes	No
		OLdataentry	Number		Integer	No	Yes	No
		efficiency	Number		Integer	No	Yes	No
		OLupdate	Number		Integer	No	Yes	No
		Complexprocessing	Number		Integer	No	Yes	No
		Reusability	Number		Integer	No	Yes	No
		Installation	Number		Integer	No	Yes	No
		Operational	Number		Integer	No	Yes	No
		MultipleSites	Number		Integer	No	Yes	No
		Changes	Number		Integer	No	Yes	No
		R73	Old_class_data E3	Project_number	AutoNumber			Yes
Class_number	Number				Integer	Yes	Yes	No
Simple_DT	Number				Integer	No	Yes	No
Complex_DT	Number				Integer	No	Yes	No
Inhereted_Simple_DT	Number				Integer	No	Yes	No
Inhereted_Complex_DT	Number				Integer	No	Yes	No
R74	Old_class_M ethods E4	Project_number	AutoNumber			Yes	No	No
		Class_number	Number		Integer	Yes	No	No
		Method_number	Number		Integer	Yes	No	No
		Ref_parameter	Yes/No			No	Yes	No
R75	New_Projects E5	Project_Number	AutoNumber			Yes	No	No
		Project_Name	Text	50		No	Yes	No
		EM1	Number		Double	No	Yes	No
		EM2	Number		Double	No	Yes	No
		EM3	Number		Double	No	Yes	No
		EM4	Number		Double	No	Yes	No
		EM5	Number		Double	No	Yes	No
		EM6	Number		Double	No	Yes	No
EM7	Number		Double	No	Yes	No		

Req . No	Data entity	Data item	Type	Size	Valid range	Unique Yes/No	null Yes/No	Secure Yes/No
		Exp1	Number		Double	No	Yes	No
		Exp2	Number		Double	No	Yes	No
		Exp3	Number		Double	No	Yes	No
		Exp4	Number		Double	No	Yes	No
		Communications	Number		Integer	No	Yes	No
		DBprocessing	Number		Integer	No	Yes	No
		Performance configuration	Number		Integer	No	Yes	No
		Transaction	Number		Integer	No	Yes	No
		OLdataentry	Number		Integer	No	Yes	No
		efficiency	Number		Integer	No	Yes	No
		OLupdate	Number		Integer	No	Yes	No
		Complexprocessing	Number		Integer	No	Yes	No
		Reusability	Number		Integer	No	Yes	No
		Installation	Number		Integer	No	Yes	No
		Operational	Number		Integer	No	Yes	No
		MultipleSites	Number		Integer	No	Yes	No
		Changes	Number		Integer	No	Yes	No
R76	New_class_data E6	Project_number	AutoNumber			Yes	No	No
		Class_number	Number		Integer	Yes	Yes	No
		Simple_DT	Number		Integer	No	Yes	No
		Complex_DT	Number		Integer	No	Yes	No
		Inhereted_Simple_DT	Number		Integer	No	Yes	No
		Inhereted_Complex_DT	Number		Integer	No	Yes	No
Association_Type	Text	50		No	Yes	No		
R77	New_class_Methods E7	Project_number	AutoNumber			Yes	No	No
		Class_number	Number		Integer	Yes	No	No
		Method_number	Number		Integer	Yes	No	No
		Ref_parameter	Yes/No			No	Yes	No
R78	AnswersTXT E8	ATXTNumber	Number		Integer	Yes	Yes	No
		TXT	Text	50		No	Yes	No
R79	Answers E9	QNumber	Number		Integer	Yes	Yes	No
		ANumber	Number		Integer	Yes	Yes	No
		ATXTNumber	Number		Integer	No	Yes	No
		AValue	Number		Double	No	Yes	No
		Help	Memo			No	Yes	No
R80	Questions E10	QNumber	Number		Integer	Yes	Yes	No
		TXT	Text	100		No	Yes	No
		Help	Memo			No	Yes	No
R81		TableName	Text	50		Yes	Yes	No

Req . No	Data entity	Data item	Type	Size	Valid range	Unique Yes/No	null Yes/No	Secure Yes/No
	Questions Wizard E11	FeildName	Text	50		Yes	Yes	No
		Order	Number		Integer	No	Yes	No
		QNumber	Number		Integer	No	Yes	No

The entities relationship

Req. No	Entities	E1	E2	E3	E4	E5	E6	E7	E8	E9	E10	E11
	E1											
R82	E2			1-M								
R83	E3		1-M		1-M							
R84	E4			1-M								
R85	E5						1-M					
R86	E6					1-M		1-M				
R87	E7						1-M					
R88	E8									1-M		
R89	E9								1-M		1-M	
R90	E10									1-M		1-M
R91	E11										1-M	

